ANOMALY DETECTION
FROM COMPLEX TEMPORAL SEQUENCES
IN LARGE DATA

By

Daniel L.C. Mack

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

May, 2013

Nashville, Tennessee

Approved:

Dr. Gautam Biswas

Dr. Xenofon D. Koutsoukos

Dr. Julie A. Adams

Dr. Douglas H. Fisher

Dr. Gabor Karsai

To my unyielding will, dazzling brilliance, and commendable modesty.

All of which is a product of

the triumphs and struggles of myself and my ancestors.

ACKNOWLEDGMENTS

There are many people to whom I am grateful for guidance, support, and encouragement during my graduate education at Vanderbilt University. Without their help, this dissertation would not have been possible.

First, I would like to thank my parents, Leif and Patricia, for their love, support, and guidance throughout my life. Their belief in my abilities, and the encouragement for me to pursue my dreams has helped craft who I am, and always keeps me reaching for the next great challenge. The culture of reaching for lofty goals is also a product of my extended family, including Uncle Mike, Aunt Dee, Uncle Milt and Ti, as well as my cousins ZoAnn, Matthew, Chelsea, Shane and Crystal. Also, I want to acknowledge the memory of my grandfather, Louis Campana, who always expected my best, but had nothing other than unconditional love for me.

Next, I would like to thank Professor Gautam Biswas, my adviser, for the opportunity to work with him at the Institute for Software Integrated Systems at Vanderbilt University. During my graduate education, Gautam has been a continuous source of encouragement, guidance, and critical thought, for which I will always be grateful. Next, I would like to thank my dissertation committee: Professor Xenofon Koutsoukos for all his guidance and collaboration, Professor Julie A. Adams for the numerous suggestions and help in furthering my research, Professor Doug Fisher for his insightful thoughts on how to present my research to the community and Professor Gabor Karsai for providing a different and enlightening perspective on my research.

I would also like to thank the other individuals and organizations who have supported my research: Dinkar Mylaraswamy, Raj Bharadwaj, and George Hadden at Honeywell Aerospace, Minneapolis, MN and National Aeronautics and Space Administration under contract NNL09AA08B. I would like to acknowledge the support from Eric Cooper from NASA for the support on the VIPR project.

One's journey to discovery and research could not be complete without the camaraderie and support of friends, and colleagues. Over the past five years, I have benefited from discussions and friendship with many other students (and their spouses) at Vanderbilt University. I would like to thank Indranil Roychoudhury, Chetan and Aditi Kulkarni, Emeka Eyisi, Heath and Alison LeBlanc, Siyuan Dai (Go Irish), Stephanie and Jeremiah Weeden-Wright, Josh Carl, and Will Emfinger for the fruitful and enjoyable discussions about our research and our lives. They put up with me at my best and worst and I appreciate their time beyond words. I would also like to thank the administrators at ISIS, including Michele Codd, Kristy Kruse, Susan McMahon, Jaclyn Carter and certainly Tonya Coleman for their help organizationally with my research and for letting me tell terrible jokes and still laughing. A special thank you to Katy Pepper who helped edit this work, and her support as this research neared the end.

My time in Nashville and at Vanderbilt would not be half as productive and even less enjoyable without those whom I call my confidants. John and Jeanna Kinnebrew have been with me every step of the way and their support and kindness will stay with me throughout each new chapter of my life. I want to thank Tony Worley for the movies and golf over the last year of my research which helped me refresh after long hours at the office. I also would like to thank Tim Denbo for the music and tennis, as well as Lyndsey Godwin and Khette Cox for the many adventures. I am indebted to you all.

Lastly, to those I've loved and lost: Godspeed and thanks for the imprint.

Daniel L.C. Mack

Vanderbilt University

31 March 2013

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

"To study the abnormal is the best way of understanding the normal." - William James

As systems become more complex and the amount of data collected from these systems increase proportionally, new problems arise about how this data can be used to better understand system operations, monitor performance, and detect unsafe behavior. Of particular interest, from a safety viewpoint, is the problem of how this data can be used to improve the effectiveness of anomaly and fault detection schemes. Also, exploratory data-driven methods provide approaches for discovering previously unknown and undetected anomalies. If such methods are reliable and robust, they could play a very important role in improving overall system safety and operability. This thesis takes on this challenge, requiring the handling of large data sets, which are often originally only available in unstructured forms. This process of finding anomalies could be compared to looking for a "four leaf clover in a grassy field." Furthermore, the complexity of the systems makes the task of interpreting and evaluating anomalies an equally complex task, and we also deal with the challenge of presenting characteristics of detected anomalies in a form that can be easily interpreted by domain experts.

This problem is especially pertinent in engineering domains. Experts in these domains deal with a number of challenges in the detection of failures and interpreting abnormal behaviors in the operation of complex systems by analyzing large amounts of operational data. In engineering, these anomalies are often attributed to degradation in components and subsystems that arise from normal wear and tear, but also because of non nominal operating conditions or because of complex interactions between subsystems that were previously unknown to the experts. As experts discover new fault conditions and previously unknown

anomalies they are able to piece together the causes, and develop detection methods, thus making it easier to detect and respond to these anomalies in future operations.

On the other hand, the consequences of not finding anomalies that occur during system operation can be numerous, but for simplicity, they can be reduced to safety and monetary considerations. Early detection of a failure can avert high consequence failures such as loss of expensive equipment or life, and give system operators and maintenance staff sufficient time to repair the system before the failure results in disaster. With the increasing complexity of systems being attributable to the interactions among a large number of subsystems, the number of potential unknown anomalies and failures increases significantly. Early detection is paramount to avoiding failures from propagating into other subsystems, which makes it harder to identify the root cause of the failure.

While the "black swan" [157] problem in complex systems cannot be eliminated, the inability to discover unknown anomalies in a timely manner may exaggerate the consequences of these failures. Our research is driven by these motivations to solving the problem of anomaly detection in complex systems by analyzing large amount of operational data.

I.1 Research Challenges

I.1.1 Challenges from the Complexity of the System

Anomaly detection methods used to identify failures must be flexible and efficient given the varying size and complexity of the systems being examined. As systems become large and more complex in their operations, detecting anomalous behavior while avoiding false alarms can become difficult. Experts that build models adopt heuristics to mitigate the effects of the complexity, often making targeted simplifying assumptions that are applicable to the known faults and operating conditions. These assumptions may be restrictive, making a number of anomalies hard to detect, or to be misclassified when they are detected. Therefore, using the operational data to model pertinent complex relationships that given

system behavior may improve the detectability and classification accuracy of the known anomalies and reduce the misclassification of the unknown anomalies.

The complexity of the system leads to two research challenges that we address. First, there is a research challenge in anomaly detection to identify new anomalies related to this increasing complexity. If we can improve the number of unique anomalies detectable, we reduce the chances for surprise failures in the system, and therefore, a diagnosis can be better prepared to mitigate these failures through early detection.

A second category of research challenges related to the complexity is to improve detection of already known anomalies. This challenge may require us to find new information that provides more support for the anomaly or to get rid of previous simplifying assumptions that were incorporated into the detection model. In fact, this challenge may encourage the discovery of new representations of the complexities in the system. Expert can leverage these new representations to better detect the known anomalies. Improving detection often means producing new information which can be used to extend previous models.

I.1.2 Challenges from the Size of Operation Data for the System

Along with the challenges presented by the inherent complexity in the system and the complex interactions they imply, large systems naturally utilize more sensors to help monitor and regulate their components. Coupled with the improvements in sensor technology, improvements in storage mean that as more data is being produced that is also being stored for future analyses. Anomaly detection in such systems must navigate increasingly larger amounts of data, including more operational runs per system, a larger number of sensors, and increased precision in the sensors resulting in a much larger collection of signals that are captured for future analyses.

This increase in the overall amount of data produces a number of research challenges. First there is a challenge in effectively curating the raw collected data to make it efficient for transforming it into structured forms that can be used for systematic anomaly detection.

The challenge of curation is in the flexibility to allow for several approaches to work across the data.

The second challenge of large data is in integrating the curated data with outside information. Expert-designed systems, and additional datasets may produce additional valuable information, such as annotations of failures, and better specification of sensor values and locations.

Another challenge of large data is curation or structuring of the data to enable more effective and efficient use of data mining algorithms for exploratory analysis. While the number of instances, the number of features, and the length of the signal for each feature needing consideration, there is a challenge in how to make this data efficient for a variety of analyses that can include both supervised and unsupervised methods. The data may not be suitable in its current form for the appropriate learning method, and even after transformation may make the learning algorithm computationally expensive to use.

The other side of computational efficiency is in allowing the data to be efficiently analyzed either by supervised or unsupervised methods. Further, the results generated by the algorithm should be translated into structure and forms so that a human expert can easily extract and interpret the new information provided and assimilated this knowledge into the detection models. When the data set is large and contains a variety of different information, characterizing anomalies becomes a harder problem and can impede the expert's ability to integrate this new information. Our goal in this work, is to produce methods for discovering anomalies in large complex systems that address these research challenges.

I.2 Problem Domains

We utilize two domains in this research. The structure of the data in both domains reflect the research challenges we are solving in this work. There are differences between these domains which highlights the nuances of the challenges and our approaches to anomaly detection. In the first domain, we examine aircraft flight operations data collected over a

five year period for a regional airline. In the second domain, we explore data recorded from pitches thrown in Major League Baseball games from the years 2009 to 2012.

I.2.1 Aircraft Flight Systems

The data in the aircraft flight domain represents the flights of multiple aircraft of the same type that belonged to a regional airline. Aircraft represent complex systems with multiple interacting subsystems. A pilot's choices while operating the aircraft and the operating environment, such as the weather, further increase the diversity the data that needs to be account for when analyzing. This data was originally stored in a very large number of CD-ROMs, where each CD-ROM contained data from multiple flights. Each flight is recorded as a separate instance, each with a set of 182 sensors, and each sensor records data for the entire flight with the data collected at sampling rates of either 1Hz, 2Hz, 4Hz, 8Hz, or 16Hz. The collection of these recorded flights all together represent a total of $\tilde{0}.7$ Terabytes of data.

The goal for anomaly detection with this data involves detecting unusual flight conditions; caused by equipment faults or degradation, environmental conditions, and pilot actions that may be characterized as aviation safety incidents. Aviation safety incorporates situations related to the aircraft that may cause harm to the aircraft, its occupants, or the surrounding environment, such as people and property on the ground. Anomalies that impact aviation safety include mechanical malfunctions, pilot decisions, and environmental conditions that are unsafe for aviation. Our research with this domain is to help improve already known mechanical failures, and to identify new and previously unknown anomalies that could potentially result in aviation safety issues.

The research challenges we address for anomaly detection include the complexity and the size of the data. Our solutions to the problems must include the curation of this data, and the integration of expert knowledge with the results generated by our data mining algorithms. In improving the models for known anomalies, we must deal with the complexities

of the aircraft systems and building subsets of the data to help target the anomaly detection process. For finding new anomalies we must deal with the inefficiency of the structure of the data for exploratory analysis as well as incorporate the complexity of the temporal signals from the features on the aircraft to look for new anomalies.

I.2.2 Analyzing Pitcher Performance

The second domain we examine in detail is pitcher data from Major League Baseball games. This data includes measurements of pitches thrown during the game. These measurements attempt to identify components of the each pitcher's specific release motion and position for a number of different pitch types. These measurements taken together approximate the "mechanics" of a pitcher's throwing motion. The mechanics represent the way a pitcher throws a specific type of pitch, and includes the release point in relation to their body, where the ball ends up at home plate, and the amount of spin they place on the ball. Thanks to other researchers efforts, these measurements for each pitch thrown in a Major League Baseball since 2007 have been collected are available in a curated database. This database contains over 4 million pitch records for 1900 pitchers.

The goal of anomaly detection in this domain is to identify games where a pitcher profile of thrown pitches differs from his expected performance. In the aircraft flight domain, the anomalies are either aviation safety related, or just rare events. The rare events in the flight domain could include unique but not dangerous weather patterns, or the pilot's decisions were unexpected but not catastrophic or dangerous. If the aircraft landed successfully without incident, the anomaly may have no immediate consequence. In the baseball domain, an anomalous game may be a bad game for a pitcher, where they gave up a lot more hits and runs than normal, or a very successful game, where they gave up less hits and runs than normal. In contrast to the goals for the aircraft flight domain, our discovery of anomalies may point to better games than normal, just as much as our approach could find subpar games. From either type of anomaly, our goal is to characterize the pitcher's mechanics

that differed from their nominal behavior, and this could form the framework for further study.

This domain and data contrast with the aircraft flight domain, because of its complexity across different dimensions. In the case of baseball, the data is based on a human interaction as opposed to mechanical equipment. Humans are decision makers which results in variations of their approach to and within a game. There are emotional and physiological parameters that affect humans and this needs to be taken into account when analyzing anomalies.

In spite of this important difference, there are a number of similar challenges for exploring the baseball data. While the curation of the data has been already accomplished, we are dealing with a domain where identifying anomalies is a relatively new approach. In exploring the data for new anomalies we must face the challenge of dealing with a large number of pitchers and in this case temporal sequences that differ in size across each game a pitcher throws. The non-standard representation is a challenge in utilizing unsupervised learning methods such as clustering. Further we must face the challenge of how to characterize anomalies effectively when dealing with varying signals. These problem domains and their contrasts provide new dimensions for exploring approaches for identifying interesting anomalies.

I.3 Approaches to Anomaly Detection

We have developed two approaches for anomaly detection in this thesis. The first approach is a method designed to address the research challenge of improving the detection of already known anomalies. This approach will identify new relationships for better characterizing a known anomaly, and makes them available for an expert in a form that facilitates their updating existing anomaly detection models. The second approach addresses the challenge of identifying new anomalies using unsupervised learning methods. This approach is

exploratory by nature, and is also designed to provide targeted information to the expert to help them characterize the nature of the anomalies.

I.3.1 Approach for Improving Anomaly Detection

- This approach leverages expert information to constrain the problem by targeting specific known anomalies and it addresses the large data challenge by reducing the size of the data considered to segments of anomalous behavior that can be compared against nominal behavior. This makes model refinement more efficient, and the constrained data makes it easier to characterize the anomaly.

- The approach is used with the Aircraft Flight Domain to help improve existing diagnostic models used in Aircraft Diagnostic and Maintenance Systems. Improving these models, improves the detection accuracy and time for known failures. Therefore, this is often framed as a knowledge engineering task. The knowledge engineering framework requires that the results of the learned models be easily interpreted by the experts in the context of the existing diagnosis model. This improvement can help avoid aviation safety incidents. The improvement may also reduce the cost routine maintenance situations to on-demand based maintenance.

- The end result of this approach are targeted improvements to the original model. These improvements increase the accuracy and detection time of future anomalies in aircraft flight systems.

- We expect that one of our main contributions in this work will be a knowledge engineering solution involving the use of a general framework for applying targeted anomaly detection using expert guidance. Another research contribution will be the application of this framework to the aircraft flight system domain. This implementation of this framework will help improve the diagnostic accuracy of an industry based diagnostic reasoner.

I.3.2 Approach for Discovering New Anomalies

- Unlike the first approach, we do not rely on significant amounts of initial expert knowledge, and thus the starting dataset is less constrained and includes more overall operational instances of many different types in the data. This change means that we can explore as much of the data as computationally possible to discover new anomalies.

- We improve the computational efficiency by exploring dimensionality reduction techniques that are applied to the high dimensional data. With the increase in the amount of data being used, dimensionality reduction addresses the research challenge of making the data more computationally efficient for future analysis using supervised and semi-supervised methods.

- This exploration produces previously unknown anomalies and our approach helps the expert characterize these new anomalies by highlighting their relevant features. Our approach to characterization addresses another research challenge of large data, by helping experts examine the larger dataset in a systematic way with higher precision.

- This approach is used with both the data from the aircraft flight systems, as well as the baseball domain. Each domain provides a contrast into the applicability of this approach and the search for different kind of anomalies.

- The end result of this approach should be a collection of nominal data and a collection of anomalies, with a series of features that best differentiate the anomalies from the nominal data. These features and anomalies will be characterized as to their impact on the specific domain.

- We expect our the first research contribution from this work to be an exploratory approach to discovering previously unknown anomalies in large data from complex

systems. This contribution includes an end to end framework for exploring large amounts of this data.

- The dimensionality reduction in this approach should produce its own research contribution, which is the testing of multiple reduction techniques to understand their effectiveness for anomaly detection.

- Lastly, our final research contribution is the successful use of our approach on the two problem domains. This contribution includes discovering new and previously unknown anomalies in aircraft flight systems that may impact aviation safety. For the baseball domain, this should include the discovery of novel anomalies for games that pitchers throw and the impact of a pitcher's mechanics in those games.

I.4 Dissertation Organization

The remainder of this dissertation is organized as follows: Chapter II provides a review and background on the primary concepts and approaches to anomaly detection and data mining. The chapter describe the different types of anomaly detection and the various types of machine learning algorithms used to build models for anomaly detection. Chapter III provides our overall research approach and methodology for our contributions in this dissertation. Starting with a more formal description of the data, and the two approaches to address the challenged for anomaly detection in complex systems with large data. Chapter IV describes our first approach using supervised anomaly detection, and how the knowledge engineering task is used to improve diagnosis. The chapter uses case studies from the aircraft flight systems domain to demonstrate the approach. Chapter V begins describing our second approach in more detail, but focuses on the dimensionality reduction techniques we apply. Using a series of experiments we explore the tradeoffs between different techniques and choose the techniques to use in our approach. Chapter VI describes our second approach in full detail, and its application to the aircraft flight systems. We compare with

previous work in the area and demonstrate the effectiveness of our through a series of case studies of discovered anomalies. Chapter VII demonstrates the same approach as applied to the baseball domain. We identify a set of pitchers and explore the anomalous games and how they relate to the specific mechanics of the selected pitcher. Lastly, Chapter VIII provides a summary of our approaches, and their demonstrations on our problem domains as well as future research directions.

CHAPTER II

BACKGROUND ON ANOMALY DETECTION

II.1 Anomaly Detection Models

Anomaly detection is defined as the process of using models to identify behavior that is different from the normal behavior of a system [21]. Anomalies can be referred to by vocabulary such as outliers, abnormal behavior, surprises, unusual instances, exceptions, and aberrations [21]. This research uses outlier, abnormal behavior, and anomalies to refer to the same concept. Early detection of anomalies in a reliable and robust manner is important to maintain system operations in an efficient and safe manner. Applying these approaches to studying the purchase histories for consumers may result in identified abnormal behavior being recognized as fraud, requiring efficient and fast detection to prevent further unwanted purchases from an account. In network systems, abnormal behavior may indicate an intrusion, requiring action to keep the rest of the computers on the network from being compromised. In every case, the problem involves understanding normal behavior, and using models that can find and flag new actions or sets of actions as abnormal. Figure 1 from the literature [21] shows a simple visual example of this problem. For complex systems, there can be multiple types of normal behavior. The focus of anomaly detection is on the behavior that exists outside of these areas. Anomalies can appear differently, for example they can appear as individual points, such as o_1 or o_2, where each is a single instance separated from the normal clusters. The point at o_2 also show that, while anomalies are different from normal behavior, they may not be radically far from the nominal clusters. Anomalies that are not well-differentiated may be problematic because they can be harder to detect, but just as critical in their consequences. Lastly, abnormal behavior may be common enough to form groups such as O_3. These small collections can become the framework for defining anomaly detection techniques, rather than the single points.

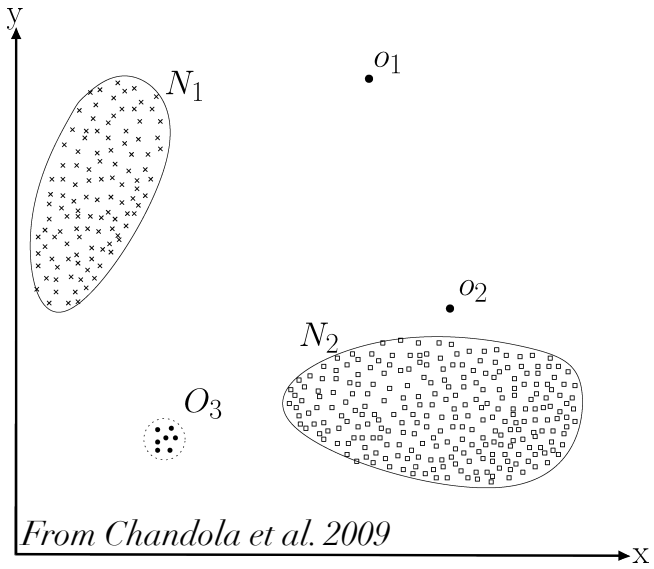

Figure 1: Abstract Example of Anomaly Detection

II.1.1 Anomaly Types

The techniques for finding these different behaviors fall into three categories of anomaly types. The first two are "point" and "collective" anomalies. The third is the "contextual" anomaly, which provides an augmentation or explanation of the data when analyzing the first two types. The choice of anomaly types changes the way we search through data and the type of models one builds for detection.

"Point" anomalies represent the most straightforward idea of an anomaly. This type considers data points as independent of one another. In general, a point anomaly is found by looking for specific individual samples in the data that are not similar to the rest of the data set. Applications of point anomaly detection include credit card fraud [3] where a single purchase data point is determined to be anomalous based on knowledge of other normal transactions. As a general idea of fraud, a single point-based anomaly against a large set of normal data points can apply to several different domains [14], including network intrusion and medical fraud.

In contrast to single points being anomalous is the notion of a "collective" anomaly, i.e., anomalies that represent situations where the anomalous behavior develops and extends

over a number of data points that extend in time or space. A collective anomaly may require several instances in the data, often occurring sequentially to define a trend to characterize the anomaly. Unlike point anomalies, collective anomalies occur only in datasets where the points that make up collective anomalies can be related to one another [55] by a function of the features. For example, in fault diagnosis, certain degradation of components occur slowly over time. This incipient type of fault can only be discovered when there are enough points that indicate abnormal activity in a progression that characterizes the failure. If the collection of sensor values is represented as a set of signals, this anomaly type is found by analyzing a sequence of signals that indicate the slow degradation. Contrast this with a group of anomalous points, which may occur at different points in time and different locations. The collective anomaly requires the set to be anomalous together with some relationship.

Applications involving collective anomalies include medical services that identify problems in physiological data such as electrocardiogram output, where a small problem over time may be more noticeable to a machine than a human eye [21]. Other sequential applications include sensor data of celestial observations [122], as well as sequence data for intrusion detection found in system calls [22]. Other applications for collective anomalies involve graphs, where a collection of nodes must all be considered part of the collection in order to identify an anomaly [156]. Applications that utilize the graph based representation include intrusion detection, such as finding botnet clients [11].

Anomalies that are discovered when characteristics of the data are used to filter relevant data are termed "contextual" anomalies. This is to say, that behavior may only seem anomalous in a specific context, but not in the data as a whole. A context may be formed from external sources, e.g., how and when the data was collected. An example is building a model for network intrusion from data on a specific cluster of machines. A context may also be formed from internal characteristics of the data, e.g., using the recorded latitude and longitude of a geospatial dataset to group instances together. The sources used to build

context in the data are referred to as "contextual" [21] attributes. When these attributes are possible features in the data, they are removed from consideration as features that may indicate an anomaly within the context. The remaining features used to discover anomalies are referred to as "behavioral" [21] attributes.

A classic example to motivate contextual anomalies is the problem of finding areas with abnormal rainfall in a particular region [21]. Certain amounts of rain fall in the data will look much different when compared to amounts in the overall dataset. However, when examining these different amounts in the context of location and time of year, e.g., the rainy season in South East Asia, the amounts may be normal. An amount of rainfall in southwestern India during the monsoon season is not abnormal, although a lack of rain at that time of year would be anomalous; however, that amount of rain at any time in the American Midwest would certainly be characterized as anomalous. There are several good applications of context-based anomaly detection that use spatio-temporal relationships with environmental data [35].

Since contextual anomalies refer primarily to building context to define behavior, inside the context we must select which type of behavior (either "point" or "collective") to focus on for detection. Point anomalies can be used to examine fraud within context. Simply by selecting the time of year as the contextual feature, certain credit card fraud may be easier to detect, since an event such as the holiday season may change what we consider to be an abnormally high purchase amount [45]. As with the point anomalies, contextualization is used for other fraud-based domains such as intrusion detection [134].

Similar to the point anomalies, a collective anomaly can be used in conjunction with contextual features. Examples that combine the two include detecting public health issues from health surveillance streams [9]. The context will be the location of the surveillance streams, and the features being considered are signal based and requiring a collective anomaly detection scheme. Another example of contextual and collective anomalies is using spatio-temporal features to build models for analyzing hyper-spectral imagery [153].

II.1.2 Data Considerations

Deciding which type of anomaly detection, and the method used to build the model requires looking at the nature of the data. The nature of the data is based on how it was collected and how it may be transformed through preprocessing. Raw data may be capable of containing several different types of anomalies, including different types of contextualization. Examining the data for the manner of collection, the type of features available, and knowledge about class labels will guide the use of appropriate types of anomalies and models to build.

In general, when describing datasets, each separate occurrence in the data is referred to as an instance or sample. Instances can be placed in order, such as with a time stamp or just considered to be independent samples. Each of these instances has a set number of measurements or details. These measurements are described as features that can be used to understand the nature of each instance. It is possible that not every instance will have every measurement, which in turn means that a feature may be missing. Missing features are important to account for when building a model, since that model may have to classify an instance as abnormal without all the information. Understanding the type of the features collected is important; numerical information, whether it be discrete or continuous, can impact the types of algorithms used or require certain types of preprocessing. Features with categorical information, such as text responses also influence these decisions.

For different anomaly types, it may be useful to transform the data and preprocess certain features, or the entire dataset to produce appropriate features for anomaly detection. For example, in flight data, it may be best to transform raw sequences of measurements about the engines during flight, into single instance for each engine that records statistical information of the signals into a set of features, such as max temperature or average engine speed at takeoff. This reduction to the dataset may allow the dataset to be used with different algorithms.

II.1.2.1 Data and Anomaly Types

Point Anomalies For point anomalies, the necessary organization of the data is minimal. Point anomalies can be sought out in any data where a data point can be modeled as an event and where events can can be compared against one another. Events can be defined by features of many types, such as pure numerical data for credit card fraud [3] and categorical data for detecting anomalous session activity using decision trees [161]. While the organization of the data can be general for point anomaly detection, the anomalies may suffer from a lack of contextualization if the data possesses relationships between features. The rain example given earlier is a case where, without contextualization, data that appears anomalous may be nominal given a contextualized comparative sample.

Collective Anomalies Collective anomalies require a dataset where each instance can be explicitly related to the others. Collective anomalies are found in data that has explicit features that are ordinal, such as timestamps or location. The example of fault diagnosis as a collective anomaly is based on the fact that each instance relates to another in time. Common data for of this type include signals from systems, so there are popular techniques to look for these changes over the signal, such as compression and complexity analysis [77]. While temporal order is straightforward to identify in many applications, finding this relationship for other datasets can require analysis before building models, such as seaport surveillance to find anomalous vessel tracks [89]. Collective anomalies exists over both numerical data such as classification of physiological signals such as electrocardiograms [31] and categorical data used in intrusion detection [55]. Similarly to point anomalies above, if the data was collected over several contexts but the detection occurs over the entire dataset, the results may be poor or misleading.

Contextual Anomalies Contextual anomalies refer to the use of contextualized attributes to search for point or collective anomalies, so the data must possess features that can be used to group the data into different contexts. This grouping may be explicit, e.g., temporal groupings like months of the year for credit card fraud. The contextual attribute may need

to be data mined, such as clustering latitude and longitude in aircraft takeoffs, to find general areas where the data was collected, like airports. These attributes split the dataset into many smaller datasets for the application of other anomaly types to be detected. Contextualization may happen on multiple levels, such as contextualizing by location of an airport and then by time of the year, to eliminate weather patterns from effecting takeoff anomalies. Unlike point anomalies, where the data can be used as it is, contextualization and collective anomalies require the expert to understand the nature of the data to find relationships and appropriate contextualization to transform the data into smaller sets for building the models.

II.1.2.2 Data Labeling and Supervision

Selecting methods for building models from data to detect anomalies requires understanding if there are labels for the data. The labeling of data is defined as selecting, or building, a feature to be the class label. The labeling marks each instance in the data as belonging to at most one type in the range of possible values for that feature. In many methods, the label is a flexible choice where the selection may be numerical, and usually discrete, or categorical. This choice in anomaly detection may be straightforward. If knowledge of what constitutes normal behavior in the data is known, then instances in the data with that behavior are labeled as nominal, and if anomalies are known, the instances that are known to be anomalous are labeled appropriately.

The existence of a label for the data focuses the choice of methods based on the level of supervision in the learning algorithm. In anomaly detection, the presence of class labels in the data change as anomalies are extracted from the data. If the labels are well known for both nominal and anomalous instances, the data may be used with supervised learning, which will attempt to discriminate between the nominal and different anomalous groups. For example, when considering different failures on an aircraft, a supervised model will attempt to discriminate between different types of anomalies as well as the nominal case. The

greater the number of classes, the more complex the model will have to be to differentiate between the larger number of groups.

If the labels are known only for one class (usually the nominal case), the use of semi-supervised learning methods is appropriate, as they build a very specific model for discriminating between the known class and the "others," which is a catch all group for anomalies and other behaviors that may have to be analyzed further before they can be labeled.

Lastly, if knowledge of nominal and anomalous behavior is completely unknown, then the data must be analyzed for common and uncommon behaviors. Unsupervised learning methods attempt to separate data based on emergent behavior between instances and other commonalities.

II.2 Using Data Mining Methods to Build Anomaly Detection Models

Knowing the domain and makeup of the data defines the methodologies and techniques we develop and use for anomaly detection. The next step in the methodology is to choose appropriate machine learning methods to build the nominal and anomaly models that we will use for detection and characterization of anomalies. For example, if the data objects are labeled as nominal and anomalous, one strategy is to directly develop supervised and semi-supervised methods for classification and characterization of anomalies. On the other hand, if no differentiating labels are initially associated with the data objects, unsupervised learning methods are applied to understand and characterize the data.

Building a complete anomaly detection methodology can involve using a number of machine learning algorithms along a chain to build a complete application. The supervision of these algorithms is one of the considerations when building the approach. For example, in cases where the class knowledge is not available apriori, the expert may build a pipeline of machine learning methods that utilize the output of one method for processing the data to use as the input of another method, to produce the desired model, i.e., using labels derived from unsupervised methods to build a model from semi-supervised methods.

As well as the supervision of the algorithm, the designing of the anomaly detection scheme is guided by the generality of the model. The use of discriminative or generative techniques offers tradeoffs in terms of the ease and speed of building the models and a model's ability to handle missing or unobservable evidence through complete modeling of the environment provided by the data. Balancing these trade-offs is a function of the domain and the expectations for the application.

Also in many real world applications, it is important to involve experts in the analysis and decision making loop. Choices in algorithm and methodology impact the ability of the expert to understand the details of a particular model, and how to interpret any anomalous results based on the model. Since one of the pillars of our research involves improving previous diagnostic models of embedded vehicle systems that requires methodologies that incorporate knowledge engineering approaches, which necessitates choosing algorithms that give the expert information in the right detail and the right format that they can interpret the new information in the context of existing models. Assessment for such a task still involves empirical testing of the improvements with data.

Among the details that are also worth considering, are the number and type of parameters to tune for optimal performance. Parameters with precise tuning requirements are not be suitable for systems that operate in diverse environments. Since the datasets for this research are large in the number of objects and high dimensional, any choice in algorithms or design methodology should be made with an eye toward applicability to large, high dimensional datasets.

II.2.1 Discriminative Models and Discriminant Functions

Discriminative models and discriminant functions represent two similar types of machine learning algorithms. Both types restrict their learning to relationships between the observations, or the features in the data and the labels. From the literature [12], for each class in the data C_k and the input x, discriminative models use the data to directly learn

posterior class probabilities $P(C_k|x)$. Decision theory is used with these probabilities to assign new input to a given class. Compared with the generative models that must learn joint probabilities over the evidence as well as the posterior probabilities, these models are computationally easier and useful for general classification.

Discriminative modeling approaches to making decisions have evolved over time, starting with simple structures. Among the original discriminative modeling techniques are purely probabilistic methods, such as logistic regression [12], and information-theoretic methods such as Decision Tree classifiers. Decision trees use the information theoretic principles to split the data and produce a tree model of decisions based on the features [170]. These structures are easy to interpret, but as the number of features, the data, and the overall decision space grows, a standard decision tree will grow too and over-fit the data, possibly becoming too large to read and interpret efficiently. Keeping the tree shallower, and thus smaller is referred to as pruning which keeps a model more general and allows for better readability [110]. Pruning is a process of removing or never growing specific nodes at the bottom of the decision tree, thus avoiding the addition of decisions that are often the noisiest. Several methods exist for pruning decision trees [76, 108].

In contrast to the discriminative models are discriminant functions, which are the simplest methods for classification, using a mathematical function to map the input x directly to a class label. An example is a two class problem, where function $f(x)$ when applied to x, returns a 0 or 1 for either class C_1 or class C_2, respectively [12]. A standard implementation of such a method is a linear discriminant function using the mean square error criterion. These functions include linear, polynomial, and radial basis functions. The expert must chose which function to use for the classifier, often experimenting with many and using empirical results to guide the final choice. Each increasingly complex function that is chosen for the classifier may improve the accuracy, but will also begin to over-fit the data and reduce generality [12].

A disadvantage of discriminant functions are that without the posterior probabilities,

it is difficult for the algorithm to understand and minimize risk (formally the expectation of loss), as well as better model the environment in terms of class priors. For an expert, while the methods are fast to learn and simple to understand, they provide little in the way of interpretable information. Examples such as data that represents decision spaces for functions, like XOR (exclusive-OR logic) demonstrates where these classifiers can fail no matter how complex the mathematical functions chosen.

Issues with such systems helped motivate the evolution of discriminant functions through the production of more complex structures. For example, the multilayer perceptron (Neural Network) algorithm [12] takes the simple perceptron classifier [111], and builds a layered structure of perceptrons trained using back propagation of information through the structure. These Artificial Neural Networks are capable of accurately separating most any decision space, given enough time and data to optimize the parameters. It is also referred to as a "black box" technique, meaning that the information learned from the data produces values that lack easily understandable semantic information. A lack of semantic information impedes an expert's ability to better understand the system through examining the data-driven model. While the Artificial Neural Network may be accurate for an application, in many fields where the model itself is important for knowledge engineering activities with an expert, the Artificial Neural Network is a poor choice.

In the 1990s, AI and Machine Learning researchers began investigating methods for building function-based discriminative models that remained simple and also focused on building generality into the classifier to prevent over-fitting, as opposed to post-processing, such as pruning in decision trees. Among the advancements were kernel methods, such as Support Vector Machine classifiers, or SVM [162]. SVM models are function-based classifiers (linear), but the coefficients learned from the data were optimized to produce a separation of the training data that would be as general as possible. These models in turn may suffer from a knowledge engineering standpoint as the coefficients are not semantically valuable to the expert.

SVM research moved towards methods for separating more complicated, or non-linear decision spaces by extending the original SVM classifier into higher dimensional spaces. The spaces are found through transformations applied to the data in techniques called kernels, with the original linear kernel being the simplest. The transformation maps the original features into higher dimensionality, while retaining a format desirable for training a SVM [114]. The use of the kernel involves further loss of semantic value for the expert because the kernel transformations abstract away the original features and produce a new feature set, which is combination of the old.

These discriminative learning classifiers are useful when the data is labeled appropriately (supervised and semi-supervised). The models are also a good choice when the application does not involve complex interactions between features that might improve their accuracy (or instead can be abstracted away with Kernel transformations).

II.2.2 Generative Models

In contrast to the discriminative models and discriminant functions are the generative models. These models do more than just learn the posterior probabilities for data and classes. Instead, generative models learn joint distributions $p(x, C_k)$ over the features and classes. These models are capable of describing the interactions among different observed variables [12], using distributions built from the data to model the environment. The process of learning these distributions is more time consuming than just the posteriors or function coefficients. Theoretical and empirical analyses have shown that generative and discriminative models differ in their generalization behavior, as well as the speed and accuracy of learning [41, 41, 73, 171]. These models are useful in their ability to better describe an environment, especially when there are no known class variables, but generative models are capable of a different over-fitting through building tenuous relationships found in the distributions of the data that do not exist in the operating environment.

II.2.2.1 Clustering Algorithms

Generative models naturally relate to unsupervised learning, where the use of the joint probability distributions can facilitate the grouping of similar objects that help discover relationships between features and to discover common patterns in the data. The major group of techniques in unsupervised learning is clustering. Clustering can be broken down according to two major attributes. The first is the process of how to form the clusters from the data. The focus is on connectivity models, centroid clustering, distributions models, and clustering by distance density. The second attribute is the types of features (numerical and categorical) a clustering algorithm can include when it looks for patterns.

Connectivity models are often referred to as hierarchical clustering. These models are constructed utilizing a distance measure between instances. These distances refer to how closely two data points may be related to one another. This class of algorithms is broken down by the direction of the hierarchical process, either "agglomerative" (build up the hierarchy) or "divisive" (move down the hierarchy) construction.

Agglomerative clustering [57] starts with each instance in its own cluster, and at each step the two clusters with the smallest distance are combined together in a single cluster higher up the hierarchy. In agglomerative clustering, when clusters contain multiple points, a distance must be measured between two clusters to determine if they contain the next smallest distance to select for joining into a larger cluster. There are several methods for determining this distance. The most common include "single link", "complete link" and "average link clustering." Single link looks for the smallest difference between the points in one cluster and the points in another and uses that to measure the distance between the two clusters. Complete link is more conservative and chooses the largest distance between any two points in the the clusters. Average link clustering finds the average distance between the clusters. Agglomerative clustering ends when all points belong to a single cluster at the root of the hierarchy. Divisive Clustering on the other hand starts with one cluster and begins to split the clusters recursively until all points have been split into single clusters [27]. This

process is often exhaustive, as the cost function for each possible value to split the cluster must be measured to find the best choice.

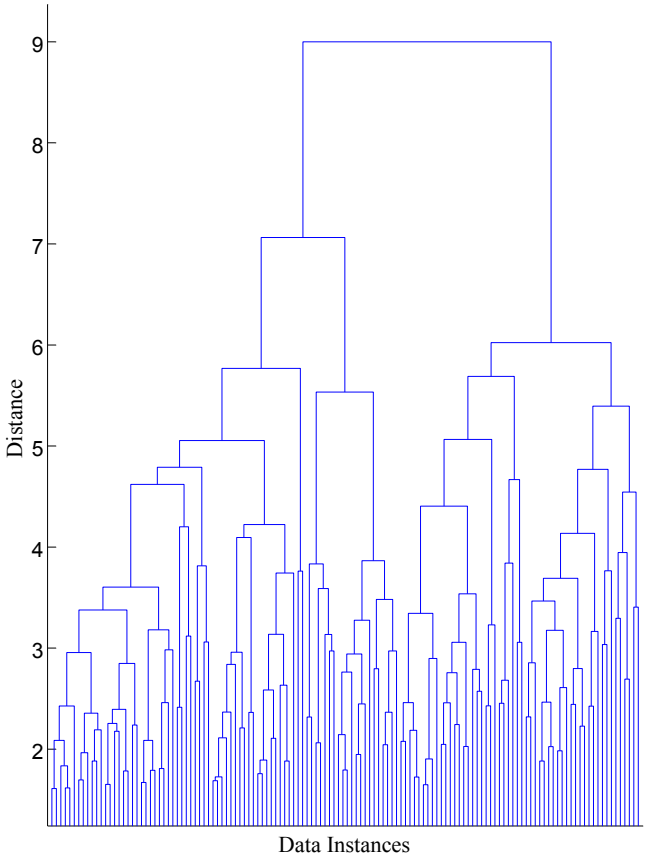

Figure 2: Example Dendrogram

The process of the splitting or merging in agglomerative clustering is recorded and displayed in a visualization known as a dendrogram [125]. The dendrogram, such as the example in Figure 2, shows the connections in a vertical manner, where clusters merge to form larger groups as one moves up in the hierarchy. The y-axis shows the distance measure and helps identify the distance at which two clusters had the smallest distance and were thusly joined. Connectivity models that utilize the dendrogram visualization produce a possible taxonomy for the domain. Understanding where instances belong to a common root allows for examination of those instances to find interesting sub-patterns

in large groups of data. The dendrogram can also be flattened into a set of clusters by utilizing a cutoff of the distance between clusters. This flattening with the cutoff ignores any sub-clusters.

There are a variety of dissimilarity metrics that operate on a variety of data types, and so hierarchical clustering is used to group data objects with both categorical and numerical features. Techniques that utilize this approach include conceptual clustering algorithms [49] such as COBWEB [47] and Similarity-based Agglomerative Clustering (SBAC) [13]. Further work on hierarchical clustering algorithms provide greater control on how to balance quality of the clusters with computational efficiency through the use of iterative strategies of organizing the data [48]. Other algorithms may only work with numerical data (discrete and continuous) such as Factorized Minimized Message Length Clustering(Factor SNOB) [166, 167]. The applications of connectivity models include diverse areas such as bioinformatics for finding relevant gene features [142], and textual based learning for areas such as search query log mining [8].

Centroid clustering constructs clusters based on how closely an instance is related to the a center point calculated from a number of other instances. K-means clustering [61] is an algorithm that implements this model. The algorithm requires a parameter of how many clusters are believed to be in the data, and initializes random means for each cluster. After each instance in the data is associated with a single cluster, the means are recalculated for each centroid and the instances are re-associated with the new means. Iterations of the algorithm occur until there is no more movement from the clusters (both in affiliation and in the centroids). The clusters in this model are "hard clusters" with "strict partitioning", meaning every instance belongs to exactly one class. K-means is an NP-Hard problem, and research exists into making the algorithm efficient through heuristics [74], as well as producing variations that relax the k-means algorithm, such as fuzzy k-means [95, 119]. The algorithm has also been transformed to handle categorical features with k-modes clustering [68].

Centroid clustering is similar to distribution based clustering in that both use mean-based centers; however, distribution based clustering goes further by defining distributions (often mixtures of several normal distributions) that represent each cluster. The common algorithm that performs mixture modeling is the Expectation-Maximization algorithm (EM) [12]. EM clustering follows a similar strategy as k-means, by taking as a parameter the number of clusters and initializing random cluster centers. Rather than only means, there are also variances for each cluster. As in k-means, the algorithm iterates, looking at cluster membership for the instances and changing the means and variances at each step. Unlike the k-means approach, this clustering model uses "soft clusters," which means that an instance can belong to more than one cluster. Among the many implications, this type of affiliation changes the condition for halting the algorithm. Instead of looking for when cluster affiliation no longer changes, EM measures the log-likelihood of how well the clusters describe the instances in the dataset. The affiliation also means that an expert can directly determine how good the clusters are for describing different behavior through a series of distributions for each feature. By definition, building a distribution requires numerical data, so EM clusters will operate on either discrete or continuous features. Clustering with EM has been used in a variety of applications, including diagnosis as an unsupervised anomaly detection technique [70], and classifying of images, such as celestial objects [90].

The "soft clusters" used in EM are also strict, meaning that objects may belong to any number of clusters but must belong to at least one. It is important when working in certain environments that instances in the data that may not align with other behaviors are not clustered for the sake of a required membership. In density based clustering algorithms, such as DBSCAN[98], the model works with "hard clusters," but allows for "strict partitioning with outliers." Instances are allowed to remain unaffiliated and examined as a separate set. Density based clustering groups points together only if they are close enough to one another by a given distance metric (such as a Euclidean distance or Mahalanobis distance).

If for a given instance, the distance with another data point is within the reachability parameter η, then the two points may be close enough to begin a cluster. If with a threshold k, enough points are within the reachability for any given instance, then the set becomes a cluster. If the points grouped together is smaller than that threshold, the basis point is left as an outlier. The benefit of distance metrics, is that they can be built for many different data types, allowing density based clustering to be successfully used with both numerical and categorical data. Distance metrics; however, can experience difficulty calculating over large feature spaces, thus DBSCAN can suffer from poor clustering for high dimensional datasets. Also, the choice for the parameters can be difficult to know ahead of time, requiring estimation. The ability for outliers in the clustering makes density based clustering suited to unsupervised anomaly detection for applications, such as Aviation Safety [94] and Network intrusion [92]. Density based clustering also has seen use as the clustering algorithm for geo-spatial databases [141].

II.2.2.2 Bayesian Networks

Other generative models that handle a variety of supervision tasks include Bayesian Learning algorithms, where the model is a directed acyclic graph connecting evidence nodes to potential hypothesis nodes ,i.e., the information of interest to be derived from the gathered evidence, e.g., fault hypotheses given symptoms. The variables in the data, including the potential class label, are vertices, and the correlations amongst the variable are represented as directed edges with a conditional probability distribution that links the evidence nodes to the conclusions or hypotheses [123]. Bayesian networks provide a compact representation for drawing inferences across the entire model in order to understand the probabilistic outcomes based on observations. Model learning with these algorithms run as a two part process. First, the algorithm looks for correlations in the variables to build the directed graph. The second step is known as parameter estimation, where the distributions later used for inference are estimated from the data [51]. These algorithms are generative

and cut across the supervised, semi-supervised and unsupervised learning spectrum. We adopt the Bayesian framework for part of our anomaly detection studies. Another benefit of these structures is that they easily handle missing evidence. In the case where a system sensor goes offline, a Bayesian Network can use that evidence's conditional probability distribution and marginalize the variable, still producing a classification outcome for the evidence albeit with less precision. That lack of precision; however, will be reflected in the likelihood (i.e., the probability distribution) associated with the result.

There are two attributes that vary for Bayesian learning algorithms: (1) limitations on the structure (and the manner with which it is built) and (2) the type of probability distributions used at the vertices (continuous or discrete). Algorithms limit the structure for many reasons including, the domain makes assumptions about the interactions with the variables, to keep the structure simple for analysis, and to make reasoning with the structure tractable for large feature spaces. Some structures are so simple, they can be built without analyzing the data. The most simple of these structures is the Naïve Bayesian network (NB) [111]. This structure connects every feature to a root node (which acts as the class node for classification tasks), and nothing else. The benefit for this structure comes from the assumption that the evidence is independent of one another, given knowledge of the class. This is utilized when using the structure for classification tasks (where the inference is a conditional probability given the class type). While this assumption may be a stretch in many practical problems, it reduces the complexity of the system, and can be quite useful for several tasks, such as text classification [93].

Relaxing the fixed structure of a NB graph is in essence relaxing the independence assumption of the evidence. Learned structures that introduce this loosening of the independence assumption, include augmenting the NB structures with a limited amount of general relationships like Tree Augmented Naïve Bayesian networks (TAN) [51]. Other structures may not form a NB structure, but instead limit the relationships between evidence based on the data. Perhaps the most well known is Markov Blanket induction and

includes Partial Bayesian Networks (PBN) [102] and methods known as Local Causal Discovery (LCD) [4]. The use of limited relationships between evidence produce DAGs with increased generality, but that still have similar structural properties. The algorithms for building these augmented structures are search algorithms, where the search is for structures that optimize a measurement such as Bayesian likelihood (like PBN or LCD searches for an optimal Markov Blanket around a class node), or search for internally similar structures, such as as a minimum weight spanning tree structure for a TAN, which is then augmented with links known ahead of time (the class node is connected to every evidence node). In either form, the data is used to discover the relationships. The improved generality from these type of structures can come at a cost of producing relations between evidence that are hard for experts to understand, and can become time intensive in very large, high dimensional datasets. These sort of structures have seen use in areas such as diagnosis (as Supervised Anomaly Detection) [91], as well as in bioinformatics such as gene expression networks [174] and molecular signature classification [152].

Discovering the structure, without any limitations produces General Bayesian Networks (GBN) [24, 26, 58, 81]. These structures are totally open, as long as they conform to a DAG structure. This generality is beneficial for causal discovery as the relationships between evidence may indicate correlations and causalities initially unknown to the expert. Without any other limits, a GBN may find many relationships missed in the more strictly structured Bayesian Networks. The lack of assumptions may also improve performance for applications in domains where the system has a high amount of correlation and causal relationships that can be leveraged for classification. The opposite side to this information is that these relationships are merely estimates from data, and thus subject to over generalization, such as the direction of a link between two variables. Over generalization can be partially mitigated by the expert analyzing the graph without directionality, which focuses on the potential relationship between two vertices.

GBN algorithms are search based, and the construction of the network can now be

any link that does not violate the DAG requirement. Greedy search algorithms that are iterative, such as K2 and Simulated Annealing [63], are the most straightforward approach. These algorithms evaluate the structure after every link, using the estimate of the likelihood that the structure would produce the data. This continues until no changes improve the likelihood. While K2 and Simulated Annealing algorithms are global algorithms (any link can be added at any time), other algorithms focus on the markov blanket and discover a single node at a time; examples include Iterative Local Search [63] and Max-Min Hill Climbing [159]. Another approach is to use genetic algorithms that iterate from several random networks that eventually converge on an optimal network [88]. These structures and algorithms face the same challenges as limited structure algorithms, where large feature spaces and large amounts of data can make the search process a time-intensive process. The search algorithms of GBNs have been shown to be NP-Complete [25].

The structures for these Bayesian Networks are agnostic of temporal effects. Explicitly finding causal relationships between evidence that exists from time n to time n+1 creates structures known as Dynamic Bayesian Networks (DBN) [115]. The structure is often limited as a first order Markov process requiring temporal relationships to exist at no more than one unit apart, i.e., no evidence at time n can relate directly to evidence beyond the time n+1 step. Graphically, a DBN is represented as a time slice, with two copies of a standard BN (time n, and time n+1) and temporal links that connect between two vertices, one at time n, to another at time n+1. Algorithms for learning these structures are much more complex, as they search for relationship in time, and across time (the Markov process limitation helps keep these algorithms from becoming even more complex). While the algorithms continue to use a general search algorithm pattern, they incorporate techniques such as Monte Carlo sampling [54] to efficiently measure the likelihood of the current structure [115]. DBNs have been used for Fault Detection and Isolation [137] as well as bioinformatics research [78].

Along with the ability to incorporate missing evidence, the inference of these networks

handle hidden variables. Latent variables are commonly used to model the state of a system, when that state cannot be measured. The Hidden Markov Model (HMM) [12] combines both hidden variables, as well as a markov process system, to model a changing of a state based on a single observation. Hidden variables can be added to DBNs as well as GBNs, although at the cost of building a model (the expert must already know the general relationships for the hidden variable) and increasing the complexity of reasoning with the model. HMMs have been used for speech [163] and text classification [79].

After these structures are constructed, the last step of a Bayesian Learning Algorithm is to build the conditional distributions for the different vertices in the structures from the data. This step, known as parameter estimation and can be done parametrically, or non-parametrically depending on the type of data used to model the system. The choice of probability distributions can be either discrete or continuous. Discrete probability distributions require the features to be discrete valued, or discretized during pre-processing. These distributions are represented as tables, meaning that for a given vertex and parents in the structure, the table is defined for a cartesian product of the range of the parents, and each distinct value for the selected vertex. The probabilities for the table are defined using a counting algorithm, which starts by selecting the appropriate data for each cell in the table, counts the instances, and divides by the total. The probabilities found through counting the data may be modified by the use of a-priori estimates of the distributions, which are useful if the expert believes the data does not have enough information to accurately estimate the conditional probability for certain combinations of evidence.

When the data is not discrete, or the distributions are better modeled as continuous, the tables are replaced by continuous conditional probability distributions. Estimating these distributions is similar to the counting algorithm where the data for a given vertex is selected with the parents in mind, and then instead of counting to build the tables, the parameters that define the distribution are estimated from the data [118]. Further work has taken kernel-based, non-parametric approaches and shows great flexibility with respect to

the data [72]. The choice for which parameter estimation to utilize is not always clear, as discretization of continuous data can achieve better results. The choice of estimation could involve empirical comparisons of the respective accuracies, or the desire of an expert (based on the application) for one estimation over the other.

Generative algorithms apply to all three types of supervision problems, and thus to different kinds of anomaly detection. Among the advantages of these algorithms include the power to look for unknown structure in the data, to accommodate different probability distributions that infuse models with a variety of rich behaviors, and to produce models, which may be useful in understanding the differences between normal behavior and abnormal behavior.

II.2.3 Supervised Anomaly Detection

Supervised detection involves the explicit belief in either a-priori knowledge or previous observations of anomalies. Using supervised detection involves knowing with a high degree of certainty where in the data an instance or pattern is considered to be expected or nominal, versus other segments that indicate an aberration. This knowledge is used in the form of labels that are applied to the data. The primary focus of models for this application are not on discovering anomalies, but being able to discriminate or distinguish between anomalous and nominal situations, or among different types of (known) anomalous situations. This sort of detection is beneficial at the beginning of modeling a physical system where an expert is looking to improve a model that contains gaps because of incomplete knowledge.

Depending on the nature of the data and the target application, the models can incorporate many different attributes or features. The model derivation process can be generative or discriminative depending on how much of the environment the application needs to model and the design requirements for computation time given the size of the system being modeled. The models produced from the training data with these labels are also used to

find which features are most important for differentiating the nominal from the anomalous and also between different anomalous groups. The ability to use different modeling techniques to understand the relationship between the features and the different classes makes supervised anomaly detection suitable as an expert-in-the-loop approach.

The use of discriminative models, such as decision trees have been used to build supervised models for different types of attacks in network intrusion [40, 145, 154]. The network intrusion domain has the benefit of involving a limited number of attack types with varying implementations. Supervised learning will help differentiate the different approaches to the same attack type, and in a Decision tree, this structure can be relearned to take advantage of the new information. Decision trees will produce a structure semantically interesting to an expert for further analysis of the system.

Neural networks provide an alternative as a discriminant function model. In intrusion detection, Artificial Neural Networks may be used to classify the user behavior, instead of classifying attacks. Training on the behavior, the output of the model (the predicted user) is compared to the current user to decide if an intrusion is in progress [140]. As mentioned above, Artificial Neural Networks may be fast to learn and accurate, but as a black box the model will be unable to directly assist the expert in discovering new knowledge. Aside from the behavior model, the Artificial Neural Network is also a popular technique for learning models at different areas in the network (such as specific nodes) [145]. The Artificial Neural Networks may be used in conjunction with Decision Trees to utilize the benefits of the Artificial Neural Networks and the decision trees help decide how to compare the output of the Artificial Neural Network with the known user [120]. Artificial Neural Networks have also been used in diverse applications where the expert may not need to inspect the model, such as distinguishing between magnetic signatures to detect land mines [150], detecting anomalous trajectories for vehicles for security [20] and the prototypical credit card fraud application [23].

Support vector machines have also been used in the ubiquitous intrusion detection genre

of anomaly detection [67, 112]. The SVM approaches provide a model that uses discriminant functions, but also is built to optimize the generality of the function. Optimizing the function for generality improves robustness for new instances generated by the system. However, unlike decision trees, which may pruned for generality, there is minimal interpretability for the SVM model when discovering new information in these applications. The SVM has also been used for other applications, such as hyper-spectral anomaly detection [1] and credit card fraud [64, 169].

The use of supervised generative techniques, such as Bayesian networks have also been used for anomaly detection [173]. A number of different diagnosis applications use Bayesian networks as a supervised anomaly detection approach. As data is collected in systems for diagnostic analysis, the data may be annotated at runtime with labels that help refine the fault detection models [34]. When such methods are implemented as Bayesian networks, the generative model may not only improve detection of failures, but also allow an expert to understand what sensors in the network cluster are the best for discovering these known failures.

Extending this use of Bayesian networks to address knowledge engineering in the anomaly detection models is used to improve previous models and inform the expert about the nature of the fault. This example also motivates the application of not only differentiating between normal operation and faulty operation, but also looks at how you can isolate different causes for adverse events. Not all failures in a system are the same, and while detecting that a system is starting to fail is important, there are times when establishing the difference between known failures is a critical component. This research has already produced work to show this advantage of supervised anomaly detection [99, 100, 101].

Models built from labeled, sequential data using Bayesian learning has shown promise in the areas of diagnosis [103]. Allowing the model to compare a likely sequence with the data can be used to identify anomalous signals and catch failures in the system.

II.2.4 Unsupervised Anomaly Detection

Unsupervised methods are employed when we have no initial knowledge or do not have reliable knowledge how to differentiate between nominal and anomalous behavior. This problem becomes even more significant when the data is high dimensional, making it hard for human experts to define precise classification labels or propose analytic methods for differentiating between nominal and anomalous data. In such situations, very little pre-knowledge about the data is assumed, and unbiased algorithms are employed to segment the overall data sets into groups, such that objects within a group are more similar to each other than objects across groups. A heuristic that is often employed in anomaly detection is to consider groups that contain a large percentage of the data objects as defining nominal behavior, whereas the data objects that fall into smaller groups or fail to be labeled in any of the other groups (outliers) to be anomalous. A number of generative modeling techniques may be employed to produce the nominal models. These techniques find an inherent structure to the data, using non-parametric algorithms that are distance or similarity-based and parametric algorithms that can be density-based or expectation maximization based Bayesian methods.

Unsupervised detection methods will utilize the model output differently depending on whether it exists as a Bayesian model of the evidence or through a number of clusters and cluster affiliations. Described in Section II.2.2.1, the types of clusters and affiliations of the instances in the data provide a variety of uses to the overall detection scheme. The easiest use of cluster output is to produce initial identifications of the data that are used as initial labels to produce a dataset for building models with supervised (and semi-supervised) techniques. Initial labeling from clustering such as K-means has been used with techniques such as decision trees for chains of algorithms for anomaly detection [53]. For an expert, the use of clustering for this purpose is mainly as a pre-processing technique.

The use of clustering can also be to reject training data immediately, by discovering the nominal behaviors and the deviations from them. Depending on the hard and soft, strict

and loose partitioning of cluster algorithms described in Section II.2.2.1, the results for the outliers can be used differently. K-means clustering will find anomalous points as groups to develop common signatures for the groups marked as anomalies [7]. These signatures are used in aviation safety domains to build fault signatures for sets of aircraft sensors to find anomalous flights. Other examples of using clustering to find and label data include intrusion detection [128] where the signatures of different attacks are discovered instead of built by experts.

When density based clustering techniques are used, the goal is to discover lingering anomalies that exist separately from any groups of behavior [94]. Unlike k-means, these outliers do not provide signatures, but provide examples in the data (such as aircraft) to examine for abnormal behavior. Density based clustering, depending on the method (such as Gaussian probability density function), allows for unusual shapes (spheres for the Gaussian pdf) to the nominal clusters, but the expert must also be vigilant that clusters are not made up of instances of similar abnormal behavior that crosses the parameterized threshold for cluster creation.

Hierarchical clustering using a cutoff at a high-level will produce small numbers of flattened clusters. These flattened clusters have been used to find large groups of nominal behaviors and small groups of anomalous behaviors in applications, such as vehicle trajectory classifications [52]. The use of hierarchical clustering also allows the expert to subdivide anomalous groups for further analysis. While this ability to delve deeper in the construction of the cluster may help the expert, finding the appropriate cutoff where nominal clusters are not yet associated with abnormal clusters is non-trivial, and may involve manual input from the expert after consulting the dendrogram.

Finally, using mixture of Gaussian clustering to detect anomalies is found in applications, such as multi-spectral image applications [62]. The soft partitioning makes this clustering useful for environments where data objects are distributed so that small numbers of features (compared to the whole) indicate the anomalies, but the number of objects with

these feature values are minimal. Finding these clusters in data that will be overwhelmingly normal, can make their discovery difficult. Mixtures of Gaussians can be used to model the entire distributions over the data to discover these anomalies. Extensions can be used in conjunction with supervised techniques, such as Artificial Neural Networks to help identify abnormal patters in sea traffic [89].

The example of density based clustering for anomaly detection was used in conjunction with feature reduction by Principal Component Analysis (PCA) [135]. The PCA was used to reduce the dimensionality of the data, and build features that are orthogonal and cluster better. Feature space reduction can apply to different types of clusters. PCA reduction can also be used with other unsupervised methods, such as distribution testing to define general probabilistic neighborhoods of expected activity [85]. The testing will identify instances in the high-variance Eigen-space that are in the tail and thus anomalous, or outside the low-variance Eigen-space and therefore do not fit the distribution of the data at all.

When generative models such as Bayesian nets are used for unsupervised learning, the structure itself can operate as a general classifier as well as use new instances to grow and augment the structure to deal with ever changing information. An example of using generative models for anomaly detection is to classify whether vehicles paths as abnormal. This may be useful for the purpose of understanding potential security risks. This application requires looking at the general structure between expected paths and then examine the instances that do not conform to this example. Once this structure is found, it can be leveraged to produce supervised structures that can form models on the attributes of the path and produce a model that possesses interpretable properties about these anomalies.[105]

Other methods in the unsupervised realm include sequence mining [121, 175], which look to find common subsequences in separate instances of the dataset. These algorithms look for statistical support that can indicate when the different sequences are significant in the data. Sequence mining has been used for unsupervised anomaly detection of aircraft anomalies [17]. Often used in environments where the data is made up of symbolic

sequences, more complex sequences that use numerical data may require complexity analysis [16, 77] to find anomalies inside the signal.

II.2.5 Semi-Supervised Anomaly Detection

Semi-supervised methods answer the issues in both supervised and unsupervised methods. Acquiring a fully or even mostly labeled dataset of both nominal and anomalous data object is unlikely. In most cases, only the number of nominal data points is sufficient to build reliable models, whereas the number of anomalous data points may be too few to generate reliable anomalous models. Therefore, the first step in semi-supervised anomaly detection may be to generate nominal models from nominal data, and compare new data objects against the nominal models. A good match implies that the new data object may be labeled as nominal, otherwise the data object is anomalous, and a candidate for further scrutiny and analysis. Therefore, semi-supervised learning will label the sample as one class (nominal), or as "everything else", reducing the error by not over-classifying the anomaly (although misclassification as nominal is still possible). Unsupervised methods; however, may not build nominal models that are specific enough, instead, as systems evolve, so too the model shifts, producing an ever changing decision space of what constitutes nominal. Since these models need to be applied for general systems, not just the systems in operation, an unsupervised model may be too forgiving of what constitutes nominal. In contrast, semi-supervised models may grow to be outdated for a specific environment, but the experts will discover decaying performance, and will be able to retrain the model for a new environment. In essence, when most of the operations are nominal and identified as such by either the system, the expert, or though the use of unsupervised techniques, semi-supervised learning is useful for building the models of this behavior and using this model to classify new data as nominal and anomalous.

The one-class SVM is one of the most popular techniques for semi-supervised anomaly

detection and has found use in diverse fields of anomaly detection such as diagnosis in aircraft [36, 37], discovery of land mines [117], business applications for churn models [177] and like so many others, network intrusion detection [124, 158]. The one-class SVM is an extension of the SVM. The extension optimizes the classifier for a single class label. This optimization for a single label constructs a general decision boundary for the training data to build a model that can accurately discriminate data with this label. This technique, like its original construction suffers from limited information for the expert, and given a kernel transformation, it produces even less information. In the presence of a noisy training set, the decision boundary may be poor, and flag more anomalies than actually exist.

Other methods that are less popular include the use of decision theoretic methods for applications like Fraud detection [146] in financial accounting and network intrusion [87]. Decision-theoretic methods are useful in the decision space of one class, where the structures for the classifier are built to isolate the single class. Unlike one-class SVMs, these methods are more open to knowledge engineering tasks due to their openness. Disadvantages of decision-theoretic methods include being more time-consuming to build and potentially more brittle without a representative dataset.

Semi-supervised learning for anomaly detection can involve generative models, such as mixture models typically for network intrusion [168]. Generative models use the entire probability distribution from the data to determine probabilistically if an instance is either in the known class, or not. Other generative model for anomaly detection involve structures, such as Bayesian Networks that have been used to classify failures in computer equipment, such as hard disks [60]. Like decision theoretic methods, the Bayesian network is also much easier to apply for knowledge engineering, but also be computationally more intensive than the one-class SVM.

CHAPTER III

RESEARCH APPROACH

The primary research problems being addressed in this thesis are the identification and early detection of failures (anomalies) in complex systems through the use of data mining and machine learning techniques that apply to big data. These systems overall behavior across time is captured by large multivariate time series data. The two general approaches to this problem are to reduce the data through either:

1. Restricting the scope of the data (both in samples and features) with expert knowledge to solve specific, constrained problems.

2. Dimensionality reduction techniques that manage the size of the data while maintaining critical information to solve a more general class of problems in an efficient manner.

Both approaches produce transformed data for performing fault and anomaly detection and building models for early detection of these anomalies.

This research makes important contributions to the fields of data mining, anomaly detection and knowledge engineering. The first approach combines knowledge engineering, data curation, and supervised learning schemes, to establish a method for combining existing expert knowledge with new information derived from classifiers to improve accuracy and early detection of known faults. The use of a Bayesian representation structure provides the seamless link between the expert's knowledge structures and the classifier-derived knowledge, by creating additional associations between existing monitors and fault hypotheses, deriving new monitors that combine old monitors, as well as probabilistic information that is used to rank potential fault hypotheses.

The second approach investigates the use of complexity measures based on compression, information theory, and signal analysis to perform dimensionality reduction on the large amounts of multidimensional time series data. The approach is designed to make as few assumptions as possible about the nature of the features and their importance relative to one another for anomaly detection, other than the fact that they represent temporal (or ordered) sequences of data. A research contribution for this problem is the dimensionality reduction approach that preserves important temporal properties of individual features, but the reduction produces dissimilarity matrices that allow for traditional unsupervised learning (clustering) methods to be applied to large data to characterize nominal and anomalous data objects, and then utilize feature selection to aid the expert in understanding the nature of the anomaly. This approach extracts useful information for domain experts to define new models to support online detection in future applications. To demonstrate the feasibility and effectiveness of this approach, we apply this methodology to a large (0.7 TB) sized flight data set, as well as a second data set that involves studying the mechanics of pitchers' throwing motions in baseball to isolate anomalous incidents. In case of the airline data, this may be correspond to faults in the equipment or pilot errors, and in the pitcher data, a change in throwing styles may be a precursor to pitcher injury or an indicator of a great performance.

This chapter is outlined as follows. First, Section III.1 examines the genesis of the problems in data from complex systems. Section III.2 provides a description of the anomaly detection tasks in complex systems and the problems we encounter due to the data. The structure of the data and the domains we focus on in this research are defined in Section III.3. We provide the details of our approaches to the problems, and our contributions in Section III.4. Lastly, we summarize this chapter in Section III.5.

III.1 Nature of the Data

When employing data-driven approaches to anomaly detection in complex physical systems, a primary challenge is the effective management of increasing amounts of data collected during the operation of these systems. These challenges include how to collect, organize, and access the data in unbiased ways so that they can be used to provide answers to problems in an effective and efficient way.

Automation in subsystems has led to an increase in computer-based control, thus resulting in more sensors and actuators and the ability to collect more measurements of system behavior. As control algorithms become more sophisticated, and sensor technology has become cheaper and more flexible, the rate and quantity of data collected has increased by significant amounts. Collectively, this implies that much larger amounts of temporal, i.e., time series, data is being collected during system operations, which opens up doors for more precise and accurate post hoc analysis of system behavior.

Given the complexities of present day systems, the larger amounts of data provide opportunities for more detailed analysis of nominal and anomalous behaviors of these systems. Systems like aircraft have many interdependent components, and overall analyses of system behavior requires use of advanced composition and causal analysis mechanisms to understand how individual subsystems and components contribute to overall behavior. Systems also operate in more diverse environments, requiring the need for including more contextual attributes in the analysis schemes. Some of the complex systems, such as pitcher's throwing motions in baseball go beyond the complex nature of pure physical processes. Human behavior is influenced by a number of physical traits, some governed by the physical state and conditioning of the pitcher, and some by their inherent traits and makeup. The large space and set of factors that define human behavior can make tracking anomalies more difficult, since the nominal behavior itself is can vary and depends on a number of interdependent factors.

III.2 Problem Description

The large amounts, the variety, and the complex nature of the data makes transforming and utilizing this data for anomaly detection a very challenging task. This raw data includes many different features, not all of which may be relevant to solving particular problems, such as the anomaly detection problem. A number of the issues that have to be dealt with before machine learning algorithms can be applied to analyzing the data are described next.

III.2.1 Task 1: Data Curation

Data curation is an essential first step in producing effective anomaly detection models for complex systems. For anomaly detection tasks, the organization of the data and metadata is critical to accurately grouping contextual attributes and making sense of the results of the models.

The raw data collected from these systems may be stored across a large number of mediums. For example, in the airline data used in this research, the flights for a specific aircraft were originally stored on hundreds of CD-ROMs of varying data integrity. There is a challenge collecting and organizing this data into a database resource to facilitate easy retrieval to solve a variety of problems. The process of collecting this data can be arduous, requiring methods that retrieve relevant segments from a variety of resources, and align and synchronize them while organizing them into a centralized location. Storing the data in a manner that increases flexibility of retrieval reduces the amount of time required to revisit the original data stores and repeat the previous tedious tasks of retrieval and organization.

III.2.2 Task 2: Data Transformation

Given that the data has been collected and stored, the next step is extracting the necessary data, and transforming it to a form that facilitates the data mining tasks. This transformation can be further split into two subtasks. The first is the process of finding relevant data for supervised learning. The second task is taking the extracted raw data and deriving

features that are structurally efficient for use with a variety of machine learning algorithm when the datasets are quite large.

III.2.2.1 Selection of Data Relevant for Supervised Learning

Our overall goal is use data-driven approaches from which information may be provided to build or enhance models that help detect and identify anomalies in system behavior. Depending on the situation and the approach chosen, these models may be constructed using supervised learning algorithms. In other words, these learning methods assume some amount of labeling is available for the data.

It is uncommon for data recorded by automated systems on equipment to record labels that establish whether the recorded data is nominal or anomalous. Therefore, building models from the data that help isolate anomalous situations and faulty components using machine learning techniques becomes a difficult task. This lack of accurate labels (both for nominal and specific failures) means that the data is by itself unsuitable for building models that utilize supervised methods.

III.2.2.2 Extracting Features for Analysis

The second task in data transformation is extracting features from the raw data. Each feature may correspond to one or more sensors that record the measurements over time, and each of these temporal measurements are typically sampled at sub-second rates such as 4Hz, 8Hz and 16Hz for the duration of system operation. The data set in this form, with each data object represented by multiple features, and each feature being made up of multiple data points does not lend itself to analysis by standard supervised and non supervised algorithms. Therefore, the data has to be reduced to a more compact and meaningful representation without compromising the features in the data that are important for anomaly detection.

The challenge is to find the appropriate dimensionality reduction approach that takes

multivariate time series data, and transforms it into a representation that applies to a wide range of supervised, and unsupervised learning methods. An additional challenge is to find the appropriate method that is both accurate and efficient for this comparison.

III.2.3 Task 3: Supervised Anomaly Detection

With a labeled and transformed data supervised learning algorithms can be applied to build classifier models that can isolate specific anomalies, such as a component fault in a system. In our research, supervised learning methods have been used to address the knowledge engineering task of finding additional relations that enhance a current diagnoser for the system. The knowledge engineering task is often mediated by human experts, and the supervised learning algorithms provide additional information to the experts to help them augment the diagnostic models. This means that the derived models should be interpretable by experts.

This is a challenge due to the diverse number of features including many that are simply not germane to the current anomaly detection task. For knowledge engineering tasks that involve anomaly detection, the use of extraneous features can interfere with an expert's ability to understand the data-driven models. Therefore, model building, task must be designed to mitigate the negative effects of extraneous features.

III.2.4 Task 4: Unsupervised Anomaly Detection

Transformed data without labels, or with a limited set of labels for some of the nominal behavior, requires the use of unsupervised anomaly detection.

The large number of data instances makes the use of unsupervised learning more difficult. The use of a clustering algorithm produces a small number of clusters containing most of the data. A rule of thumb or heuristic used in these situations is to assume that the smaller clusters may represent anomalous situations, and are worthy of further analysis by comparing them to the large, nominal clusters. The method applied for doing this may

be labeled as a feature selection algorithm. Depending on the feature extraction method employed in Task 2, this challenge may be made more or less difficult.

Another challenge in the analysis is that the nominal models may not be very compact and include a variety of different behaviors. In that case, the anomaly detection has to be carefully designed to take into account this diversification, without significantly increasing the false positives in the data. For example, when dealing with humans as the systems producing the data, this challenge becomes intricate. Compared to a mechanical system built according to a well-known specification, humans possess greater variance from one sample to the next. Data collected during human operation can be diverse, making the modeling of nominal behavior difficult, and requiring the model to be more forgiving when applying anomaly detection to a new participant.

III.3 Problem Domains

The two domains that we study in this thesis:

1. aircraft flight data, and

2. Major League Baseball pitcher data,

provide a unique set of challenges for anomaly detection. In each, we are looking to discover and understand the anomalies that occur during operation.

In the flight domain, our goal is to first understand and better model known physical failures, using knowledge of the their occurrence and supervised learning methods. The models then are used by experts to find improvements to diagnostic systems on board the aircraft. A second goal is to build models suited for discovering previously undiscovered anomalies at takeoff. These anomalies and models can be used to help an expert identify new types of faults and features to classify them for the diagnostic model.

Using the baseball domain, we want to look for anomalies in the pitches that a pitcher throws in a game compared to the his normal pitching motion. Similar to the flight domain,

we want to use unsupervised techniques to look for anomalies and build models for what constitutes nominal behavior. In contrast to the flight data, the novelty of pitcher data is reflected in the challenges of Task 4 above, specifically dealing with human produced data, which may have a wide definition of nominal behavior.

III.3.1 Describing the Raw Data as a Data Cube

The aircraft flight data and the baseball pitcher data have similar characteristics. Each instance in the data is a multivariate collection of a time series. Specifically, each instance is made up of M features, and each feature is a time series that goes from time 1 to T_m, where m can be different for each feature that is defined by the sampling rate at which the data is collected. For simplicity, we assume that while the signals may be different lengths, they represent the same amount of time across all features and all instances. Without this assumption, the data is difficult to interpret, since signals could represent different periods of time. The uniform length in temporal units from instance to instance, allows for a uniform transformation.

We refer to this notion, illustrated by Figure 3, as the "data cube."[1] The cube is represented by N instances (each instance is labeled d) each with M features, each with a length of T_m time series samples. This cube represents the problems encountered in the subtask for feature extraction in Task 2, and the choices for feature extraction impact Tasks 3 and 4. Most learning algorithms used for anomaly detection operate in two dimensions: features and instances such as a Support Vector Machines, or an instance and a univariate time series such as a Linear Dynamical System. Finding methods that will efficiently collapse one of the dimensions is imperative to making the anomaly detection task manageable, which is an important problem that we solve in this research.

[1] This is a misnomer, as the the shape of the data is more unique due to the varying lengths of features (an uneven third dimension), data cube is used as an abstract term.

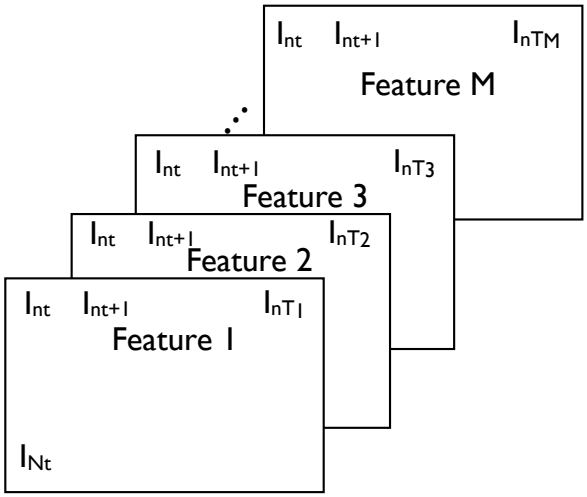

Figure 3: Data Cube Representation

III.3.2 Aircraft Flight Systems

Aircraft flight system data represents a canonical example of large operational data for anomaly detection, i.e., to discover a variety of faulty situations that can be attributed to the aircraft, unusual environmental conditions in which the aircraft operates, and pilot errors during flight. As aircraft are becoming more sophisticated, data collected from the aircraft includes a large number of sensors that are recorded at high sampling rates, and in systems that are increasingly regulated by digital controllers. Early detection of anomalies in aircraft directly addresses aviation safety matters. Diagnostic systems are already in place on modern aircraft that model the system and attempt to detect, mitigate, and respond to safety matters as quickly as possible. Since these models are incomplete for a variety of reasons, such as a lack of expert knowledge or new technologies on board the aircraft, we desire to improve these models through the use of data-driven techniques. First, we would like to improve detection of already known failures through supervised learning and knowledge engineering. Secondly, we would like to use the data to identify new anomalies, specifically during the takeoff phase of a flight. From these anomalies, the goal is to produce information about detection so that an expert can add new models of these phenomenon to the diagnostic reasoner.

The data that we use in this thesis was provided by Honeywell Aerospace and was recorded from a former regional airline that operated a fleet of 4-engine aircraft, primarily in the Midwest region of the United States. Each plane in the fleet flew approximately 5 flights a day for 5 years. This produced over 25,000 flights. Since the airline was a regional carrier, most flights durations were between 30 and 90 minutes. For each flight, 182 features were recorded at sample rates that varied from 1Hz to 16Hz. The data from these flights has been anonymized for research purposes but maintains the sensor information and a modest amount of location information such as departure and arrival airports.

When examining this data in the context of the data cube, the instances are flights. The number of features in the cube is the number of sensors being considered from the total set of 182. For the problem of detecting anomalies during takeoff, the time series will focus on the short amount of time when the aircraft has left the ground, and before it has started a controlled ascent. The sampling rates of the variables vary, making the length of the time series as long as the operational time in seconds or a multiple of that length with higher rates such as 4Hz, 8Hz and 16Hz. This cube is very large in terms of instances, and features and in terms of the different operation times. When we limit the time series to only takeoffs, this results in features that are around 30 seconds in length per instance, but over 100 samples long with the features using higher sampling rates.

III.3.3 Analyzing Pitcher Performance

The domain of pitchers from Major League Baseball represents an area of increasing interest from the research community. Similar to the approach for aircraft systems, our goal is to identify novel anomalies for a pitcher, based on the way they pitch from game to game. Identifying these anomalies as ones that correlate to potential injuries and above average performance are crucial in improving the understanding of conditioning and pitching mechanics of these athletes. After finding these anomalies, our goal is to produce new models for detecting them in other pitchers. Considering the similarities of these goals

with those in the aircraft flight systems, the domain of pitchers in Major Leagues Baseball presents contrasting issues about large data and anomaly detection that makes it novel for this research. The varying nature of mechanics for throwing the ball coupled with the range of human body types for baseball pitchers makes this large data set extremely diverse. This diversity makes anomaly detection through the use of clustering and model building a more challenging problem, since there are more types of nominal behaviors. The anomaly detection challenge here is to find reliable indicators from diverse data that identify problems leading to pitcher injuries.

The general form of this data is collected through the use of web-scripting and parsing of XML files from Major League Baseball Advanced Media or MLBAM. MLBAM oversees the devices used to collect data during the game. These devices known as the Pitch f/x system, are a pair of two cameras in the stadium, each calibrated for the baseball park's location and height of the mound from its position. These Cameras record the pitcher's movements as well as the baseball, and uses imaging algorithms to measure information about the pitch. This information includes:

- The location on a 2D axis projected by the plane where it leaves the pitchers hand.
- The Speed of the ball when the pitcher releases the pitch.
- The spin rate of the baseball as it moves through the air.
- The location and speed when the ball crosses a projected plane half way across home plate.

Each pitch is also annotated with the type of pitch thrown (fastball, changeup, slider, etc.), game information such as the score, inning, any runners on base, and the result of the pitch (was the pitch a strike or a ball, did the batter swing, and if so, what happened on the play). These are then stored on-line for visualization and for use by fans and researchers.

For example, we may discover by looking at the data that a given pitch was a four-seam fastball, thrown from a side arm position relative to the pitcher, and left the pitcher's hand

at 95MPH. Further, we can identify that this pitch broke downward halfway between when it was released and crossed home plate. Finally, we know that the pitch arrived at home plate at 92MPH, and was located in the upper right of the strike zone to a right handed batter who swung and missed, for a third strike that caused the third out, which ended the inning. This type of information has been recorded for every pitch thrown since 2008.[2].

The data used in this research is provided by Harry Pavlidis and his company, Pitch Info LLC of Chicago, Illinois. The data from Pitch Info is the same data as MLBAM with a few improvements. It has been curated into a database for easier dissection, and annotated with more accurate strike zone information, and pitch type classification. This data is unlabeled with respect to injury information.

This data is large instance-wise, as it is collected over every game-active pitcher every day during the season (and for pitchers in the post-season playoffs). When examining this data in the form of the data cube, we consider an instance to be a game per pitcher, which is the data for a pitcher for a single game. For example, if a pitcher threw pitches in 30 games a season for 10 seasons, then he would have 300 instances in the data cube.

The features in the cube are data that could be recorded for a given pitch-type thrown during the game, such as the starting location on the y-axis for all fastballs, the starting speed for all fastballs, etc. This produces 7 features for each of the 6 types of pitch types thrown, therefore, 42 features per instance. The time series aspect in this case for each feature is recorded for each pitch thrown. Much like the different sampling rates in the aircraft, a pitcher's chosen pitch types will make some signals longer than others. As above, where we use takeoffs to constrain the data, we chose only starting pitchers, and only games where those pitchers threw at least 100 pitches total. We consider this a "routine" start that involved natural fatigue for the pitcher. This restriction allows the signals to be longer and more uniform. In the pitcher domain, the time series themselves are likely to be smaller

[2]The system was in place in 2007, but due to its experimental nature, that data is often ignored because of noise and incompleteness

per instance than the flights in the data cube. This adds to the diversity of our problem domains.

III.4 Research Problems

Together, these two domains and their data illustrate the general problems encountered in this work. Our approaches to these challenges and the problems we solve form the contributions to the fields of knowledge engineering, diagnosis, data mining, and anomaly detection.

III.4.1 Supervised Learning Methods to Support Knowledge Engineering for Diagnosis

A primary goal is to provide experts with models of anomaly detection derived from flight data that can be easily integrated into existing diagnostic reasoners. This approach, supports a knowledge engineering task, and begins from curation of the data, to the choice of models for anomaly detection, to the process of implementing suggested improvements to the diagnostic reference model. We use the aircraft systems data described earlier for this task.

Our approach in this contribution first superimposes layers of expert information to aid in the curation of the data. The expert information used for curation of the data includes a Federal Airline Administration database of aircraft incidents to facilitate the labeling of data into specific faults and nominal behavior, as well as an expert-built list of features for feature extraction. These features are values that represent the conditions of systems on the aircraft during different phases of operation. Together, this information is applied to the flight database and produces a labeled and transformed dataset for use with supervised learning techniques.

The approach for the supervised learning of this data is to build models that provide new information about the nature of the system during a failure. These models are meant

to be both interpretable by the expert and rich enough to incorporate new information mined from the data. We utilize Bayesian structures to model this information. Next, we produce a framework for taking the data-driven models and incorporating the new information as additions to the diagnostic system. Lastly, our approach provides a test to validate the improvements to the diagnostic system.

This work produces research contributions to the fields of diagnosis, data mining and knowledge engineering. The contributions are centered on the creation of a framework for improving the accuracy of expert-based models of diagnosis and detection for vehicle based reasoners. This is presented as a data mining induction method, detailing the creation and application of these techniques with industry based models and aircraft data, and culminating in case studies to show the validity of the approach.

III.4.2 Unsupervised Learning Methods to Support Anomaly Detection for Multivariate Time Series Data

The first approach uses expert information in a layered fashion to constrain the data. This approach develops a method for discovering anomalies in large data using an unsupervised, exploratory approach.

This approach first focuses on the feature extraction task to reduce the dimensionality of the data domains, and produce data that is efficient to use in building anomaly detection models. We explore a range of techniques involving compression, information theory, and signal analysis for reducing the time series dimension to a single value. This reduction leaves only the instance and features as the two dimensions of the new dataset. The approach looks at these techniques and their effectiveness. This is accomplished through the use of experiments that test the nature of the different dimensionality techniques in the case of identifying anomalies. These experiments range from controlled signals, to the use of a real world test set.

Once we have selected the dimensionality techniques based on empirical results, we

revisit the aircraft systems domain to look for anomalies that occur during takeoff. After feature extraction and transformation of the aircraft data using our selected methods, we use unsupervised techniques to cluster the the data and look for anomalous instances. Expert knowledge is used during this exploration to isolate anomalies that represent serious events such as a safety hazard. Through feature selection, our approach isolates the sensors in the aircraft which are most likely to identify these anomalies, helping the expert to further improve their diagnostic models.

Lastly, we turn our attention to the pitcher data, and apply our unsupervised, exploratory approach to a second domain. The approach is very similar, with feature reduction being applied to the time series portion of the data cube, and exploratory techniques being applied to this reduced data to identify anomalies. The approach differs from the first domain in that more organization of the transformed data must be done, to account for the diversity in the different pitchers. The goal is still the same, to identify relevant anomalies and build models that can both aid an expert in identifying the warning signs of these anomalies.

The contributions of this work are in the fields of data mining and knowledge engineering. Our experiments to test the different dimensionality reduction techniques set a baseline for the future comparison of such techniques as they apply to reducing the time series dimension of complex data. Our applications to aviation safety and baseball show the ability of these techniques in diverse domains with similar objectives. In both cases, the application is designed to produce new, interpretable information to the expert for creating better models. These contributions are explored in terms of case studies and analysis of the exploratory methods.

III.5 Summary

This chapter gives an overview of the general problem, data, and research methodologies of our work. We describe the systems and issues that make data large and unwieldy for anomaly detection. We described briefly the data domains that focus on these issues

in our research. Lastly, we describe in brief our approaches to these problems and the contributions of this work. In the next chapter, we begin a more detailed exploration of our approaches, and start with supervised learning to support knowledge engineering for diagnosis.

CHAPTER IV

IMPROVING DIAGNOSTIC REFERENCE MODELS

In this chapter, we detail our supervised anomaly detection approach to support the knowledge engineering task for diagnosis. When constructing diagnostic systems, models are often first built manually by experts, crafted from physics and engineering knowledge of the particular system, and augmented with expert experience from observations of the construction and use of similar systems. When engineers modify a system's specification or redesign a component, the original expert knowledge may have to be updated to accommodate the changes. The goal of our supervised data mining approach is to help experts improve and revise these models using data from the measurements of already running systems. This work addresses several research and logistic issues related to data-driven approaches for knowledge engineering:

- How can the raw operational data be transformed systematically into a curated dataset for building data-driven models?
- How can models be produced from operational data to accurately detect faults and improve detection time?
- How can data-driven models be used to provide insight about the system and transfer information to improve the original models used by experts for diagnosis?

These issues are critical in aviation safety, where the early detection and mitigation of potential adverse events caused by system or component failures can prevent aircraft damage and loss of life.

Researchers and domain experts face challenges in the data mining and knowledge engineering tasks due to the nature of complex systems. For example, the degradation and faults

in one component may cascade to other components during flight operations. As a result, multiple sensors spread across the system may report anomalous or faulty behaviors; consequently, combining this sensor information to detect and isolate faults in a timely manner becomes a difficult task. Aircraft Diagnostic and Maintenance Systems [151] use (1) a system reference model that describes causal relations between potential faults in aircraft components and sensor readings and (2) reasoning software that combines abductive [127] and Naive Bayesian reasoning [81] methods to infer and rank potential fault hypotheses. A widely used Aircraft Diagnostic and Maintenance Systems in operation today is the Boeing 777 Central Maintenance System [5].

A benefit of separating the reference model from the reasoner software is it allows subsystem manufacturers to encode proprietary fault models for individual subsystems in the reference models. The system integrator, the aircraft manufacturer, designs the integrated solution that combines information from the subsystem reasoners to make global diagnostic inferences [66]. Bayesian methods address the uncertainty in the diagnostic relations and improve robustness in the presence of missing and noisy evidence, producing a better overall ranking of the potential diagnostic hypotheses. The accuracy, robustness, and timeliness of the reasoner is very much a function of the accuracy of the system reference model.

For system experts, building diagnostic reference models is a difficult and time-consuming task. While experts extract substantial knowledge about fault propagation from their knowledge of subsystems and earlier aircraft designs, gaps arise because: (1) manufacturers update components to improve performance of newer aircraft (for example manufacturers may migrate to active surge control from passive on-off surge prevention), and (2) complex interactions between subsystems are hard to characterize and model a-priori. Often, such knowledge comes from years of experience, and only when an abnormal situation or fault has occurred a number of times.

Recently proposed data mining approaches, applied to the vast amount of operational data collected by the airlines, produce targeted anomaly detection and fault diagnosis applications [17, 36]. This work develops an approach that employs targeted search techniques with a Bayesian learning algorithm to detect and analyze the onset of faults that lead to adverse events during future operations. The methodology is supported by case studies that demonstrate how existing system reference models can be updated by a combination of data mining methods and system expert input to improve Aircraft Diagnostic and Maintenance performance and not endanger the reasoner's certification status.

To choose appropriate data mining methods for this application it is important to develop an understanding of the current reasoner algorithms and the role that the Aircraft Diagnostic and Maintenance plays in aircraft flight operations. Many flight management and flight control functions on an aircraft are now handled by software [147]. This software has to meet stringent certification requirements (DO-178 or Level 1 certification). Aircraft Diagnostic and Maintenance systems are certified at Level 4, implying they play only an advisory role during flight. Changes made to a Level 1 certified system after initial development requires the system to go through an expensive and time-consuming re-certification process. Aircraft Diagnostic and Maintenance are not on the critical path for making flight control decisions on the aircraft, therefore, the system reference model can be treated as data, and can undergo reasonable updates by system experts without re-certification. However, the reasoner algorithm is certified and any changes to the reasoner algorithm would incur expensive re-certification costs. Therefore, updates made to improve Aircraft Diagnostic and Maintenance accuracy and performance are invariably in the reference model, implemented in a way that requires no changes to the reasoner algorithm. This implies that data mining solutions, need to take on the role of supporting system experts in their knowledge engineering tasks of upgrading the system reference model in such a way that no changes have to be made in the diagnostic reasoner algorithm.

Current Aircraft Diagnostic and Maintenance reasoner algorithms make a couple of independence assumptions in defining the system reference model, such as: (1) independence of the fault hypotheses and (2) independence of the evidence nodes given a fault hypothesis. As a result, the system reference model is characterized as a set of Naives Bayes classifiers, which simplifies the approach the reasoner uses to compute the likelihood of fault hypotheses given evidence [81]. Some evidence nodes map directly to sensor values, or monitors that use a computational procedure to generate evidence by combining information from one or more sensors on the aircraft. Updates to the reference model, using the results derived from data mining cannot violate the independence assumptions of the Naive Bayes model.

In this work, we assume the availability of existing reference models for aircraft engine subsystems. In addition, we have access to flight data from a U.S. regional airline that operated a number of jets. The available data ranges over a period of *five* years. The data collected is from a large number of aircraft monitors and sensors, many of them associated with the four engines on the aircraft. Therefore, we have access to a very large amount of flight data, which requires us to design significant data curation solutions [66] to find data relevant for a targeted knowledge engineering application, e.g., improving the detection accuracy and timeliness of detection for a leak in a fuel line.

We address these issues using a 3-step framework for the knowledge engineering task:

1. Select relevant data from which we could derive new knowledge for targeted diagnostic analysis;

2. Apply our targeted data mining algorithms to derive the new knowledge, and with the help of a domain expert isolate updates to improve the reference model; and

3. Perform experiments to demonstrate the augmentations lead to overall improvements in the reasoner performance.

With a domain experts help, we established the specific improvements that could be derived for the reference model within the Naive Bayes classifier framework. These improvements are characterized as local changes to the model structure: (1) *Improve the accuracy of existing reference model relations* by making the evidence more sensitive to particular failure hypotheses without increasing the overall false alarm rate; (2) *Discover new relations between existing information and fault hypotheses or create new ways of combining sensors and fault hypotheses* to improve overall diagnostic accuracy; and (3) *Create new Component to Component relationships* that take into account the dependency between two pieces existing evidence and create a new monitor that combines past evidence to provide stronger evidence in support of a fault hypotheses. We present three case studies to illustrate these updates to the system reference model.

This chapter is organized as follows. Section IV.1 briefly reviews the important characteristics of the on-board model-based diagnostic reasoner systems. Section IV.2 explains how the learned Bayesian model forms the basis for updating the Aircraft Diagnostic and Maintenance reference model without violating the assumptions and properties of the reasoner. Section IV.3 describes the overall framework from curating the data, to using the information in section IV.2 to produced suggested changes. Section IV.4 discusses the implementation of the framework and the a discussion of the knowledge engineering task. Section IV.5 presents the results of our three case studies that demonstrate how the human expert utilized the framework to interpret and utilize the information generated by our TAN structures to update existing reference models. Section IV.6 presents a summary of the approach, and outlines the contribution of this work in our research.

IV.1 Aircraft Reference Model Structure and Diagnostic Reasoners

We briefly review the reference model structure and reasoner algorithms employed in typical Aircraft Diagnostic and Maintenance systems. A traditional system reference model structure, such as the one used in the Boeing 777 Central Maintenance System [46]), can be

represented as a flat bipartite graph with two types of nodes: (1) failure modes or hypotheses and (2) evidence nodes as sensor and monitor variables. Figure 4 shows an example reference model for an engine subsystem. More recently, the aircraft reference models add hierarchy to the structure ,e.g., the Vehicle Integrated Prognostic Reasoner project [66], to manage the complexity of aircraft systems.

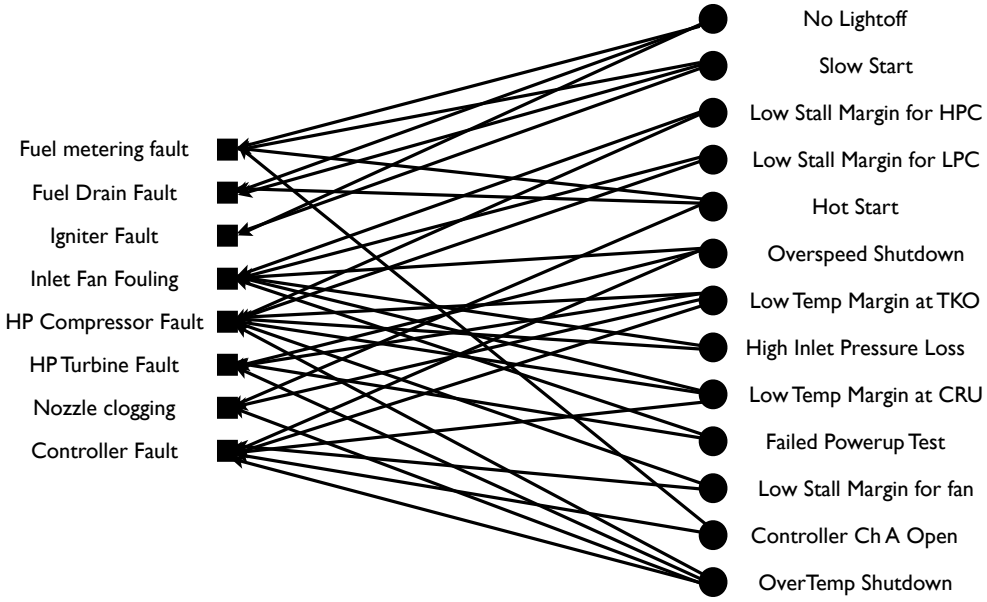

Figure 4: Example Reference Model

"Diagnostic monitors" represent the evidence nodes in the system. In more detail, a monitor provides comprehensive or aggregated information that is based on mathematical and logical functions of raw sensor readings from a component or subsystem. Designing a monitor often requires deep domain knowledge about the component or subsystem, but the details of this information are typically not available to the system integrator. An abstract view of a monitor is shown in Figure 5. With few exceptions, most diagnostic monitors are derived by applying a threshold to a time-series signal. This signal can be a raw sensor value or a derived quantity from a set of one or more sensor values. The intermediate derived quantities are labelled as *condition indicators* (CIs), $x(t)$. Assuming a pre-defined

threshold value θ, we set $m = 1 \Leftrightarrow x(t) \lesseqgtr \theta$. A diagnostic monitor may specify the underlying condition indicator and the threshold or simply provide the net result of applying a hidden threshold. The binary output of the monitor makes the computational framework of the Bayesian reasoner easier to implement. In our work, we use the results of data mining to improve on existing thresholds employed by the monitors, and thereby improve diagnostic accuracy.

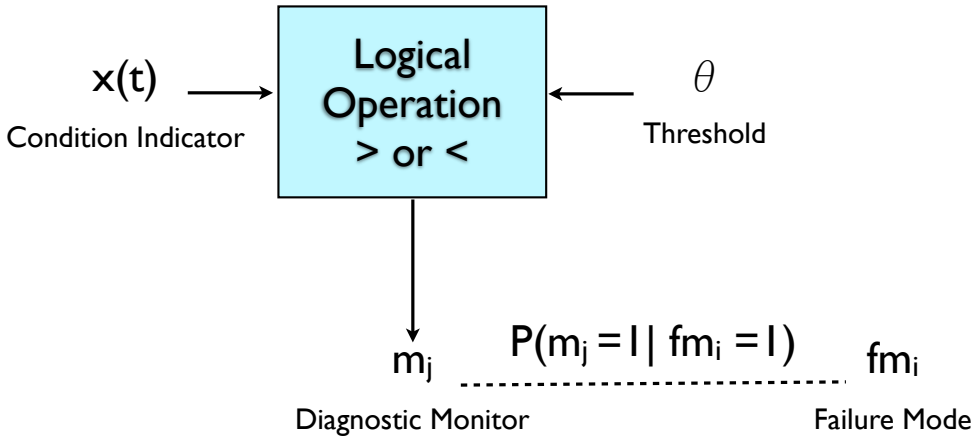

Figure 5: Abstraction of Diagnostic monitor

Given F, the set of distinct failure modes in the system and DM, the set of diagnostic monitors, each failure mode variable, $fm_i \in F$ takes a binary value:

$$\begin{aligned} fm_i = 0 &\Leftrightarrow \text{The failure mode is not occurring} \\ fm_i = 1 &\Leftrightarrow \text{The failure mode is occurring} \end{aligned} \tag{IV.1}$$

In addition, a value of -1 is sometimes used to denote that the failure mode is unknown. The priori probability of failure mode fm_i is denoted by $P(fm_i = 1)$. Failure modes are assumed to be independent of one another, i.e., given any two failure modes fm_k and fm_j, $P(fm_k = 1|fm_j = 1) = P(fm_k = 1)$.

A diagnostic monitor, $m_j \in DM$, either *indicts* or *exonerates* a subset of failure modes called its *ambiguity group*. Each monitor m_i in the system is labeled by three mutually

exclusive values allowing a monitor to express indicting, exonerating or unknown support for the failure modes in its ambiguity group, as shown in equation (IV.2).

$$\begin{aligned} m_i = 0 &\Leftrightarrow \text{Exonerating evidence} \\ m_i = 1 &\Leftrightarrow \text{Indicting evidence} \\ m_i = -1 &\Leftrightarrow \text{Unknown evidence} \end{aligned} \tag{IV.2}$$

An ideal monitor m_j fires only when one or more failure modes in its ambiguity group are occurring. Given the fact that the i^{th} failure mode is occurring in the system, d_{ji}, the detection probability of the failure mode fm_i given indicting evidence provided by the j^{th} monitor is given by:

$$d_{ji} \Leftrightarrow P(m_j = 1 | fm_i = 1), \tag{IV.3}$$

False alarm probability, the probability that an indicting monitor fires when the corresponding failure modes in its ambiguity group are not occurring in the system, is given by

$$\varepsilon_j \Leftrightarrow P(m_j = 1 | fm_i = 0, \forall fm_i \in \textit{Ambiguity Set}) \tag{IV.4}$$

As monitors activate, the reasoner algorithm first performs an elimination step where failure modes that do not associate with that newly activated monitor are removed from the set of probable failure hypotheses. As additional monitors fire, the set should become smaller, and may reduce to a single hypothesis. In situations where there are more than one failure hypothesis, the reasoner uses the probabilistic information in the reference model to generate likelihood values to rank these hypotheses. The probability of false alarms is also calculated to indicate that the current set of monitors may be noisy. As more monitors fire, the numeric values of these probabilities increase or decrease, until a specific failure mode hypothesis emerges as the highest-ranked or the most likely hypothesis. This ranking can be used by mechanics to determine the order in which components need to be checked for repair and possible replacement.

The probability calculations assume that only one fault mode could be active at any given time (single fault hypothesis), and that the monitors are independent of one another given this information. This results in the probability update function for each fault hypothesis, $\forall i fm_i \in F$, being computed using a Naïve Bayes model, i.e., $P(fm_i|m_j,m_k,m_l\cdots) = \alpha \times P(m_j,m_k,m_l\cdots|fm_i) = \alpha \times P(m_j|fm_i) \times P(m_j|fm_i) \times P(m_j|fm_i) \times \cdots$ where α is a normalizing constant. The direct correspondence between the reference model and the simple Bayesian structure provides opportunities to use data mining methods based on a class of generative Bayesian model algorithms for diagnostic reasoning. These newly learned structures can form the basis for designing systematic knowledge engineering techniques for updating the system reference model. We discuss this approach in the next section.

IV.2 A Bayesian Framework For Updating Reference Models

In current Aircraft Diagnostic and Maintenance systems, expert knowledge is central to the creation of diagnostic monitors and the links between these monitors and the failure modes in the reference model. The Naïve Bayes framework governs the reference model structure, and how updates suggested by the data mining results can be incorporated into existing reference models. The example reference model in Figure 4 is reasonably complex because of the multiple-connected nodes, so we use a simpler example shown in Figure 6 to illustrate the model updating methods that we have developed. We revisit the proposed updates to discuss how these translate into updating the reference model for improving diagnoser performance:

1. **Update Monitors.** Update the threshold θ associated with a diagnostic monitor to improve the accuracy of existing relations between monitors and fault hypotheses. The goal is to make the monitor i more sensitive to failure mode j (so that the fault can be detected earlier) without sacrificing the false alarm rate. As an example, consider a change in the threshold for monitor DM_2 with respect to fault FM_1 (see

Figure 7). The threshold value may be made lower to make the fault mode more sensitive to the monitor value, or it may be increased to decrease the false alarm rate;

2. **Add new links between Monitors and Failure Modes.** This is equivalent to discovering new relations between monitors and fault hypotheses, which results in added links between monitors and failure modes. Specifically this could take two forms: (a) creating a new monitor DM_j and deriving the conditional probability d_{ji} to associate it with the failure mode FM_i, or (b) assigning a non-zero d_{ji} between an existing monitor DM_j and a fault hypotheses, FM_i if that link did not exist before. An example of the latter is a new link created between FM_1 andDM_3 in Figure 7; and

3. **Create Super Monitors.** New monitors are derived that absorb the dependency between existing monitors to avoid violations of the Naïve Bayes assumptions. An example of this situation would be the discovery of a dependency between DM_1 and DM_4 in Figure 7.

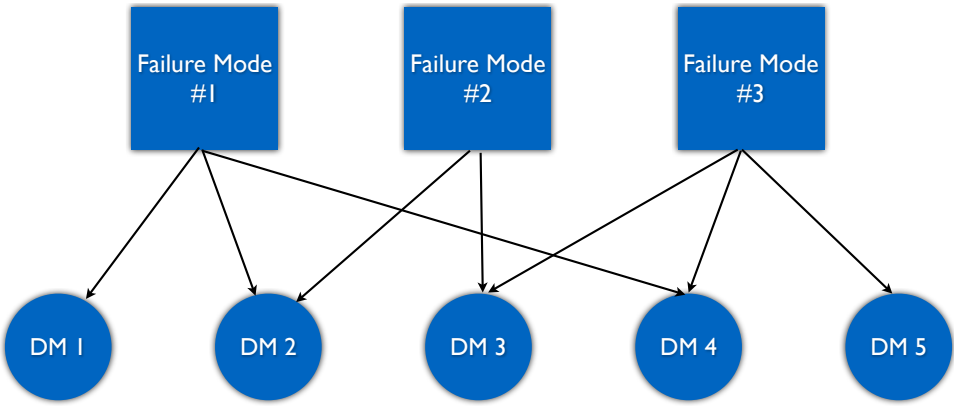

Figure 6: Graphical Representation of a Reference Model.

Limiting our approach to this set of reference model updates to avoid increased certification costs presents two important challenges. The first is related to scaling problems for conditional probability distributions for large models. Consider the example

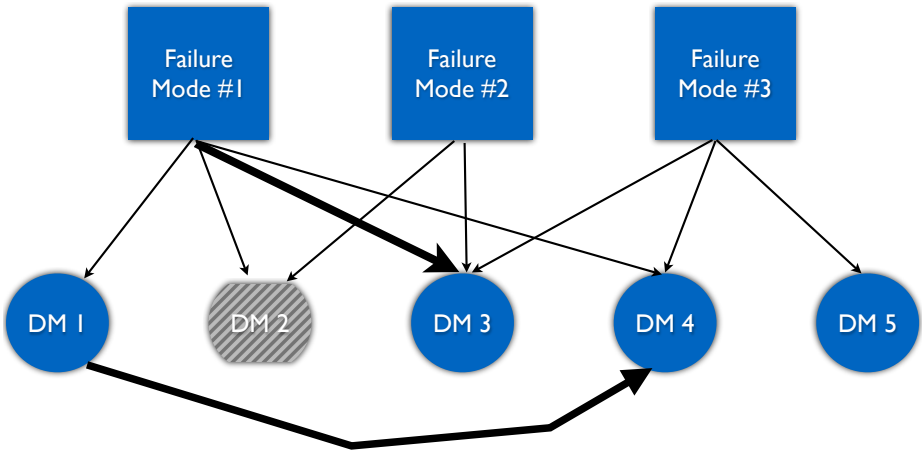

Figure 7: Additional Information derived from data: (a) update to monitor threshold DM_2 with respect to fault FM_1 (b) finding a new relation between FM_1 and DM_3, and (c) Discovering that monitors DM_1 and DM_4 are causally related

where the conditional probability between FM_2 and DM_2 has to be updated because the data mining algorithm finds a better threshold for monitor DM_2. Since DM_2 is a shared monitor between fault hypotheses FM_1 and FM_2, which means the faults are causally dependent. Therefore, to reason about the likelihood of FM_1 being indicted by the evidence, i.e., $P(FM_1|DM_1,DM_2)$, we have to consider marginalization of the joint distribution $P(FM_1,FM_2,DM_1,DM_2,DM_3)$ with respect to nodes FM_2 and DM_3. Generating the joint probability distribution table requires much more information, which the domain expert may be unable to provide, and it is hard to directly derive this information from data [139]. Preserving the Naive Bayes model structure assumptions, i.e., the independence of the fault hypotheses and the independence of the monitors associated with a fault hypotheses, simplifies this task of deriving the conditional probabilities. In our example, the discovery of a new link between FM_1 and DM_3 makes all of the failure modes dependent, which greatly increases the number of parameters needed to specify the joint probability distribution. The Naive Bayes assumption allows for a simplified re-factoring of the problem, making the conditional probability tables easier to specify. Figure 8 shows the local structure used for failure mode FM_1.

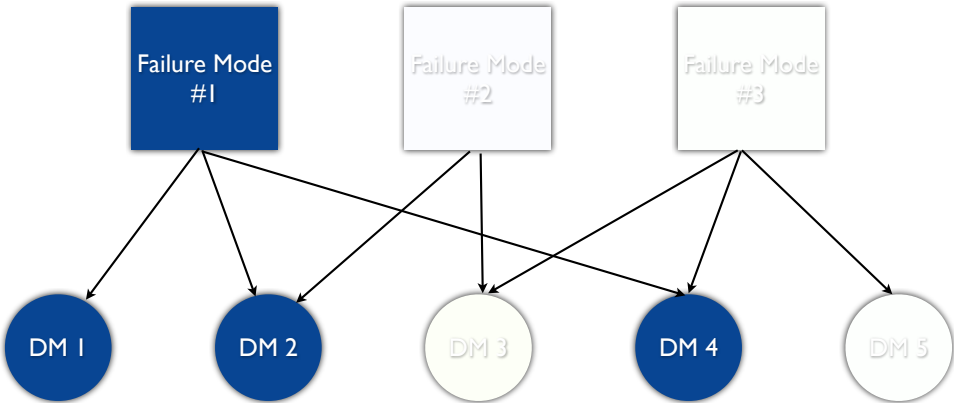

Figure 8: The relevant structure after isolating a Failure Mode

A second challenge arises when the data mining algorithm finds dependencies among monitors, such DM_1 and DM_4 in Figure 7. This clearly violates the assumption of independence of monitors given the fault mode. We address this problem by defining the notion of a "super monitor." To accommodate the dependency between DM_1 and DM_4 while retaining the Naive Bayes modeling framework, the two monitors are combined to form a "Super Monitor" and the sub-structure between FM_1, DM_1, and DM_4 is replaced by a new node SM_1 and a link from FM_1 to SM_1, as shown in Figure 9. In general, combining existing monitors, M_i and M_j implies stronger indictment evidence for the failure mode FM_k. That is, $P(DM_i = 1, DM_j = 1|FM_k = 1) > P(DM_i = 1|FM_k = 1) \times P(DM_j = 1|FM_k = 1)$. Note that monitors DM_i and DM_j are not removed from the reference model because they may provide supporting evidence for other faults. This illustrates yet another local update method applied to the reference model. The creation of this new monitor is triggered by the presence of the edge in the learned network, and isn't concerned with the direction of the edge. This helps alleviate issues involving the manner with which the directionality is assigned in the learning algorithm.

A number of Machine Learning techniques for building Bayesian networks from data has been reported in the literature [51], [24],[58]. For example, state-based hidden Markov

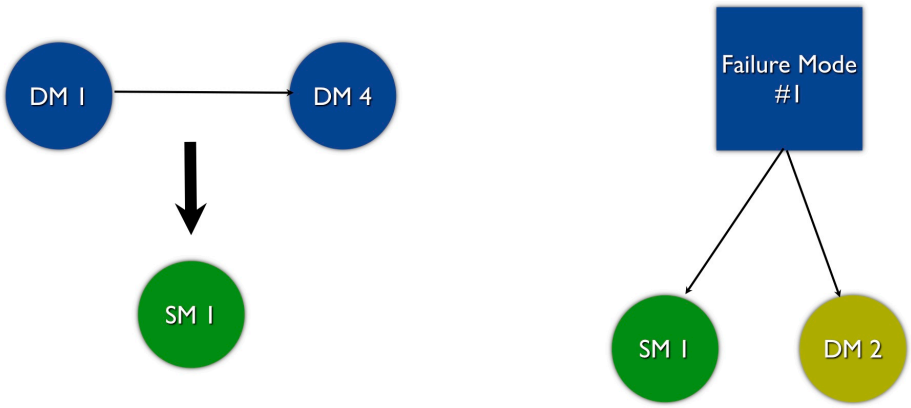

Figure 9: The construction of a Super Monitor

Models [149], and more general Dynamic Bayesian Network [39], [91], [137], [164] formulations can be employed to capture the dynamics of aircraft behavior and effects of faults on system behavior and performance. However, given that our primary task is to extend and improving performance an existing Aircraft Diagnostic and Maintenance and not violate the Naïve Bayes model assumption imposed on the reference model, we have adopted data mining algorithms whose output is similar in nature to the reference model structure(although not equivalent). The output is also easily interpreted by experts making it easier for them to update existing reference model. Our approach learns Tree Augmented Naïve Bayesian networks from operational flight data. The approach is justified in Section IV.4

IV.3 Three Step Knowledge Engineering Approach

In this section, we start by recalling the outline of the three step knowledge engineering approach.

1. Select segments of operational flight data from which we can derive the new knowledge for diagnostic analysis;

2. Apply our targeted data mining algorithms to derive the new knowledge and with the

help of a domain expert come up with updated structures that meet the constraints discussed in Section IV.2 to improve the reference model; and

3. Perform experiments to demonstrate the augmentations lead to overall improvements in the reasoner performance.

IV.3.1 Curating Data

INPUT: Raw flight operations data for a similar set of aircraft for an extended set of flights, Existing Reference Model from which we can derive the set of fault hypotheses, and the set of known monitors.

OUPUT: A Curated Database that contains all of the cleaned up flight segments for the set of aircraft.

The first step starts with the large operational flight data for a set of similar aircraft. We assume that the set of fault hypotheses whose detection performance needs to be improved has been selected by the domain experts or aircraft engineers. The goal in this step is extract relevant data segments from which additional diagnostic relations can be derived.

Flight data can be extracted by the aircraft tail number and a complete flight segment, which includes the following phases, startup, taxiing, takeoff, cruise, descent, and landing. Each flight segment not only contains a time series report of the diagnostic monitor values (normal or abnormal), but also the time-stamped CIs.

It is important that the curation process be general enough to be applicable to different types of temporal and time-series data. To maintain the generality and scalability of the curation approach, while making it efficient and effective for multiple scenarios, we employ a database schema that performs dual tasks. First, it plays the role of a lookup for the raw files, and second it uses normalized relational tables for the different condition indicators. The tables, ordered by the atomic operations that provide links to all of the meta-data, are structured to allow retrieval of data for different systems of the vehicle and different flight operations modes. Information such as the length of a flight, the flight date, and relevant

annotations make this control table easy to filter for more complex queries. Creation of the initial database is time-intensive because of the need to transform the raw data, but this is a one-time process that can support multiple data mining analyses and validation studies. The curation process also results in a clean up where incomplete and inconsistent flight segments are dropped.

IV.3.2 Causal Discovery Methods to Update the Reference Model

Applying data mining methods to discover new relations and update the reference model require finding the right flight segments from which this information can be derived, and then applying the appropriate data mining algorithms to find the relevant relations.

IV.3.2.1 Building Relevant Flight segments

INPUT: We start with the existing reference model, the curated database, and the fault hypothesis of interest.

OUTPUT: Flight segments from which the relevant new information to support diagnosis of the fault hypothesis can be generated.

A structured dataset is created that includes: (1) flight segments where the fault manifests as well as nominal flight segments, (2) a set of monitor and condition indicator values that are relevant to the fault under consideration; this set may be obtained by analyzing the reference model and by seeking expert input for additional features. The flight data segments with failures are identified by looking for additional sources that may report failure information in aircraft, such as the ASIAS database[1] that is maintained by the FAA. The ASIAS database provides information about the aircraft tail number, the date, and the flight when the failure occurred, and additional information about the failure event. To ensure that we capture enough information about the failure, especially indicators that may imply early onset of the failure, with expert help, we trace back a number of flights from the

[1] http://www.asias.faa.gov/portal/page/portal/asias_pages/asias_home/

adverse event report. Capturing labeled and faulty flight data allows us to develop classifier algorithms that help differentiate non faulty and faulty behaviors.

IV.3.2.2 Building Classifier Model

INPUT: Data segments divided into nominal and faulty behavior; Classifier type to build.

OUTPUT: Classifier model.

The classifier type chosen should be effective at classifying nominal from faulty behavior and provide diagnostic information that can be accommodated into the existing reference model structure. This step produces a structure for testing the usefulness of the data in providing diagnostic information and examining the results for updating the reference model.

IV.3.2.3 Validating Classifier Results

INPUT: The derived classifier model; data segments divided into training and test sets.

OUTPUT: Results of N-fold cross validation studies.

It is important to run cross-validation studies to get good estimates of the accuracy of classification and the false-alarm rate for the data with this type of structure. It is important that these numbers satisfy the requirements of the aircraft diagnosis task.

IV.3.2.4 Exploring Classifier Structures to Find Augmentations

INPUT: Set of validated labeled data vectors that represent nominal and faulty flight segments

OUTPUT: A Classification structure that clearly indicates which feature best support the fault/no-fault binary classifier

In our case studies, on expert advice, we further segmented the flight data into different phases of flight operation, e.g., engine startup, take-off, and engine shutdown, and ran the classification studies on individual segments. The intuition was that certain faults would be

more prominent in particular phases, e.g., engine faults show the largest effects during take-off when the engine is stressed the most. We discuss the details of the classifier algorithms in Section IV.4. Another specific approach we applied to determine early fault indicators was to define the flight segments into "bins." A bin represented a set of flights, for example, bin 1 could be defined as the 10 flights just before an adverse event or failure occurrence, bin 2 would be flights 11-20 before the failure occurrence, and so on. This procedure is also discussed in greater detail in Section IV.4.

IV.3.3 Updating Reference Model and Verifying Performance Improvements

INPUT: Expert Generated Augmented Reference Model, Reasoner

OUTPUT: Augmented Subsystem Reference Model

The new monitors and relations between fault hypotheses and monitors have to be integrated into the original reference model. This is done with the help of the domain experts. The experts make judgements using the results generated by the classifier algorithms to update the conditional probabilities and false alarm rates associated with the fault hypotheses and monitors. To test reasoner performance after the updates, traces of the incidents from the dataset are then fed to the reasoner with both the original model as well as the new reference model. Each trace will look at successive runs of the aircraft over a stretch of time that ends with the failure occurrence. The expert determines whether the traces with the augmented reference model provide sufficient improvements in detection and isolation of the correct fault. Improved performance leads to earlier maintenance decisions and greater overall safety. The output will either be confirmation of the approved changes, or empirical proof to reject the changes.

IV.4 Implementation

This section discusses the implementation of the three-step knowledge engineering approach defined in Section IV.3. This implementation has been formed from initial testing

on simulated engine data [99]. The curation process is presented in Section IV.4.1, and the resulting flight segments generated for the classification studies are described in Section IV.4.2. Section IV.4.3 discusses the learning algorithm based on the Tree Augmented Naïve Bayes model (TAN) [51] used for deriving the classifier structures for the set of faults that define our case studies. Augmenting the reference model using the generated classifier structures and expert input is presented in Section IV.4.4.

IV.4.1 Aircraft Data

The data comes from a fleet of 30+ identical four engine aircraft that composed a U.S. regional airline. The data covers about five years of flight operations, with each aircraft operating 2–5 flights each day. The Aircraft Condition Monitoring System collects sensor data from the propulsion subsystem, the airframe, the aircraft bleed subsystem, and the flight management system in a central location on the aircraft during flight to support fault analysis by the Aircraft Diagnostic and Maintenance, and maintenance operations when the aircraft lands. The aircraft sensors have different precision levels, and different sampling rates, therefore, not all sensors collect the same amount of data per flight.

This data is typically stored in raw, uncompressed form as binary files. On landing, the Aircraft Condition Monitoring System recorded data is transferred to permanent storage (in our case, the data was stored on CDs). We apply our initial data retrieval and pre-processing algorithms to this raw time-series data from the multiple CDs. From this initial step, flight data is generated indexed by the tail identification number of the aircraft, and date and time of flight.

In addition to the flight data, we have independent access to Federal Aviation Administration (FAA) reports through the Aviation Safety Information Analysis and Sharing (ASIAS) database system, which is a collection of adverse events reported by various airline operators. Examples of adverse events related to our flight data included incidents, such as loss of an engine and engine on fire. Many of these incidents are major safety

hazards, and cause the affected aircraft to abandon its flight plan and make an emergency landing at the nearest airport. A list of such adverse events and the root cause failures associated with these events define the case studies discussed in this paper. We ignored ASIAS events like sprinkler incidents in the main cabin, because they did not have serious implications on aircraft flight safety.

Two of our three case studies are computer-aided engine shutdown events during flight, and the third is an excessive engine vibration that resulted in a crew-initiated shutdown of that engine. From the ASIAS records, we identified the aircraft (by its tail number) and the exact flight in which the adverse event occurred. Since the goal of this knowledge engineering study is update the reference model, to enable early and reliable detection of an evolving fault[2] and thus avoid the adverse event, we made sure that the data segments chosen included N previous flight segments along with the flight segment in which the adverse event occurred. Our domain experts used their knowledge of the temporal characteristics of the particular fault (slow versus fast evolving) to determine the value of N for each case study.

IV.4.1.1 Brief overview of Case Studies

The first case study pertains to an engine overheating problem, which triggered the alarm systems and engine shutdown on the belief that the engine was in danger of catching fire. Simple analysis from the graphs of raw sensors attributed this to a faulty fuel metering hydro-mechanical unit(Fuel HMA) in the third engine that cause the overheating, which eventually led to the engine shutdown. The fuel metering unit is a controller-actuator that controls fuel flow into the engine combustion chamber to produce the desired thrust. Our domain experts informed us that a Fuel HMA fault is a slowly evolving *incipient*) fault. The experts suggested that manifestations of this fault could likely occur about 50 flights *before* the engine shutdown event took place. We made the assumption that only the one engine

[2]Early detection allows mechanics to make necessary repairs, and thus avoid the adverse event

with the Fuel HMA issue was faulty, so we had 50 instances of faulty engine flight segment data and at least 150 (50×3) instances of nominal engine flight segment data under the same flight conditions.

The second event involved excessive vibration in an engine that forced the crew to shut the engine down manually. In this case, the FAA report attributed the excessive vibration to a broken blade in the turbine bucket of the engine. Again, with expert help, we identified 50 prior flight segments to capture the faulty engine situation, and the data from the other three engines for these flight segments was labeled as nominal.

The third event, like the first, was an engine shutdown triggered by the fire alarm system on the engine. After the fact, FAA investigators determined that the cause was a leaking fuel manifold. But the fault was not detected by any of the existing sensors and monitors on the aircraft. The third failure is different from the first two in that the cause is not isolated to a specific subsystem, i.e., an engine. Instead the fault occurred in a mechanical unit that regulates fuel to two of the four aircraft engines. The manifold leak was also characterized as an incipient fault by our experts, and they suggested that 50 prior flight segments could be used as examples of faulty flight instances. This case study produced a different result from the first two. The experts attempts to update a subsystem model to better detect the fault was not successful, therefore, the conclusion was this fault was better handled at a system level as opposed to the subsystem level.

IV.4.2 Description of Flight Segments

Our case studies, focus primarily on the aircraft engine subsystem and fuel flow into the engines. A set of condition indicators related to engine health were extracted as time series data, and then annotated by the different modes of operation of the engines: (1) startup (2) takeoff, and (3) shutdown. We did not include data from the other primary phases: climb, cruise, and descent/landing in our analyses, because our experts surmised that the engines were most stressed during takeoff, and knowing the initial and final state of the

CI Name	Description
StartTime	This CI provides the time the engine takes to reach its idling speed. Appropriate threshold generates the *no start* diagnostic monitor.
IdleSpeed	This CI provides the steady state idling speed. Appropriate threshold generates the *hung start* diagnostic monitor.
peakEGTC	This CI provides the peak exhaust gas temperature within an engine start-stop cycle. Appropriate threshold generates the *overtemp* diagnostic monitor
N2atPeak	This CI provides the speed of the engine when the exhaust gas temperature achieves its peak value. Appropriate threshold generates the *overspeed* diagnostic monitor.
timeAtPeak	This CI provides the dwell time when the exhaust gas temperature was at its peak value. Appropriate threshold generates the *overtemp* diagnostic monitor.
Liteoff	This CI provides the time duration when the engine attained stoichiometry and auto-combustion. Appropriate threshold generates the *no lightoff* diagnostic monitor.
phaseTWO	This CI provides the time duration when the engine controller changed the fuel set-point schedule. There are no diagnostic monitors defined for this CI.
prelitEGTC	This CI provides the engine combustion chamber temperature before the engine attained stoichiometry. Appropriate threshold generates the *hot start* diagnostic monitor.

Table 1: Startup Features Transformed from the Raw Data

engine at the start and end of a flight, was more important for diagnostic purposes. The flight segment data was obtained in two steps: (1) Data from all flights for the selected condition indicators was collected into the curated database for all four aircraft engines; (2) The labeled flight segments, representing nominal and faulty situations was extracted into individual data sets for the classifier studies. Lists for CI's used for each flight segment are found in Tables 1 and 2:

The flight segments were further broken down so that each engine represented a separate data point. The data included 50 time segments, so for the four engines we had $4 \times 50 = 200$ data points, and each data point was defined by 25 features corresponding to the 25 CI's. For the first two case studies in Section IV.4.1, only one of the four engines

CI Name	Description
tkoN1, tkoN2, tkoEGT, tkoT1, tkoPALT	These CIs provide the fan speed, engine speed, exhaust gas temperature, inlet temperature and pressure altitude, respectively, averaged over the time interval when aircraft is operating under takeoff conditions. There are no diagnostic monitors defined for these CIs.
tkoMargin	This CI provides the temperature margin for the engine during takeoff conditions. Appropriate threshold generates the *medium yellow* and *low red* diagnostic monitors.
Rolltime	This CI provides the time duration of the engine's roll down phase. Appropriate threshold generates the *abrupt roll* diagnostic monitor.
resdTemp	These CI provide the engine exhaust gas temperature at the end of the engine's roll down phase. Appropriate threshold generates the *high temp* diagnostic monitor.
N2atDip, dipEGTC	These CIs provide the engine speed and the exhaust gas temperature at the halfway point in the engine's roll down phase. There are no diagnostic monitors defined for these CI.
N2cutoff	These CI provide the rate of change of the engine speed at the halfway point in the engine's roll down phase. There are no diagnostic monitors defined for these CI.

Table 2: Takeoff and Shutdown Features Transformed from the Raw Data

is faulty, and the other three were categorized as "nominal." A quick note, that the term "nominal" here does not indicate the absence of failures in the engine, but rather that it does not include effects of the fault under investigation. We developed additional operators to break this into multiple tables, one for each mode of operation.

IV.4.3 Learning Tree Augmented Naive Bayesian Networks

Our choice of the data mining algorithm is governed by the desire that the learned structure closely match the reference model structure, which implies that the learned structures satisfy the Naïve Bayes assumptions. However, CIs for aircraft subsystems may not be independent given a fault hypothesis for multiple reasons: (1) two CIs may be based on dependent measurements, where one measurement is downstream from the other, e.g., a CI derived from a pressure measurement at the end of a pipe is not independent of a second CI whose value is derived from a pressure measurement at the inlet point in the pipe; and (2) two CIs may share one or more sensor measurements, e.g., two different measures of health state of an aircraft engine may both use the engine temperature in their computations. This dependency information, when known, can be used to improve diagnosis results. Therefore, in this work, we prefer learning algorithms where the independence assumptions may be systematically relaxed to capture additional discriminatory evidence for diagnosis.

A method that satisfies these requirements is the Tree Augmented Naïve Bayesian learning algorithm [51], also called the TAN classifier. The TAN provides a simple extension to the Naïve Bayes network model. The fault hypothesis presented as the root or the class node is causally linked to every evidence node, which correspond to the CIs that support that fault hypothesis. In addition, an evidence node (CI) can have at most two parents: (1) the class node, and (2) a causal connection to another evidence node (CI). These constraints maintain the directed acyclic graph requirement of Bayesian networks, and produce a more nuanced tree that allows for additional dependency relationships among the

CIs without becoming too general and thus harder for the expert to interact with and use when augmenting the reference model.

The TAN Structure can be generated in several different ways. One approach uses a greedy search that constrains the graph from building “illegal" edges from the evidence nodes[3] [34]. We employ a procedure that builds a Minimum Weighted Spanning Tree of the evidence nodes and then connects the fault node (root) to all of the evidence nodes in the tree [51]. A standard algorithm (e.g., Kruskal’s algorithm [84]) can be employed to generate the Minimum Weighted Spanning Tree. The mutual information function is used for pairwise edge weight computations [51]. This metric calculates the how much information one variable provides about the other. Note that the Mutual Information measure is not directional. After the minimum weighted spanning tree is built, one of the nodes is designated as the root node and the direction of the edges is based on that choice. This search path for this choice uses a likelihood measure with respect to the training data to find the optimal node for the root.

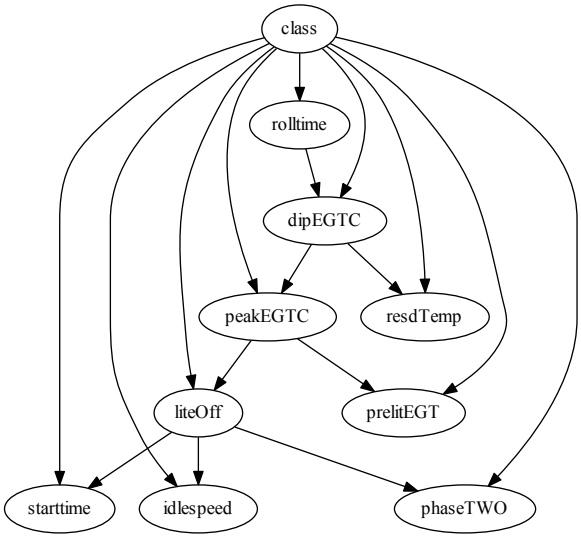

Figure 10: Example TAN Structure

[3] an illegal edge is created when an evidence node is assigned more than one parent

An example TAN structure generated using our minimum weighted spanning tree algorithm is illustrated in Figure 10. The root node, labeled class, is the fault hypothesis of interest. The other nodes represent evidence supporting the particular fault hypotheses. For the structure in Figure 10, rolltime, a monitor associated with the shutdown phase of the aircraft is the anchor evidence node built with the minimum weighted spanning tree. We refer to the anchor node as the *observation root node* in the TAN structure. Like a Naïve Bayes classifier structure, the fault hypothesis node (class) is linked to all of the relevant monitor nodes that support this hypothesis. Dependencies among some of the monitors, e.g., rolltime and dipEGTC, are captured as additional links in the Bayes network. Note that the TAN represents a static structure; it does not explicitly capture temporal relations among the evidence. The observation root node is important; in some ways, it represents an important monitor for the fault hypothesis, since it is directly linked to only this node. This means the distribution used in the observation root node (whether it be a discrete CPT, or a continuous distribution) is conditioned only on the priors of the class distribution. The rest of the minimum weighted spanning tree structure is also linked to this node but all other conditional probability tables (CPTs) generated for this TAN structure include the class node and at most one other evidence node.

The structure of the TAN found using a minimum weighted spanning tree and the choice of the root provides a heuristic ranking of the features. We discovered in close consultation with the system experts, that the closer a CI node is to the root of the tree (fault hypothesis), the more important this CI is for diagnostic analysis. Much like Information Gain in a decision tree [133], the mutual information calculations of edges between node variables (i.e., CIs) in the minimum weighted spanning tree will produce an ordering of CIs from greater to lesser impact. The generated TAN illustrated above first points the domain expert to the observational root node, i.e., the condition indicator just below the fault hypothesis node. As one moves down the tree hierarchy, the corresponding CI's have a smaller impact in establishing the fault hypothesis.

We used an implementation of the TAN algorithm from the Weka [59] toolkit for our case studies. Weka uses CPT based Bayesian structures, and preprocesses the data using a discretization algorithm. The discretization algorithm bins the individual features into ranges that create the biggest unbalance in the class labels for each feature value (or pairs of feature values when there is a dependency between features), to generate CPTs that provide the most differentiation between classes. The choice of the observational root node is determined by the CI node that provides the best discrimination among the nominal versus faulty class as calculated by the mutual information measure. The value of the CPT and more specifically the ranges found by the preprocessing algorithm are essential for updating existing monitor thresholds and adding new links between monitors and the fault hypotheses in the reference model.

IV.4.4 Using TAN Models to Update the Reference Model

The TAN structure is similar to the structures shown in Section IV.2. The bins used for examining performance as described in Section IV.3.3 produce the best TAN structure to use for updating the reference model.

As discussed, augmentations created from the TAN structure support all of the following updates:

1. **Update Monitors** Updating a monitor is equivalent to updating the threshold on the CI associated with the monitor. This requires studying the discretization of the CI used to create the CPTs. Applying marginalization to the CI parent (if one exists) will produce general probabilities for each set of ranges found through the discretization. The fault range is established from the range that has the highest probability of the failure mode given the marginalized CPT. The value that defines the border between the nominal range and faulty range is taken as the new threshold for the monitor. Given the data associated with the structure in Figure 10, the derived CPT for the rolltime CI is given in Table 3. The table indicates that the fault node, Fuel HMA

Class	(-inf-34.875]	[34.875-inf)
Nominal	.823	.177
Fault	.227	.773

Table 3: Example CPT for Finding Thresholds

failure, is more likely when rolltime is > 34.875. With an expert's approval this change may be introduced into the reference model to improve the accuracy and time of detection of the FuelHMA fault.

2. **Add Monitors to indict Failure Mode** A new CI that appears in the TAN structure, may imply a new monitor. Again, consultation with the domain experts will help determine the relevance of this CI, and the choice of threshold (like the previous step) to optimize fault detection. If, for example, *resdTemp* appears in the TAN structure of Figure 10, but it does not exist in the reference model, the experts and the data mining researchers may examine the CPT for this CI jointly in the manner discussed above, and add a new monitor that uses a threshold based on the value discovered in the CPT.

   A second possibility is that the threshold associated with an existing CI contradicts the threshold value of a monitor that already exists. For example, the CPT associated with this CI indicates the higher likelihood of a fault when the CI values exceeds a threshold, but the existing monitor is designed to generate an alarm when the CI value falls below a threshold. After careful examination, the domain experts conclude that the addition of a new diagnostic monitor defined by the new threshold is helpful in improving detection performance. Using the example of the threshold for *rolltime* in Table 3, a new monitor is defined with the threshold of greater than 34.875 because the previous *rolltime* monitor was designed to generate an alarm for values less than a threshold value $\theta = 34.875$. The earlier monitor will be replaced by the new monitor for fault detection and isolation.

3. **Creating new Component to Component relationships** If a relationship between an observational root node and a child node in the TAN is deemed important to the expert, this can be the basis for forming a "super monitor" This coupling of the CIs can be transformed into a new monitor that adds information not only in a single flight, but across adjoining flights segments as well. For example, if the original structure showed a possible relationship between monitors in flight n followed by flight $n+1$, the causality might result in this new monitor to fire only when the two original monitors fire in that explicit sequence, flight n and flight $n+1$. Not only does this super monitor combine the results from other monitors, but it also indicates cyclic behaviors that again provide useful diagnostic information not originally captured by the reference model.

   In general, super monitors can model complex interactions thus increasing the overall discriminability properties of the reasoner. The consequence of using a super monitor, is that the usefulness of the two monitors used in the construction are diminished. The links from the monitors to the isolated failure mode are removed (they remain active for any other failure mode in the original reference model). The new monitor is created which uses logic, such as AND and OR to combine the original monitors. Also a new monitor may be subsumed into a super monitor relationship. An expert uses the TAN in Figure 10 and the monitors associated with rolltime and dipEGTC, and decides that the relationship between the two is strong enough to produce a super monitor that indicts a fault if and only if, both the monitors for rolltime and dipEGTC would have indicted the fault. This would remove the direct relation between the monitors and the failure mode, and instead replaces them with the single super monitor.

IV.5 Case Studies

We use three case studies to demonstrate the effectiveness of our three-step approach to updating the subsystem reference model to improve diagnostic performance. Domain experts play an integral role in interpreting the TAN structures derived from flight data, and determining how to update the reference model. We employ two standard metrics to evaluate the TAN models generated: (1) the classification accuracy and (2) the false positive rate.To systematically evaluate these metrics we utilize a 10-fold cross validation approach.

After updating, the reasoner is applied to the new system reference model to determine if the new model provides an improvement in diagnostic performance, i.e., higher accuracy and faster detection time, with the new model. The test traces to evaluate performance are generated from relevant flight data.

The three case studies are discussed in greater detail below.

IV.5.1 Case Study 1

This first case study involves the Fuel HMA fault, which resulted in engine overheating and eventual shutdown. The TAN classifier was derived by comparing the data from the faulty engine against the three other engines on the aircraft, which were assumed to operate normally during the period of 50 flights before the adverse event.

IV.5.1.1 Experiment 1: Classification Accuracy of Generated TAN structure

Experiment 1 in this case study studied the effectiveness of the generated TAN classifier structures in isolating the fault condition using the set of CIs that were chosen by our experts. The values for the CIs over the 50 flights was calculated from the flight data. Data from the three engines of the aircraft that showed no abnormalities (1, 2, and 4) was labeled as nominal, and the data associated with engine 3, where the shutdown incident occurred, were labeled as faulty.

The average classification accuracy of the derive TAN structures after running 10-fold

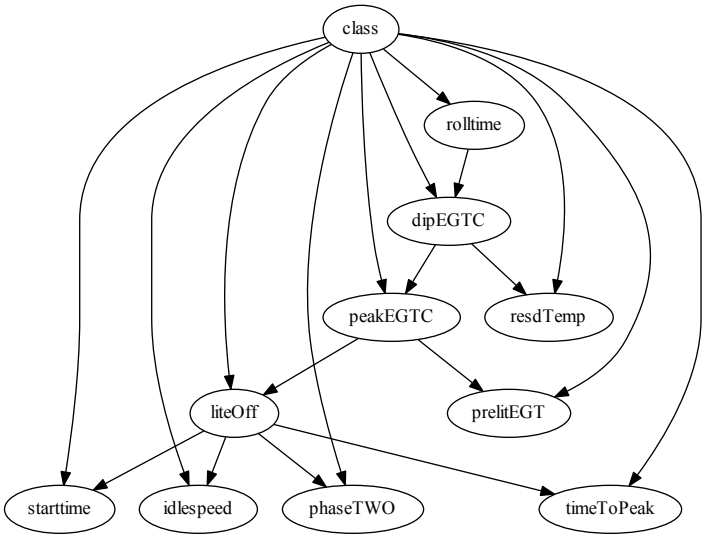

Figure 11: TAN Structure Generated using Data from all 50 Flights

cross validation was 99.5% with a .7% false positive rate.This clearly implied that the set of CIs are appropriate for detecting and isolating the Fuel HMA fault. The next step was to conduct further experiments to ensure that the classification structure was not just as artifact of engine position, i.e., engine three versus the other engines on the aircraft. This involved running the TAN classifier generation using training data from engine 3 (faulty) versus one of the nominal engines (engines 1, 2, or 4). The data from the other two nominal engines was used as test data. If the classifier split the test data between the nominal and faulty classes, it would indicate that the TAN structure was more likely an artifact of engine position on the aircraft. For the three experiments (one of the nominal engines used for training and the other two for test), the fault classification accuracy remained at or above 90%, indicating that the TAN classifier was truly differentiating between the fault and no-fault conditions.

Bin	Flights	Acc.	FP%
1	1 to 10	97.65%	2.30%
2	11 to 20	93.90%	5.70%
3	21 to 30	94.65%	5.30%
4	31 to 40	96.62%	3.50%
5	41 to 50	96.06%	4.10%

Table 4: Accuracy, False Positive Rate from Different Data Segments

IV.5.1.2 Experiment 2: Using the TAN structure to Update Reference Model

The domain experts were more closely involved in Experiment 2. First, the experts examined the TAN structure shown in Figure 11 created from all 50 flights set used in Experiment 1. The expert's attention was drawn to the relationships between different pairs of CI's for different phases of the flight:(1) rolltime and dipEGTC during the Shutdown phase, and (2) PeakEGTC and Starttime during the Startup phase. The expert concluded that there was a likely dependence between the shutdown phase of flight n and the startup of the next flight, $n+1$. The reasoning was that an incomplete or inefficient shutdown in the previous flight created situations where the startup phase of the next flight was affected. The expert hypothesized that this cycle of degradation from previous shutdown to the next startup resulted in the fault effect growing with each flight, and eventually impacted a number of CIs of the faulty engine. This phenomena indicated a causal relation that was not captured in the current reference model. The experts suggested that a super monitor that combined CIs associated with a landing and subsequent take-off would aid the diagnostic reasoner. However, the experts wanted to gain a better temporal understanding of how this relationship between monitors evolved over multiple flights.

To address this, a binning procedure was developed, and the 50 flights were divided into 5 bins of 10 flights each. The data from the 10 flights for a corresponding bin was used for training, and the data from the other 40 flights were used as test data. Additional test data was also generated from flights after engine three was repaired after the adverse event. Table 4 shows the accuracy and false positive rate(FP%) metrics reported for the

Bin	Flights	Obs. Root Node	Children of ORN	Notes
1	1 to 10	IdleSpeed	StartTime	Thresholds Chosen from this Bin due to low FP
2	11 to 20	peakEGTC	liteOff,dipEGTC	peakEGTC Important Node
3	21 to 30	peakEGTC	liteOff,dipEGTC	peakEGTC Important Node
4	31 to 40	startTime	peakEGTC	Links startTime and PeakEGTC
5	41 to 50	liteOff	phaseTwo,RollTime	Links Startup and Rolldown CI

Table 5: Observational Root Node and Immediate Child Node for Classifiers Created from Different Data Segments

five experiments corresponding to five bins of 10 flights each (for a total of 50 flights). The observation root node, and its immediate child in the generated TAN structures are listed in Table 5.

The conventional wisdom was that the accuracy and false positive metrics would have the best values for the classifiers generated from data close to the adverse event, and performance would deteriorate for the TAN structures derived from bins that were further away from the incident. The results show partial agreement. The bin 1 experiment produced the highest accuracy and lowest false positive rate, but the next best result came from the bin 4 data. The high performance of the TAN in bin 1 meant that the discretization used in the CPTs would be used for threshold updating and adding any new monitors to the reference model.

While performing this threshold updating, additional information was discovered by the domain expert. The discretization of the startTime CI allowed the expert to discover that the startTime showed a higher probability of a fault when the value indicated a faster than nominal start. The original monitor for this CI was based on a greater than relationship threshold for a slowStart monitor. This discovery in the CPT implied a new monitor called fastStart that examined if the startTime was much faster than nominal could be added to

detect the failure mode. The new monitor and it associated threshold value derived from the discrete CPT for this CI was used to update the reference model.

The results of bin 1 and bin 4 prompted the domain expert to study the bin 1 to bin 4 TANs more closely. The expert concluded that two CIs, startTime and peakEGTC showed a strong causal connection for bin 4, and startTime was highly ranked for the bin 1 TAN. On the other hand, PeakEGTC was the root node for bins 2 and 3. This study led the domain expert to believe that a new monitor that combined startTime and peakEGTC would produce a reference model with better detection and isolation capabilities.

This new diagnostic monitor combined information from the newly formed fastStart monitor and the HighTemp monitor to improve detection of the fuelHMA fault. To accommodate the super monitor, the connection from the FuelHMA fault hypothesis to the individual monitors was deleted to avoid redundancy and preserve the Naive Bayesian structure. Therefore, the updated reference model included improved threshold values for some monitors, as well as the new super monitor.

IV.5.1.3 Experiment 3: Verifying Improvement in Reasoner Performance

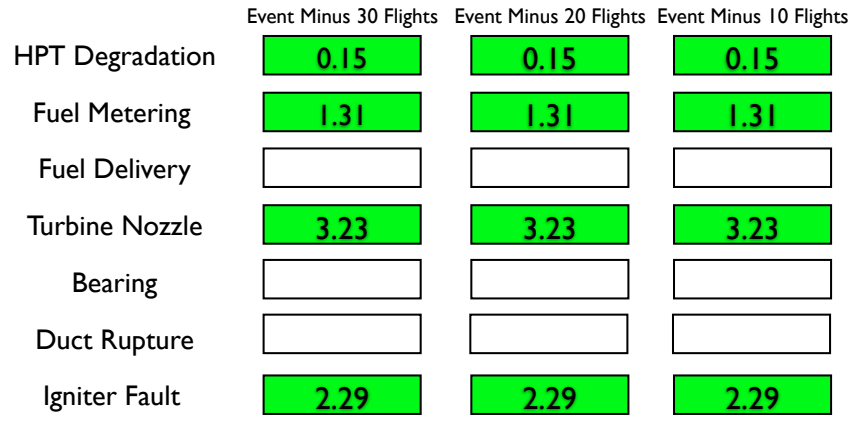

Figure 12: Trace of the Reasoner on the Original Reference Model

Experiment 3 was directed to verifying the improvement in the reasoner performance

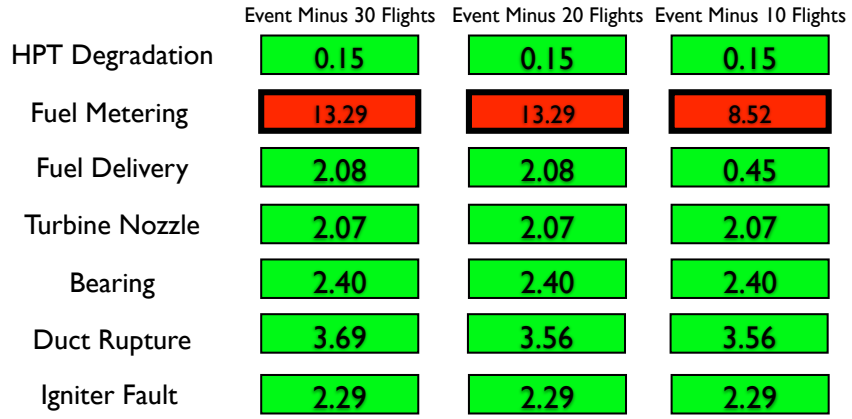

Figure 13: Trace of the Reasoner with the improved Reference Model

with the updated reference model. These results from the reasoner simulations are shown for the the original reference model in Figure 12 and the augmented reference model in Figure 13. The traces illustrate the reasoner's inferences through a progression of flights before the incident occurred. A green shade on a failure mode indicates that there is a likelihood of the fault given evidence and the number in the box indicates the calculated relative likelihood value. A failure mode shaded red, indicates a high likelihood for that hypothesis, and when the failure mode is bolded, "Fuel Metering" in this case, it indicates that the failure mode has been isolated with very high likelihood, and this mode is added to a report for the mechanics. In this case study, the red indicator appeared about 30 flights before the adverse event, which would give the mechanics a number of opportunities to avoid the adverse event occurrence. Verification experiments of this kind are critical not just to establish the fact that the early detection metric is improved, but also that the new information added is not creating side-effects, such as increasing the number of potential diagnostic hypotheses, and complicating the mechanics decision making process. In this case, the expert deemed the verification test a success, and the updated reference model was accepted.

IV.5.2 Case Study 2

The second case study discusses a broken turbine bucket blade fault called the "HPT degradation" failure mode. The broken turbine blade resulted in excessive vibration that resulted in an engine shutdown that resulted in an emergency landing of the aircraft. This failure, still associated with an engine, was chosen because of its physical difference from the Fuel HMA failure. The results of the three experimental steps are presented, and then additional experiments were conducted to show that the false alarm rates would remain low, even when the Fuel HMA and HPT degradation faults were compared.

IV.5.2.1 Experiment 1

This case study used the same CIs as case study 1 and employed the same 10-fold cross validation framework on the 50 flights that led to the engine shutdown incident. The result was an average accuracy of 92.18% and a false positive rate of 2.1% for the derived TAN classifier.

IV.5.2.2 Experiment 2

The same binning procedure as case study 1 was applied, and the results for this experiment are shown in Tables 6 and 7. Results from Bin number 2 were chosen for updating thresholds and looking for new monitors. The StartTime monitor indicating a slow start turned out to be the most important monitor for fault detection and isolation. There was no overlap between the thresholds found here with case study 1. This means that in spite of shared monitors, the non overlapping thresholds would not result in ambiguity of fault hypotheses.

The expert found that the structures of the TANs generated by binning were very similar, and, therefore, the decision was to focus on the TAN generated from all 50 flights. From the structure shown in Figure 14, the expert focused on the connection between resdTemp, the residual temperature of the engine at shutdown, and the peakEGTC, which is

Bin	Flights	Acc.	FP%
1	1 to 10	90.625%	4.2%
2	11 to 20	92.50%	2.5%
3	21 to 30	87.5%	5%
4	31 to 40	88.125%	12.50%
5	41 to 50	85.625%	11.7%

Table 6: Accuracy and False Positive Rate for Classifiers Created from Different Data Segments for Case Study 2

Bin	Flights	Obs. Root Node	Children of ORN	Notes
1	1 to 10	StartTime(Slow start)	Every CI	Thresholds Chosen from this Bin
2	11 to 20	StartTime(Slow start)	Every CI	Similar Structure
3	21 to 30	StartTime(Slow start)	Every CI	Similar Structure
4	31 to 40	StartTime(Slow start)	Every CI	Similar Structure
5	41 to 50	StartTime(Slow start)	Every CI	Similar Structure

Table 7: Observational Root Node and Immediate Child Node for Classifiers Created from Different Data Segments for Case Study 2

the peak temperature of the engine after startup. This relationship with the casual direction would imply that the residual temperature is causally related to the peak engine temperature. The expert decided that this was most likely a relation between the resdTemp of flight n and the startup temperature in flight $n+1$. The expert used this relation to design a super monitor that indicted the fault, if and only if, the high temperature monitors associated with resdTemp of flight n, fired, and the high temperature monitor connected to peakEGTC of flight $n+1$ were also indicting the fault. Therefore, this super monitor captures some temporal information between flights for diagnostic reasoning. Like before, the updated reference model included updated thresholds and the new super monitor.

IV.5.2.3 Experiment 3

Utilizing these changes, we ran this scenario with the reasoner and the augmented set of monitors. From the trace generated by the updated reference model (Figure 15), 12 flights

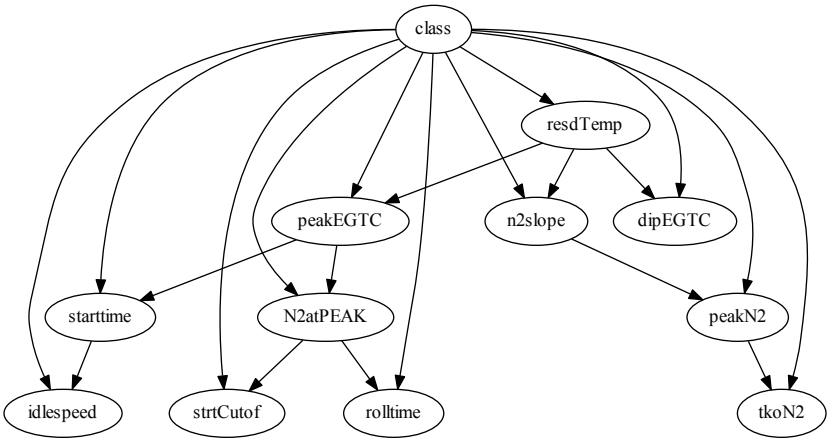

Figure 14: TAN Structure Generated using Data from Case Study 2

before the adverse event occurred there were symptoms pointing towards degradation in the high pressure turbine(HPT). The failure mode listed as "HPT degradation" is an aggregate term that captures loss of turbine function and includes the broken turbine blade fault. Typically this would trigger a maintenance request wherein the mechanic would use a special camera called a borescope to visually inspect the damage and determine if this should result in an engine removal action to avoid safety incidents in future flights. The maintenance procedures would result in replacing the broken blade before the engine was put back into operation.

While the HPT degradation was hypothesized 12 flights before the adverse event, its likelihood increased progressively through subsequent flights and eight flights prior to the event it became a highly likely candidate. However, the fault condition was not uniquely isolated because another failure mode, "fuel nozzle clogging" was also a strong second candidate. Our domain experts surmised that the reasoner would have generated a maintenance alert about eight flights before the adverse event, although it could not uniquely isolate the problem. A borescope inspection following this alert would have clearly identified the broken turbine blade.

Figure 16, shows the trace without the augmented reference model. The baseline case of the monitors merely creates a bearing failure mode fault, and this is detected just before the flight where the adverse event occurred. In this case the maintenance crew would take action, but for an incorrect fault hypothesis. The check on the HPT blades may never have occurred before the adverse event. In contrast, the augmented reference model produces a relevant alert eight flights prior to the event, which would give the mechanics ample opportunity to take action that would avoid the inflight engine shutdown event.

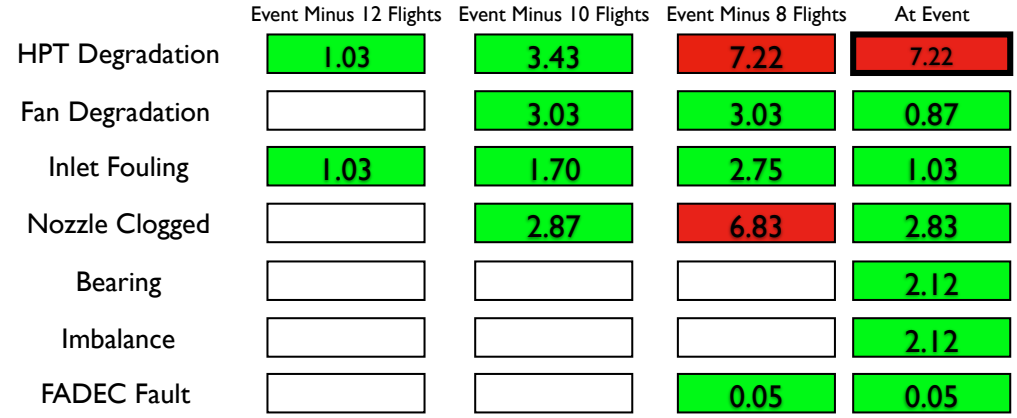

Figure 15: Trace of Data from Case Study 2 with the Reasoner using the Augmented Reference Model

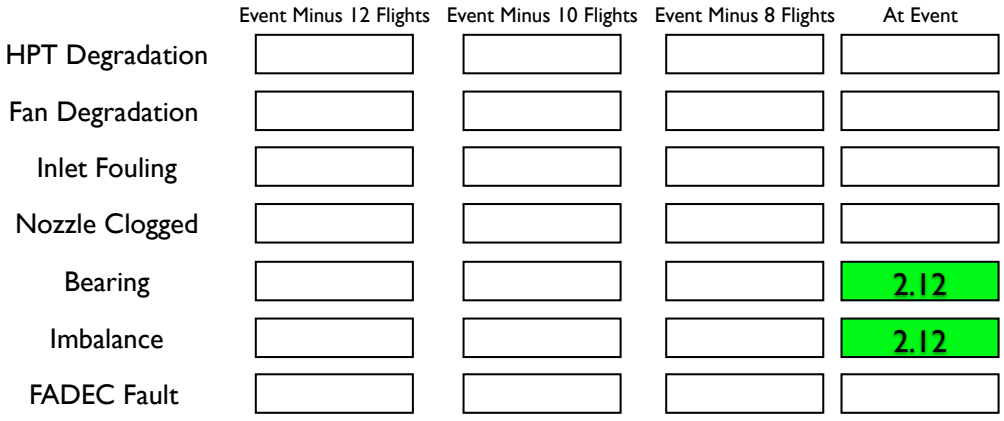

Figure 16: Trace of Data from Case Study 2 with the Reasoner using the Original Reference Model

IV.5.3 Robustness Experiment

With the reference model updated for two faults, we decided to run a robustness experiment to check the performance when comparing one fault against the other. When we used the TAN classifier generated using the Fuel HMA data, an experimental run with the Turbine Bucket Blade(TBB) fault data was classified as nominal with 95.93% accuracy and a false positive rate of 4.10%. The TBB TAN achieved 85% accuracy with a false positive rate of 15% when the experiment was conducted on the Fuel HMA fault data. This showed that the Fuel HMA TAN was tuned to detecting the Fuel HMA fault without increasing the false alarm rate, but the TBB TAN was less precise. The expert concluded that in this case, additional CIs, such as a vibration detector, was necessary to better isolate the TBB fault. This second case study establishes the generality of our approach across faults in the engine subsystem. It also shows that robustness analysis is another tool that helps the expert understand the nature of the failures and the feature sets being used to distinguish between those failures.

IV.5.4 Case Study 3

The third case study investigates a fuel manifold leak fault that also caused an engine shutdown event in flight, leading to an emergency landing. The fuel manifold leak is not associated with a single engine subsystem, rather it contains the fuel lines that supply two of the four engines of the aircraft. This failure also impacts the engines, producing similar effects to other failures, however, our analysis helped the expert determine that this fault was not associated with one of the engine subsystems, and, therefore, should be analyzed using the system level diagnoser. We show that using the process developed along with the robustness analysis.

IV.5.5 Experiment 1

The experimental set-up provided an accuracy value of 90.31% and a false positive rate of 5.4% using 10-fold cross validation on the Fuel Manifold TAN. Utilizing the other two datasets as the test set reveals more about this fault. The Fuel HMA data has a 77.5% accuracy and 22.5% false positive rate, which is a much weaker result. The broken blade failure scored worse with an accuracy rate of 44.4%. This means that the Fuel Manifold TAN is not serving the purpose of differentiating between its own failure and the blade failure. On further reflection, the expert realized that this failure could not be reliably isolated at the engine subsystem level.

This case study reveals that the data mining methods are useful not only for finding additional relations and monitors to augment subsystem reference models, but they also provide useful indicators to knowledge engineers and system experts, when the approach being used is not a good fit for the fault being analyzed.

IV.6 Conclusions

The supervised data mining method employed for improving existing diagnostic reference models for aircraft derives Bayesian TAN classifiers from selected segments of aircraft flight data, and with the help of domain experts, augment the existing Aircraft Diagnostic and Maintenance reference models by : (1) updating threshold values on monitors, (2) discovering new monitors, and (3) combining monitors to build super monitors to improve overall diagnostic performance. Experiment 3 in Studies 1 and 2 demonstrated that the knowledge engineering processes that combines supervised learning with the expert interpretation and updates not only improved fault isolation capabilities, but fault detection times are reduced in the flight sequence, thus aiding mechanics in their decision making tasks for maintenance and improving overall safety by mitigating the occurrence of adverse events. Case Study 3 demonstrates how the classifier performance alerts the knowledge engineers and experts through TAN classifier accuracy and false positive rates, showing that

the fault under consideration does not fit the reference model structure that is being used. This led the experts to better understand the nature of the fault, i.e., the fault was at the system level as opposed to the engine subsystem level.

It is important to note that this method is developed with the rarity of known failure data in mind. It may be difficult to find enough data to build a robust classifier that works across a number of different single faults. The consequence is that the generality of the classifiers in this method across multiple faults is hard to test. Obviously as more data is collected, the more confident an expert would be about the results, and the tighter the discretization used in the CPTs, deriving even better thresholds and the avoidance of false alarms and misclassification errors. One augmentation that cannot be done reliably is the augmentation of probabilities used directly in the reference model by the reasoner. This is because it requires a formalization of the necessary size of the dataset, to account for the a-priori probability of a failure being present.

Whereas the knowledge engineering approach has produced successful results that are practically useful but conceptually of limited applicability, we extend our data analysis methods to more open approaches. These approaches are scalable in bit data, and have the potential to discover previously unknown faults and anomalies in complex systems.

CHAPTER V

EMPIRICAL STUDIES OF DISTANCE MEASURES FOR DIMENSIONALITY REDUCTION OF TIME SERIES DATA

Chapter IV addressed supervised anomaly detection over large data by exploiting expert knowledge to identify relevant flight data and simplifying the detection problem to a single fault versus no fault analysis. In this case, the data curation process was well-defined and we demonstrated a successful approach to building accurate models by producing new knowledge for experts to analyze and incorporate this knowledge into existing diagnostic reference models. The obvious disadvantage of the previous method is that it is limited to known adverse or faulty situations. This constrains the data to small subsets which are effective in supervised learning, but do not easily scale to dealing with multiple single fault situations, and leave a large portion of the curated data unused.

We adopt another approach that uses exploratory analysis methods such as clustering to detect and characterize previously unknown anomalies in our problem domains, which results in new information data mined from large amounts of unlabeled data. Our approach first identifies new anomalies in the data. We then characterize the anomalies by comparing them against a nominal set to provide insight into the set of features that best differentiate the anomalies from nominal situations.

This approach encounters the challenge of large dimensionality of the data. Our problem domains produce data with many instances, many features and temporal signals for each feature. Specifically, the time series dimension of our data interferes with the use of several unsupervised techniques for exploratory data mining. In order to leverage this unused data we simplify its time series dimension. We face another challenge in finding ways to extract features and transform the data into organized datasets, while still maintaining information about the temporal characteristics of each feature. Our solution to this challenge

must not only identify anomalies but also characterize them to help the expert understand possible root causes and learn from these anomalous instances. Our approach for helping the experts characterize the new anomalies is to extract the features that best differentiate the anomalies from the common features found in the clusters that characterize the nominal set. This means the methods we use to simplify the temporal sequences maintain the original feature information.

This chapter focuses on the dimensionality reduction task of the time series dimension. This task starts with the three dimensional structure of the data cube described in Section III.3.1 and reduces it to a structure of dissimilarities of each pair of instances, for each original feature. This structure is then reduced to a two dimensional dissimilarity matrix comparing each instance, which we apply to traditional unsupervised learning algorithms. The chapter primarily focuses on studying different dissimilarity (distance) measures that can be used in our approach for reducing the temporal signals, specifically for exploring the aircraft flight systems and baseball domain. We explore possible choices for these distance measures and perform experiments that focus on the properties of the different measures when comparing different forms of temporal signals. We intentionally use well-defined mathematical functions, so we can make systematic comparisons. The distance measures chosen are representative of different approaches that have been employed to analyze time-series data and the experiments are designed to test how distances react to the parameters of the chosen test set.

The rest of the chapter is organized as follows. Section V.1 outlines the approach for measuring the complexity of the time series aspect of the data cube and reducing it to a form where traditional clustering algorithms can be applied. Section V.2 describes the different dissimilarity measures and justifies the choices of complexity measures considered in this work. Section V.3 describes the test suite of data and evaluation method used for performing these experiments. Section V.4 presents the results of the test suite and identifies the best performing dissimilarity measures. To further demonstrate the effectiveness of

the approach, Section V.5 takes the best performing measures and applies them to a labeled multivariate time series dataset. The ability to classify the data objects provides us with an indication of the effectiveness of these measures in analyzing real data. Section V.6 provides a summary of the experimental results, and picks the best performing dimensionality reduction method for application to our two primary problem domains in later chapters.

V.1 Dimensionality Reduction Approach

Considering the data representation modeled in the data cube defined in Chapter III, there are two ways in which the data cube can be simplified:

1. by reducing the number of features in the data cube, and

2. by reducing feature descriptions that are represented as time series data.

Our overall goal is to apply unsupervised techniques to identify anomalous instances. An approach to achieving the first type of reduction is performed where researchers unroll the data cube, so that each sample of each feature is transformed into a feature itself [94]. This unrolled cube in two dimensions and each instance now possesses M times T_M features, where M represents the dimensionality of the original feature space. Principal Component Analysis is then applied to build a reduced and orthogonal feature space. From this new, shortened dataset, the Density based clustering [94, 98] is applied to identify outliers. This approach is effective at reducing the dataset size but possesses downsides for our data. The unrolling transformation and the nature of PCA means each time sample for a signal is assumed to be independent of the rest of the signal points. Also, any correlation between the original features is removed. There is definitely a loss of information for each feature value because the time correlations are ignored. The reduction of the feature dimension in to a new set also impacts the ability to quantify the relevant features for an anomaly since the feature space has lost the semantic meaning during both the unrolling and the PCA transformation.

The second class of approaches are applied to compress the data along the time series dimension. The reduction of this dimension retains the instance information, and allows for characterization of the anomalies in the original feature domain without loss and transformation of the feature set. Our approach is to reduce the time series information into as small of a number of values as possible. This is balanced with maintaining as much information about the signal's temporal characteristics. In other words, we may assume that the method reduces the complexity of the signal. Our approach is based on using measures of complexity to define signal characteristics by a small vector of values or evan a single value.

The complexity measures studies can be used to characterize a signal in terms of a small set of discrete values such as wavelet coefficients or they provide a mechanism for computing between pairs of signals. Pairwise differences between these new reduced values are defined as a dissimilarity metric. Primary metrics include the Manhattan, Euclidean, and Mahalanobis metrics [71]. Not all distance based measures have metric properties ,i.e., they satisfy the triangle inequality.

Given the measurement chosen to represent the time series, the dataset is reduced to either a two dimensional data set or a cube made up of distance matrices. Single signal reductions produce a dataset made up of the instances and the complexity measurements for each of the features. When a dissimilarity measure is utilized, our data cube is transformed into another similarly shaped cube, as shown in Figure 17. This is now as series of M matrices, each of dimension NxN instances. The final reduction of the cube to a data set for cluster studies combines these matrices into one dissimilarity matrix for analysis.

Our approach is to apply a modified form of the Euclidean distance to each pair of instances in the data. We define this as

$$mED(D) = \sqrt{\sum_{m=1}^{M} \omega_m * D_m^2}$$

, where D is the vector of dissimilarity measures for each feature, for a given pair of instances i and j. The value ω_m normalizes the pairwise distance by the max distance for any pair in the distance matrix for the particular feature. The weighting of each distance in the vector ensures that each feature is represented on an appropriate scale. Applying the function to every pair of instances reduces each pairwise distance to a single value and the cube into a two dimensional matrix.

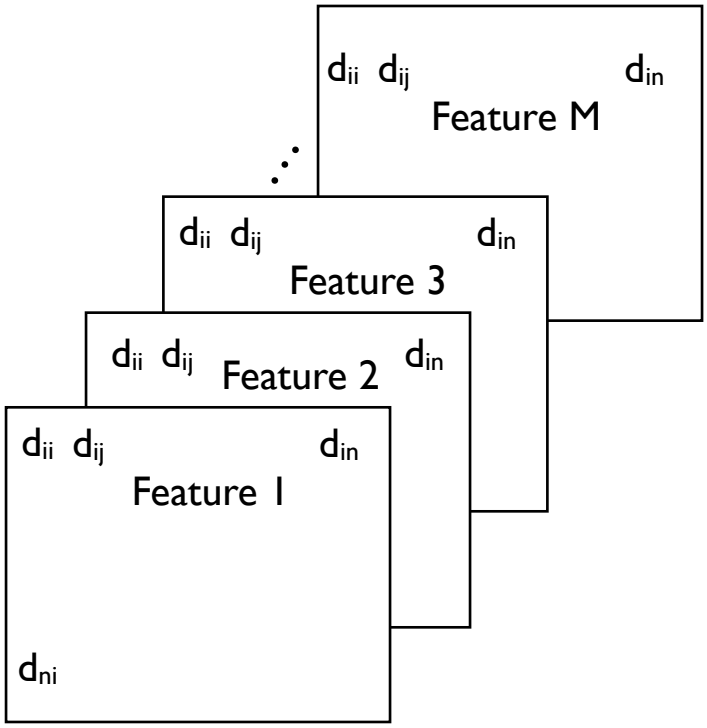

Figure 17: Example of a Data Cube of Dissimilarity Measures

V.2 Background on Complexity Measures

There are many choices for computing the complexity of temporal signals for comparison and modeling. In anomaly detection, there are needs that these measures must satisfy to be effective. Primarily, the measure needs to be sensitive to the important differences between signals, but not be overwhelmed by unnecessary details. Methods can be

as simple as Autoregressive Model Order Estimation [69], which measures the complexity in a signal by the coefficients, i.e., the order of the regression function, in the polynomial function model of the temporal signal that minimizes the mean square error estimate [75]. While this method is easy to apply to our data, it may not have enough power to isolate small differences in signals of the same type (such as two quadratic functions). Secondly, in complex system domains, where operations may generate large amounts of data, the measures should be relatively efficient in its computation. The Embedded Space Eigen Spectrum [65] looks at the fractional spectral radius in a pre-determined set of n eigenvalues found after decomposing the set of embedded vectors that capture a signal over a time window. This method has great power in locating differences, but the computation necessary for eigenvector decomposition is an obstacle to using for dimensionality reduction.

We investigate a number of choices for measuring this complexity using compression, information theory and signal analysis methods. Our choices include Approximate Kolmogorov Complexity, Complexity-Invariant Distance, Approximate Entropy, and the Haar Wavelet transformation for representing a signal.

V.2.1 Compression-based Methods

Compression based methods utilize compression algorithms such as the standard bzip algorithm that takes a data string and reduces it to a smaller string for storage purposes [50]. Compression algorithms are designed to take advantage of redundant information in the original string and reduce that information into a smaller sequence that can be reconstituted at a later time. Lossless algorithms produce compressed strings which when decompressed are identical to the originating input. Lossy algorithms, on the other hand, may ignore information in the string during compression, and a decompressed string will be similar, but not exactly the same as the original. Lossless algorithms are important for text and sensitive data that requires precision to use. Lossy algorithms are used for efficient

transmission of images and multimedia where a loss in precision can be tolerated. Compression algorithms are most effective if the data can expressed as a repetitive pattern. The simpler this pattern, the smaller the size of the compressed string. In time series data, if the signal can be described as a repetition of a basis function, it can be compressed more effectively than a signal with a more random sequence. For time series data, we want to use compression for identification of when the signal is more complex and than another signal with more pattern based behavior. There are two primary methods we examine that use compression methods to compare strings of data: Approximate Kolmogorov Complexity and Complexity-Invariant Distance.

V.2.1.1 Approximate Kolmogorov Complexity

Kolmogorov complexity (KC) [77] defines the complexity of a signal (an ordered string of values) as the smallest Turing machine that can reproduce that signal [82]. Complex signal patterns require longer program segments to recreate the pattern, as compared to simpler and repeating patterns. However, Turing Machines are a theoretical construct, and methods to compute exact KC values for an arbitrary set of signals is non-existent.

A practical alternative is finding reasonable approximations of the measure using compression algorithms that can be applied to compacting the data. Similar to the KC theory, a complex signal will require more space even after compression, much like a longer Turing machine would be necessary to produce the same signal. The measures are best used with lossless compression algorithms so that information in the data is not lost during compression. The memory footprint of a signal after lossless compression can be used as the approximate measure of the complexity for that signal. A compression algorithm that captures relevant information from the data using a minimal footprint is a more accurate description of the KC approximation. An approach to understanding how efficient a compression algorithm is for a given signal may involve a comparison of the compression result

for that signal compare to other signals. Therefore, the Approximate Kolmogorov Complexity is represented as a relative measure, i.e., a measure of the pairwise distance between two signals, which measures the comparative complexity of one signal given another.

There are a number of distance measures that have been employed to compute approximate Kolmogorov complexity, including Normalized Compression Distance [32], the Chen-Li Metric [144], the Compression-based Dissimilarity Measure [77] and Compression-based Cosine distance [144]. Most of these dissimilarity measures initially appear to be different, but after careful examination they can be deconstructed to show that they have the same basic structure [144].

The dissimilarity metrics assume a compression function C that compresses a string and for any pair of strings x and y, we may compute a number of parameters $C(x)$, $C(y)$, $C(xy)$, and $C(x|y)$. The output of $C(x)$ or $C(y)$ is the compressed lengths of strings x and y. We measure this compression in bytes of storage for the compressed strings. $C(xy)$ is the number of bytes required to store the concatenated string "xy" and $C(x|y)$ is the compression of x, using the compression profile of y. This last measurement is semantically similar to the original idea behind Kolmogorov-Complexity, where one string forms the basis for finding the information in the other. The nature of compression algorithms does not guarantee that the corresponding dissimilarity measure satisfies the symmetric property as $C(xy)$ may not equal $C(yx)$.

A number of different dissimilarity metrics may be derived from the primary compression measures, but a few have become popular in the literature:

1. the Compression-based Dissimilarity Measure (CDM), and

2. the Normalized Compression Distance (NCD).

CDM measures the dissimilarity between signal x and signal y as

$$CDM(x,y) = \frac{C(xy)}{C(x)+C(y)}$$

. This value is bounded by the interval $[.5, 1]$, where 1 implies complete dissimilarity and .5 represented the case when the two signals are identical. This measure does not satisfy the triangle inequality.

NCD measures the dissimilarity as

$$NCD(x,y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}}$$

. The values for NCD are bounded by the interval $[0, 1]$, where a value of 1 implies complete dissimilarity. Unlike CDM, the interval is twice as large, with 0 representing complete similarity between signals. This larger interval, and the fact that NCD has metric properties:

1. $NCD(x) = NCD(xx)$, and
2. $NCD(xy) = NCD(yx)$, and
3. $NCD(xz) \leq NCD(xz) + NCD(yz)$

makes it a more appealing dissimilarity metric.

Moreover, it has previously been employed by clustering applications [32]. From this information, we use the NCD measure over CDM in our experiments.

V.2.1.2 Complexity-Invariant Distance

Approximations of Kolmogorov Complexity attempt to identify the distance between two signals purely based on the complexity as a form of compression. The Complexity-Invariant Distance Measure (CiDM) was built to addresses shortcomings in that approach for time series signals [6]. The CiDM was built with time series data in mind, and instead of using only compression to find the dissimilarity, CiDM uses compression as a way of normalizing the Euclidean distance. The CiDM is defined as:

$$CiDM(x,y) = \frac{(ED(x,y) \times max\{C(x), C(y)\})}{min\{C(x), C(y)\}}$$

and the Euclidean Distance is defined as:

$$ED(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

This measure is invariant to complexity, which is to say that it is designed to account for the fact that signals of the same general type with issues such as noise contain varying levels of complexity. By normalizing the Euclidean distance by the measured complexity, CiDM removes this variance and measures the signals more effectively. Considering our approach is geared towards anomaly detection, in which the domain will remain fairly consistent, this seems an appropriate choice.

V.2.1.3 Compression Algorithms

The choice of compression algorithm and the ability of the algorithm to work with certain types of data can improve the efficiency of the compression. As mentioned for the KC Complexity, this choice of compression can make the measure of complexity a more accurate approximation. Given the possibility that the data to be compressed could be noisy, we want our compression method to be sensitive to differentiating among signals, and at the same time, robust to noise. Note that a signal that is pure noise, i.e., sampled from a random distribution, will have high compression values masking the contribution of the actual signal.

The first compression algorithm chosen comes from the DEFLATE family of algorithms. This algorithm is an implementation known as DZIP [50]. The DEFLATE algorithm is based on a two step process of running the LZ77 compression algorithm [10] and then using Huffman Coding [80] to find a compact representation. LZ77 is a sliding window based compression algorithm. The operation of LZ77 is to locate a pattern of a specified window length, and replace all occurrences of this pattern except for the original with two markers. The first points to the index of the original pattern and the second

describes how long the pattern was repeated from the markers placement. The output of LZ77 is a series of original patterns interspersed with markers for these patters as repeated. Huffman coding is then applied to the reduced signal to find a compact representation. The implementation of DZIP specifically deals with finding the appropriate window length, and reducing the overhead of storing the Huffman code trees in the data string.

The second compression algorithm we used is the extension of the Lempel-Ziv compression of LZ77 called Lempel-Ziv-Welch (LZW) [109, 143]. LZW is used in GIF image compression and has a hardware implementation. Simply, LZW is similar to the window-based method of LZ77, but attempts to encode the pattern representation of LZ77 using an incremental bit representation that mimics the Huffman Coding. This choice of compression is to provide a similar approach to the DZIP implementation of DEFLATE.

A third algorithm is prediction by partial matching (PPM) [33, 176]. PPM diverges from the LZ77 base that the previous two compressors utilize for pattern decomposition. PPM counts the original string's symbols and uses probabilistic methods to find the most common repeated patterns. The more common the pattern, the smaller the number of bits to represent it in the compressed representation. More complex strings will have fewer higher probability patterns and require a less compact compression. These values are used during decompression to produce the next symbol in the decompressed string by using the current portion of the decompressed string as the predictive value.

Our final compression algorithm is the Burrows-Wheeler transform (BWT) [19, 104]. BWT operates more as a front end to a compression algorithm, by pre-processing the string before another technique compacts the string representation. This preprocessing is a sorting algorithm. This sorting rearranges the string to a format that isolates as many patterns as possible, making the compression of the string more efficient.

The number of different ways in which we compute our compression-based complexity methods of a signal is the Cartesian product of the set of distance measures that are chosen and the possible compression algorithms used on the signals. This leaves us with 2 Distance

Measures and 4 compression algorithms, for 8 possible choices. A compression algorithm and accompanying distance metric may be better suited for certain kinds of signals, and our initial experiments help discover "the best" compression methods and distance measures. The empirical studies help to determine the properties of the compressions algorithm, and then determine the one that best matches our needs for comparing time series.

V.2.2 Approximate Entropy (ApEn)

ApEn [65, 126] is a measure that reduces the entire signal to a single value, representing how much entropy it contains over time. This is in contrast to the compressor methods which utilize a pairwise distance and measure the values of compressed versions of the signals. Entropy is a probabilistic measure from information theory linked to information gain or information content. A measure of information content is the number of bits needed to encode a signal for transmission. The more predictable the signal, the smaller the number of bits required to encode information. Pincus defines ApEn formally as the measure of "the likelihood that runs of patterns that are close for [a number of] observations [in the signal] remain close on the next incremental comparison. [126]" The ApEn measure examines the change in entropy of the signal over increasing windows of time, and computes the increase in entropy as the window size is increased over a signal of length N. The function to calculate ApEn has two parameters, a starting window size m and a tolerance r which helps determine if two sections of data are similar enough. This function is defined as

$$ApEN(m,r) = \phi^{m+1}() - \phi^{m}()$$

where

$$\phi^{m}(r) = (N-m-1)^{-1} \sum_{i=1}^{(N-m-1)} ln(C_i^m(r))$$

where ln is the natural logarithm and $C_i^m(r)$ is the measure of the frequency of patterns of size m which are similar, within a tolerance r, in the data starting from point i.

ApEn has been used to help classify biomedical signals, such as EKGs and respiratory responses [86] [148]. Recently it has been applied to analyzing noisy vibration signatures for diagnosis problems [172]. The accuracy of the ApEn measure increases with the window length (i.e., large number of data samples), but empirical studies show that it is effective for sequences of the order of about 70-100 samples.

The output of ApEn is a single value for a signal. The compressor methods discussed earlier compute distances by pairwise comparison of signals. ApEn is not a distance measure by itself. We calculate a distance, so that it can be compared with the methods above, by computing the distance between two signals as the absolute value of the difference between the ApEn of the two signals.

V.2.3 Wavelet Based Representation

A number of interpolation methods have been developed for representing signals, such as Fourier transforms [15], iterated function systems [107] and Wavelet transforms [38]. We have selected Wavelet transforms as a complexity measure because when applied continuous-time signals (both discrete and continuous) the transform returns sets of scaled components for each signal that capture the temporal and spatial properties of the signal. Wavelets transforms offer advantages over Fourier transforms, which are purely frequency based, because they are localized in space. This localization means scale components can return a smaller number of components for the same function, making them ideal for compression, and removing noise [165]. Wavelets are also faster to compute than methods using iterated function system approaches. The complexity measure from the wavelet representation will be based on the number of components, as well as the values for the components.

Given a signal represented as an ordered series of values $[a,b,c,d]$, the calculation of a wavelet transform begins with producing coefficients that correspond to the average, e.g., $[\frac{a+b}{2}, \frac{c+d}{2}]$, and coefficients that correspond to the difference, e.g., $[\frac{a-b}{2}, \frac{c-d}{2}]$. The application applies the same two transformations to only the coefficients of the average.

This occurs until the size of coefficients of the average is a single value. The transform then collects the set of coefficients for each average and difference. This set can be used to reconstruct the signal, and these coefficients can be compared to another signal, to see how similar the scaled components are to one another and to provide a relative basis of complexity.

Similar to ApEn, the output of the wavelet decomposition is not a distance but a reduced definition for a signal. In this case, the wavelet produces a series of coefficients for the signal. An Euclidean distance measure can again be applied to the corresponding coefficients of a pair of signals to produce a distance measure comparable with the one for ApEn and the compressor methods. For our work we use these measures to reduce the dimensionality of our data cube by transforming the temporal dimensions of each feature to pairwise comparisons of these signals that produce a measure of dissimilarity between pairs of instances.

V.3 Experimental Approach of Studying Dissimilarity Measures

We have presented three different approaches that are commonly used for measuring the complexity of time series signals and reducing them to methods for pairwise comparisons of signals. As a next step, we run a set of empirical studies to compare the properties of the different approaches by applying these measures to a range of signals, as well as discovering how specific these measures are at discerning differences within a given type of signal. Our experiments focus on a set of artificial temporal signals that correspond to first and second order dynamics. We are primarily interested in studying:

- the ability of the measures in differentiating between signal types, and
- the sensitivity of the measure to parameter changes within a type of signal.

We explore these differences in each measure by creating a series of experiments. We first create a test data suite to examine a number of common signal types, and to explore

the differences within each type across a set of values for selected parameters. We apply our chosen complexity measures across the data suite and look for how well each measure differentiates the different signal types using a classifier, and then measure how well each measure differentiates the different parameters within each signal type. The goal is to select a smaller number of measures that can be applied to our problem domains and used to identify anomalies in real data. We finish our experiments by taking the best choices of complexity measures from the test suite and applying them to a multivariate data set produced from real world measurements. This data includes known labels to identify each instance. We explore how the selected measures classify and cluster the data. This provides insight into their effectiveness at identifying similar behaviors.

V.3.1 Test Data Suite

We have constructed an experimental test bench that includes a selection of the primary signals characterizing basic dynamics: linear, quadratic, sinusoid, and a sinusoidal signal with exponential attenuation. These signals are illustrated in the plots shown in Figure 18. The linear and quadratic signals have similar parameters, as do the two types of sinusoidal examples. The experiments conducted study the sensitivity of the distance metrics to the parameter value changes in the signals of a particular type. Comparisons are also made across the different types of signals.

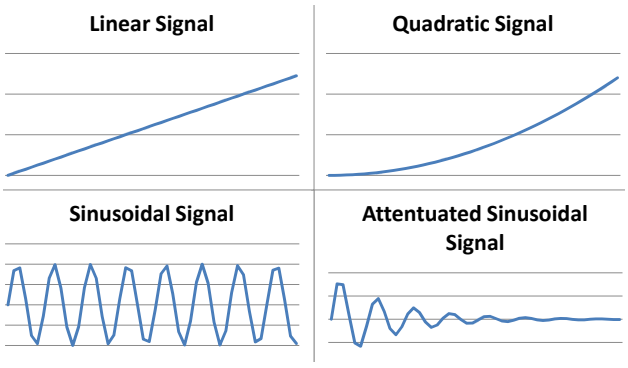

Figure 18: Example Plots of a Signals for Test Data Suite

The following parameters are varied for each signal type:

1. Slope and y-intercept for the linear time-varying signals,
2. The coefficient of the quadratic and linear terms for the quadratic signals,
3. The frequency and phase of the sinusoidal signals, and
4. The frequency and attenuation of the attenuated sinusoidal signals.

For small values of the quadratic parameter, the quadratic signals should be more similar to the linear signals, similarly for small attenuation values, the sinusoidal and attenuated sinusoidal signals should be quite similar.

Our goal is to identify choices for these measures and metrics for use in anomaly detection schemes, and these different studies of the signals help identify their general abilities. Specifically, these studies test the ability to separate different signal types ,e.g., linear from quadratic, and to also understand how sensitive the measures are to parameter values changes in the same signal type.

With the exception of the attenuation parameter, the parameters were chosen to be $[1, 10, 50, 100, 500]$. These values include a range of magnitude and multiplicative differences. The values of the attenuation parameter were chosen from the set $[.001, .01, .01, .5, 1]$, which allows the attenuation to be very small and therefore, similar to non-attenuated sinusoidal behavior, and then a much bigger attenuation of the signal at the other end of the spectrum. Since each signal was defined by two parameters, we ran 25 experimental combinations for each signal. The dynamic functions and the parameters values used for these experiments are summarized in Table 8.

In addition to the comparison of the idealized ,i.e., noise free signals, we also made comparisons of noisy signals by varying the signal-to-noise ratio. The noise model was Gaussian with zero mean, and variance values were either 2% or 10% of the signal magnitude. These comparisons of the noisy signal experiments to study the robustness of the measures are also made over multiple trials.

Function	Values for k_1	Values for k_2
$\dot{x} = k_1 * t + k_2$	$[1, 10, 50, 100, 500]$	$[1, 10, 50, 100, 500]$
$\dot{x} = k_1 * t^2 + k_2 * t$	$[1, 10, 50, 100, 500]$	$[1, 10, 50, 100, 500]$
$\dot{x} = \sin(k_1 * t) + k_2$	$[1, 10, 50, 100, 500]$	$[1, 10, 50, 100, 500]$
$\dot{x} = \exp^{k_1 * t} \sin(k_2 * t)$	$[.001, .01, .01, .5, 1]$	$[1, 10, 50, 100, 500]$

Table 8: Functions and Parameters Used for Test Data Suite

V.3.2 Structure of the Experiments with the Test Data Suite

Our experiments with the test suite are divided into two parts. Part 1 examined the compression-based distances with the different compression algorithms. From this analysis, we selected the best combination of distance and compression algorithm from the eight possible choices. Part 2 of this study compare the best compression based measure against the two other measures, ApEn and Wavelets. The combination of the experiments produces a selection of the top choices for complexity measures. As a last step, we then apply the chosen measures to a real world set that we acquired from the UCI Database.

The purpose of these experiments are to understand the effectiveness of the complexity measures and to do this we

1. study the properties of the different complexity measure in differentiating between the signal types, and

2. further study the ability of each measure to differentiate between signals of the same time as the parameters are varied.

Before the start of either study, for each distance measure under consideration, we build a distance matrix for each pair of signals in our data. This produces a matrix of 100 by 100 distances.

V.3.2.1 Differentiating Across Different Signals

The study of properties in the complexity measure are based on examining how well the distances separate the different classes of signals (Linear, Quadratic, Sinusoid, and

Attenuated Sinusoid). Taking a cue from previous work [77], we chose a one-*nearest neighbors classifier* to compare the distances across all the signals in an effort to gage the discriminating power between the signal types. The results are compiled into a confusion matrix. We examine overall accuracy for the signals, and then study the confusion matrix in more detail to identify the weak points in the classification. This evaluation is in terms of the noise-free and noisy (two levels) signals.

V.3.2.2 Differentiating Within a Signal Type

Our examination of the measures abilities to differentiate the varying parameters for each signal, requires measuring attributes about each signal and the distances between each parameter choice. For each signal type, we examine how the distance measure varies as one of the parameters is progressively increased. For example, we picked a linear signal corresponding to slope 50 and a fixed intercept as the based signal S_1, and computed the distance from all other linear signals, i.e., those with slopes, 1, 10, 100, and 500 and the same fixed intercept with S_1. A similar experiment was conducted with linear signal S_1 having intercept $= 50$ and a fixed slope, and then comparing linear signals with different intercept values, but the same slope value.

Examining these trends, we produce two measurements to help identify behaviors of the complexity measures. When examining the compression based methods, we measure how **sensitive** the distances are for the range of parameters across the different compression algorithms. Secondly, we record for all complexity measures, whether the distances in the trend are **monotonic**.

Sensitivity of the measure is linked to the difference in the parameter values for the two signals. In other words, the difference in the measure should be proportional to the absolute difference between the two parameters that vary for the signal. The larger the proportionality, the more sensitive is the complexity measure for that parameter, and the better it is at differentiating that parameter space of a signal. This is useful to identify in

the measure, since our anomaly detection may be looking at two signals that appear to be generated by the same dynamic system, but whose coefficients are different. Less sensitive measures are likely to generate false negatives.

The sensitivity measure is application to comparison of measures of the same type, because two measures may have very different scaling factors, thus there is no pre-determined framework in which to compare the values of the measures. Therefore, we do not repeat this comparison for the ApEn and the Haar Wavelet distances.

Due to the number of experiments run, specifically for the noisy signals, we describe the sensitivity of signals in the result tables as a ranking, rather than by the range. This ranking is an inverse number, i.e., the higher the number the more sensitive, the signal. We also use a '-' implying the parameter distances were not monotonic, so the sensitivity is not applicable.

The other measure within each signal is monotonicity. Monotonicity is the property where as a parameter changes in value with respect to S_1, the distance between the corresponding signals change in a way that preserves the order of the changes. If we find a measure is not monotonic for a given signal it impacts how useful the distances are for predicting relative anomalous behavior to a normal value. A failure for a complexity measure to be monotonic means that there is a possibility than an outlier signal for a sensor appears closer to a nominal signal, than other signal behavior which is closer to nominal. The practical impact in our approach will be clusters and models that are not very informative in terms of separating anomalous behavior from nominal. In our result tables, we summarize the monotonicity result as: yes (Y) or no (N) with exceptions indicated if the trend was mostly monotonic, except for the rare cases when noise corrupts the results for a single parameter value.

As previously mentioned, the sensitivity values are not reported if the measure is not monotonic. Taken together, these measures of within signal variance can help identify measures that will be more suited to detect changes when a signal deviates from an expected

behavior, but not changing radically from one signal type to another. These measures then will be more specific in possibly identifying anomalous behavior in our problem domains.

V.3.3 Experiments with Real World Multivariate Time Series Data

The test data suite and experiments are used as a framework for gauging the distance measures using tightly controlled parameters for the signals. The controlled nature of the data and experiments limits our ability to explore how these measures may work in real-world domains. Specifically, the test data suite explores univariate signals, whereas our data in the problem domains is multivariate. TThe use of a real world dataset containing both complex, multivariate time series data, as well as known labels provides another experiment for exploring the best complexity measures in our test data suite.

V.3.3.1 Real World Data

The selected dataset[1] is a series of Electroencephalographies from a biomedical experiment on addiction. The Electroencephalographies come from 120 people broken into two groups:

1. A group diagnosed as alcoholics.

2. A control group made up of non-alcoholics.

Each participant was shown images and an Electroencephalographies measurement of their brain activity was recorded. A hundred and twenty such trials were conducted for each of the participants. Each Electroencephalography signal is made up a set of 64 channels, and each channel records a temporal brain signal, creating a multi-variate time series. These measurements are captured over 4 seconds and recorded at a sampling rate of 64Hz that produced 256 time samples per channel. In our data cube representation, each participant trial is the instance, and the cube has 64 features. Each feature for an instance is 256

[1]Found in the UCI data repository and donated by Henri Begleiter at the Neurodynamics Laboratory at the State University of New York Health Center at Brooklyn.

samples long. This cube is also annotated with information identifying the participant, the trial for the data collected, and a label indicating whether or not the participant is an alcoholic.

This data has previously been used for classification studies using methods such as Auto Regressive modeling [44], HMMs [178], ApEN [2], and even SVMs trained with Haar Wavelets and PCA [83]. These studies have produced good accuracy results using a variety of complexity measures found in our own study, combined with supervised classification algorithms.

We limited the size of original the data cube for our experiments. This was done by sampling without replacement 100 trials for each label. The sampling focused on the same image type, but was agnostic of the participants (hence a participant may show up more than once). This produced a cube that focused the data on a contextually similar series of the Electroencephalography experiments. This provides a consistency when comparing the complexity of the time series for each participant.

V.3.3.2 Approach to Exploring the Data

Using this data, and the selected distance measures, we build a distance matrix for each feature and produce the data cube made up of distance matrices. The next step is to build a distance matrix that represents the entire feature space, for exploration. Our modified Euclidean distance measure that normalizes each feature, is used to compute an overall distance between a pair of instance in the data.

The labels in this data present a chance to explore the effectiveness of the distance matrix to identify the two classes of participant. In order to do this, we run a series of N-Nearest Neighbor classifiers (N=1,3,5,7) to classify each instance in the data and study the classification accuracy.

Besides a classification study based on our measures, we utilize the fact that the distance measures we find are suited to the task of unsupervised methods. The second experiment with this data is to cluster the distance matrix and explore the smaller clusters for anomalies. Exploiting the knowledge of the labels, we explore the relationships between the participants.

V.4 Empirical Studies of the Models

The results from the test suite are presented below. We first examine the impact across different signals and measure effectiveness within signals to find the best compression-based method and compression algorithm. We then compare that choice against the results of ApEn and the Haar Wavelet using the same data.

V.4.1 Experiment 1: Selecting the Best Compression Algorithm and Complexity Measure

Using the test data suite, we look at the classifier results first to identify which distances and compression algorithms are the most robust to noise. We then look at the within signal measures to find the most sensitive and monotonic combination.

V.4.1.1 Classifier Results

The confusion matrices of the one-nearest neighbor classifier for data built using NCD are shown in Tables 9,10, and 11. The results show that the NCD measure combined with PPM and BWT produce the best results accuracy-wise for non-noisy and noisy signals. The confusion matrix for NCD with PPM and BWT shows that as the noise increases, the results for linear signals and the quadratic signals degrade. For LZW, the weaknesses are the inability to differentiate between sinusoids for attenuated sinusoids with a very small attenuation. The issues with the classifier show in the no noise scenario and are consistent across the noise range.

			Linear	Quadratic	Sinusoid	Atten. Sinusoid
Compression Algorithms	DZIP	Linear	96%	4%	0%	0%
		Quadratic	0%	96%	0%	4%
		Sinusoid	0%	0%	92%	8%
		Atten. Sinusoid	0%	0%	0%	100%
	LZW	Linear	96%	4%	0%	0%
		Quadratic	4%	96%	0%	0%
		Sinusoid	0%	0%	96%	4%
		Atten. Sinusoid	0%	0%	0%	100%
	PPM	Linear	64%	36%	0%	0%
		Quadratic	0%	96%	0%	4%
		Sinusoid	0%	0%	96%	4%
		Atten. Sinusoid	0%	0%	0%	100%
	BWT	Linear	100%	0%	0%	0%
		Quadratic	0%	96%	0%	4%
		Sinusoid	0%	0%	92%	8%
		Atten. Sinusoid	0%	0%	0%	100%

Table 9: NCD One Nearest Neighbor Classification Accuracy - No Noise

			Linear	Quadratic	Sinusoid	Atten. Sinusoid
Compression Algorithms	DZIP	Linear	89.2%	10.8%	0%	0%
		Quadratic	0%	96%	0%	4%
		Sinusoid	0%	0%	90.8%	9.2%
		Atten. Sinusoid	0%	0%	0%	100%
	LZW	Linear	86.8%	13.2%	0%	0%
		Quadratic	0%	96%	2.8%	1.2%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	0%	100%
	PPM	Linear	88.4%	11.6%	0%	0%
		Quadratic	0%	96%	0.4%	3.6%
		Sinusoid	0%	0%	90.4%	9.6%
		Atten. Sinusoid	0%	0%	0%	100%
	BWT	Linear	92%	8%	0%	0%
		Quadratic	0%	96%	0%	4%
		Sinusoid	0%	0%	96%	4%
		Atten. Sinusoid	0%	0%	0%	100%

Table 10: NCD One Nearest Neighbor Classification Accuracy - 2% Noise

			Linear	Quadratic	Sinusoid	Atten. Sinusoid
Compression Algorithms	DZIP	Linear	88.8%	11.2%	0%	0%
		Quadratic	0%	96%	0%	4%
		Sinusoid	0%	0%	90%	10%
		Atten. Sinusoid	0%	0%	0%	100%
	LZW	Linear	87.6%	12.4%	0%	0%
		Quadratic	0%	96%	2.4%	1.6%
		Sinusoid	0%	0%	78.8%	21.2%
		Atten. Sinusoid	0%	0%	0%	100%
	PPM	Linear	79.6%	20.4%	0%	0%
		Quadratic	0%	96%	2%	2%
		Sinusoid	0%	0%	94.4%	5.6%
		Atten. Sinusoid	0%	0%	0%	100%
	BWT	Linear	92%	8%	0%	0%
		Quadratic	0%	96%	2%	2%
		Sinusoid	0%	0%	95.6%	4.4%
		Atten. Sinusoid	0%	0%	0%	100%

Table 11: NCD One Nearest Neighbor Classification Accuracy - 10% Noise

The CiDM results shown in Tables 12,13,and 14 revealed a more robust handling of noise. This positive result is countered by a sharper loss in accuracy with all compression algorithms when comparing low attenuation sinusoids and the original signal.

This weakness for both CiDM and NCD with LZW can be explained by the fact that low attenuation values do not result in big differences from the non-attenuated sinusoids that have low frequency parameters. CiDM's use of Euclidean distance caused more confusion than NCD that is based on pure compression values. In general, we find that CiDM may be problematic for sensor values that show small decay in the measurements. In general, however, both metrics work quite well with all of the compression algorithms.

V.4.1.2 Monotonicity and Sensitivity Analysis

Tables 15 and 16 list the monotonicity and sensitivity results for the four signals types as non-noisy signals. Each entry in the table corresponds to a signal type run with a compression and distance measure pair. The first value for each entry deals with changes in

			Linear	Quadratic	Sinusoid	Atten. Sinusoid
Compression Algorithms	DZIP	Linear	100%	0%	0%	0%
		Quadratic	0%	100%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	LZW	Linear	96%	0%	0%	4%
		Quadratic	0%	100%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	PPM	Linear	100%	0%	0%	0%
		Quadratic	0%	100%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	BWT	Linear	100%	0%	0%	0%
		Quadratic	0%	100%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%

Table 12: CiDM One Nearest Neighbor Classification Accuracy - No Noise

			Linear	Quadratic	Sinusoid	Atten. Sinusoid
Compression Algorithms	DZIP	Linear	96%	0%	4%	0%
		Quadratic	4%	96%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	LZW	Linear	96%	0%	0%	4%
		Quadratic	3.2%	96.8%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	PPM	Linear	100%	0%	0%	0%
		Quadratic	4%	96%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	BWT	Linear	100%	0%	0%	0%
		Quadratic	4%	96%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%

Table 13: CiDM One Nearest Neighbor Classification Accuracy - 2% Noise

			Linear	Quadratic	Sinusoid	Atten. Sinusoid
Compression Algorithms	DZIP	Linear	96%	0%	4%	0%
		Quadratic	4%	96%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	LZW	Linear	96%	0%	0%	4%
		Quadratic	3.2%	96%	0%	0.8%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	PPM	Linear	100%	0%	0%	0%
		Quadratic	3.6%	96.4%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%
	BWT	Linear	96%	0%	4%	0%
		Quadratic	4%	96%	0%	0%
		Sinusoid	0%	0%	80%	20%
		Atten. Sinusoid	0%	0%	20%	80%

Table 14: CiDM One Nearest Neighbor Classification Accuracy - 10% Noise

the parameters for slope (linear) , the x^2 coefficient (quadratic), frequency (sinusoids), and level of attenuation (attenuated sinusoids). The second value stands for the y-intercept (linear), the coefficient of the first order term, x (quadratic), phase change (sinusoid) and frequency (attenuated sinusoid).

Overall, there is not much change in the results from noiseless signals to 2% noise, but there is significant deterioration in the quality of the results (monotonicity and sensitivity) when the noise levels reach 10%. On closer observations, CiDM has better monotonic and sensitivity properties across signal types and parameter value changes; The exception is for sinusoidal signals for all compression measures, indicating that none of the compression measures are effective for periodic signals. The NCD measure has poor monotonicity properties across the board, but has its best results for LZW compression. For CiDM, LZW, BWT, PPM and DZIP show similar results.

		Signal Template	Distance Measure	
			NCD	
			Monotonicity*	Sensitivity*
Compression Algorithms	DZIP	Linear	N, N	-, -
		Quadratic	Y, N	1, -
		Sinusoid	Y, N	3, -
		Atten. Sinusoid	N, Y	-, 1
	LZW	Linear	Y, N	1, -
		Quadratic	Y, Y	2, 1
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 3
	PPM	Linear	N, N	-, -
		Quadratic	N, N	-, -
		Sinusoid	Y, N	2, -
		Atten. Sinusoid	N, N	-, -
	BWT	Linear	N, N	-, -
		Quadratic	N, N	-, -
		Sinusoid	Y, N	1, -
		Atten. Sinusoid	N, Y	-, 2

Table 15: Monotonicity and Sensitivity with No Noise

		Signal Template	Distance Measure	
			CiDM	
			Monotonicity*	Sensitivity*
Compression Algorithms	DZIP	Linear	Y, Y	2, 2
		Quadratic	Y, Y	2, 2
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 1
	LZW	Linear	Y, Y	2, 2
		Quadratic	Y, Y	2, 2
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 4
	PPM	Linear	Y, Y	2, 2
		Quadratic	Y, Y	2, 2
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 2
	BWT	Linear	Y, Y	2, 2
		Quadratic	Y, Y	2, 2
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 3

Table 16: Monotonicity and Sensitivity with No Noise

Tables 17 and 18 show these same analyses for 2% noise. The major difference compared with Tables 15 and 16 is that NCD for every compressor save BWT is less monotonic. For CiDM, monotonicity also suffers with the high values of the parameters. In particular, LZW shows the worst results. This may be attributed to the nature of the LZW algorithm which uses 8 bit integers in the compression algorithm, which is insufficient for large signal values. Even with the robust nature of LZW above, this limitation disqualifies LZW from being a possible compressor choice in the anomaly detection phase of this work. In this case we find CiDM as the superior measure with PPM and BWT the better compressors.

		Signal Template	**Distance Measure**	
			NCD	
			Monotonicity*	Sensitivity*
Compression Algorithms	DZIP	Linear	N, N	-, -
		Quadratic	Y, N	1, -
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, N	-, -
	LZW	Linear	Y (except 1), N	2, -
		Quadratic	N, N	-, -
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 2
	PPM	Linear	Y (except 1), N	1, -
		Quadratic	Y, N	3, -
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 1
	BWT	Linear	Y, N	1, -
		Quadratic	Y, N	2, -
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 1

Table 17: Monotonicity and Sensitivity with 2% Noise

Lastly, Tables 19 and 20 represent the analysis with the maximum amount of noise introduced into the signals at 10%. Here, only BWT is robust in place for NCD. In CiDM, both PPM and BWT remain robust with the monotonicity of the linear signal for y-intercept

		Signal Template	Distance Measure	
			CiDM	
			Monotonicity*	Sensitivity*
Compression Algorithms	DZIP	Linear	Y, Y (Except 500)	1, 1
		Quadratic	Y, Y (Except 500)	1, 1
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 3
	LZW	Linear	Y, Y(Except 500)	2, 2
		Quadratic	Y(Except 500), Y(Except 500)	1, 1
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 4
	PPM	Linear	Y, Y(Except 500)	1, 1
		Quadratic	Y, Y(Except 500)	1, 1
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 2
	BWT	Linear	Y, Y(Except 500)	1, 1
		Quadratic	Y, Y(Except 500)	1, 1
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 2

Table 18: Monotonicity and Sensitivity with 2% Noise

monotonicity disappearing , otherwise both compressors remain similar to the 2% noise analysis.

These results point to a choice between PPM and BWT with CiDM as the best choices to implement the KC approximation. Between PPM and BWT, we picked the PPM compressor because it more efficient to compute on the data. Thus we choose the CiDM/PPM combination as the best choice from these combinations.

V.4.2 Experiment 2: Comparison with Approximate Entropy and Wavelets

V.4.2.1 Classifier Results

As a next step, we chose the best combination for the KC measure (CiDM/PPM) and compared them again the ApEn and the Haar Wavelet based distance measures using the same comparison criteria as the last experiment. The confusion matrix for ApEn with no noise in Table 21 shows one of the higher accuracies in the presence of no noise with a

		Signal Template	Distance Measure	
			NCD	
			Monotonicity*	Sensitivity*
Compression Algorithms	DZIP	Linear	N, N	-, -
		Quadratic	Y(Except 500), N	1, -
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, N	-, -
	LZW	Linear	P, N	3, -
		Quadratic	N, N	-, -
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 2
	PPM	Linear	Y(Except 1), N	2, -
		Quadratic	Y(Except 10), N	2-, -
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 1
	BWT	Linear	Y, N	1, -
		Quadratic	Y, N	2, -
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 1

Table 19: Monotonicity and Sensitivity with 10% **Noise**

		Signal Template	Distance Measure	
			CiDM	
		Monotonicity*	Sensitivity*	
Compression Algorithms	DZIP	Linear	Y, N	1, -
		Quadratic	Y, Y(Except 500)	1, 1
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-0.1
	LZW	Linear	Y, N	2, -
		Quadratic	Y, Y(Except 500)	1, 2
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 2
	PPM	Linear	Y, N	1, -
		Quadratic	Y, Y(Except 500)	1, 1
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 1
	BWT	Linear	Y, N	1, -
		Quadratic	Y, Y(Except 500)	1, 1
		Sinusoid	N, N	-, -
		Atten. Sinusoid	N, Y	-, 1

Table 20: Monotonicity and Sensitivity with 10% **Noise**

98% accuracy and is clearly able to distinguish between the information in the different signals. The confusion matrix shows that the few errors for noiseless signals with ApEn occur in differentiating the quadratic signals from both types of sinusoidal signals. This overall excellent result is undermined by the results in the presence of noise, such as those shown for 2% noise in Table 22 where the ability discriminate between the linear and quadratic functions breaks down. This result grows worse with 10% noise having less than 50% accuracy overall. This result is simply much worse than the best choices from the Complexity-based measures.

	Linear	Quadratic	Sinusoid	Atten. Sinusoid
Linear	100%	0%	0%	0%
Quadratic	0%	92%	4%	4%
Sinusoid	0%	0%	100%	0%
Atten. Sinusoid	0%	0%	0%	100%

Table 21: Classification Accuracy: ApEn Wavelet One Nearest Neighbor Results - No Noise

	Linear	Quadratic	Sinusoid	Atten. Sinusoid
Linear	44%	66%	0%	0%
Quadratic	48%	48%	4%	0%
Sinusoid	0%	8%	92%	0%
Atten. Sinusoid	0%	0%	12%	88%

Table 22: Classification Accuracy: ApEN Wavelet One Nearest Neighbor Results - 2% Noise

The Haar wavelet based distance is a more robust measure and produces accuracy results similar to the CiDM measures. The confusion matrix in Table 23 shows the Haar doing particularly well with an overall 97% accuracy. The measure has problems when differentiating the attenuation coefficient sinusoid from similar sinusoid based signals. Closer

examination of the 8% misclassification of the attenuated sinusoid as the standard sinusoid shows the problem with classification as an issue with the Haar wavelet being unable to handle the lowest attenuation coefficient and confusing the signal with the sinusoid having the same frequency. Any noise increase with the Haar wavelet distance produced the same confusion matrix and accuracy. This showed the most robustness to noise of any of the measures.

	Linear	Quadratic	Sinusoid	Atten. Sinusoid
Linear	96%	0%	0%	0%
Quadratic	0%	100%	0%	0%
Sinusoid	0%	0%	100%	0%
Atten. Sinusoid	0%	0%	8%	92%

Table 23: Classification Accuracy: Haar Wavelet One Nearest Neighbor Results

From these experiments, ApEn would be the best choice for noise free situations. CiDM and wavelets compare favorably, with similar problems in terms of classification errors. The wavelet transformation shows the best robustness property for noisy signals.

V.4.2.2 Monotonicity Analysis

	No Noise	2% Noise	10% Noise
Linear	I,I	N,N	N,N
Quadratic	N,Y	N,N	N,N
Sinusoid	N,N	N,N	N,N
Atten. Sinusoid	N,Y	N,N	N,N

Table 24: ApEn Monotonicity Results

The results for monotonicity for APEn in Table 24 and for the wavelets in Table 25 are contrasted with the choice of CiDM with PPM and provide insight into the classification

	No Noise	2% Noise	10% Noise
Linear	Y,Y	Y,N	Y,N
Quadratic	Y,Y	Y,N	Y,N
Sinusoid	N,N	N,N	N,N
Atten. Sinusoid	N,Y	N,Y	N,Y

Table 25: Haar Wavelet Monotonicity Results

results. For example, in ApEn, the linear signals are invariant ('I') and zero. However, only the 'x' coefficient of the quadratic signal and the attenuation coefficient are monotonic for the signals with no noise. Once noise is introduced, the linear signals no longer satisfies the monotonicity property, and the rest of the parameters are also non-monotonic. This would make sense, since the ApEn's classification accuracy was poor, for even a moderate amount of noise.

Investigating the monotonicity of the wavelets produces results similar to the CiD-M/PPM combination. The dissimilarities are under the presence of noise. Comparing the wavelet results in Table 25 where the secondary coefficients for the linear case (the slope) and the quadratic (first order term) both lose monotonicity with those in Tables 19 and 17 , where CiDM/PPM maintains monotonicity for all cases of parameters but the largest. While CiDM/PPM appears to be the choice for identifying differences in a signal type, the Haar wavelets produce comparable results.

V.5 Multivariate Time Series Experiments on Real Data

The experiments on the test data suite show that the wavelet transform and the CiD-M/PPM combination are the two most accurate measures that are both robust to noise while remaining sensitive to detecting changes in signals of the same general type. We utilize both measures with the selected real world data of the EEGs to look at both their classification abilities as well as how they cluster the data.

Both measures using the 1,3,5, and 7 nearest-neighbor classification produce contrasting results as shown in Tables 26 and 27. The CiDM/PPM combination resulted in an highest overall accuracy of 60.8% on the 3 nearest neighbor classifier. The breakdown of this classification was 73.73% accuracy on the alcoholic group, and a low 48.48% accuracy on the control. Using the Haar Wavelet distance, the accuracy was highest with 1-NN classifier at 69.84%, with the breakdown of 71.71% accuracy on the alcoholic group, and 68.68% accuracy with control. Classification of with the wavelets was slightly less accurate on the alcoholic group, but was much better at identifying the control group.

N-Nearest Neighbor	Overall Accuracy	Alcoholic Accuracy	Control Accuracy
1	59.79%	66.66%	53.53%
3	60.80%	73.73%	48.48%
5	57.28%	74.74%	40.40%
7	59.29%	76.76%	42.42%

Table 26: Nearest Neighbor Classification Accuracy of EEG data with CiDM/PPM

N-Nearest Neighbor	Overall Accuracy	Alcoholic Accuracy	Control Accuracy
1	69.84%	71.71%	68.68%
3	63.81%	70.70%	57.57%
5	63.31%	73.73%	53.53%
7	63.31%	74.74%	52.52%

Table 27: Nearest Neighbor Classification Accuracy of EEG data with Haar Wavelet Transform

These results while less accurate than the methods designed specifically for classification, show similarities to some of the conclusions drawn from previous work [44] with the dataset. Specifically, the classification methods of other researchers and our use of distances measures show the alcoholic group as easier to classify than the control. The control group may have a larger variance of response to the stimuli whereas the alcoholic

group is more cohesive in signal shape. Also of note, in the use of supervised learning with the EEGs, among the best techniques was the Haar wavelet with PCA and a Support Vector Machine classifier [83]. This coincides with the improvement in our experiments with grouping the control samples using the wavelets.

We also clustered the data using our distance measures to look for properties of the data. Since the wavelet transform performed the best in the classification study, we used the wavelet distance matrix with a complete-link agglomerative clustering algorithm to produce a hierarchy. The resulting dendrogram was used to analyze the clusters and examine the distribution of the labels. The overall dendrogram is shown in Figure 19. The participants are singletons at the bottom, and the horizontal line across the dendrogram at distance 11.3, presents a reasonable separation with 7 different clusters. We label them clusters 1 through 7 from left to right.

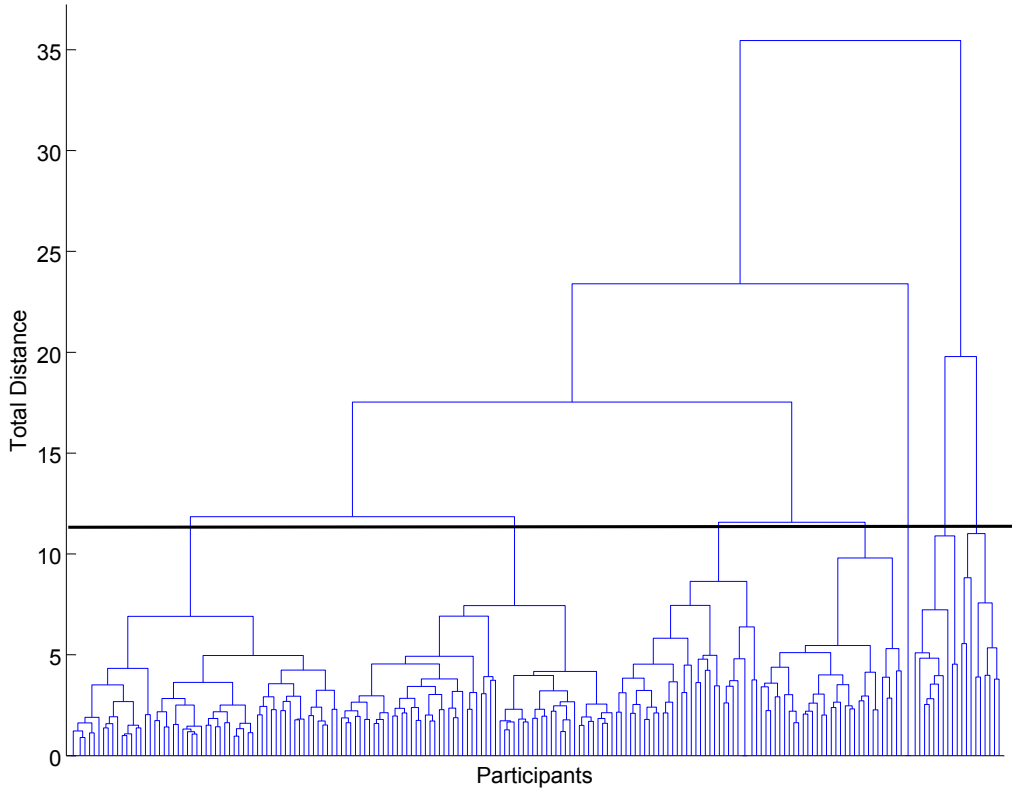

Figure 19: Dendrogram of EEG Data Using Haar Wavelet transform

We examined this grouping and analyzed it for accuracy of labeling by group. We compare the actual labels to the distribution in Table 28. We find that there are a couple of large clusters which are fairly balanced. Since our best classifier with the wavelet based distance matrix was a 1-Nearest Neighbor classifier, it makes sense that the structure for the labels may be in the pairs at the bottom of the dendrogram.

Cluster Number	# of Alcoholics	# of Alcoholics
1	39	19
2	25	34
3	14	17
4	13	18
5	1	1
6	4	6
7	4	5

Table 28: Distribution of Labels in Clusters

V.6 Conclusions

We examined a number of dissimilarity measures designed to reduce the time series dimensionality of the data produced from complex systems. These distance measures cover compression based methods, information theory, and signals analysis. The goal is to identify measures that can reduce this dimensionality and retain as much information about the complexity of the signal for anomaly detection. We compared two compression based measures, NCD and CiDM, and for each, we examined their effectiveness over a set of compression algorithms. These choices were compared to Approximate Entropy, and the Haar Wavelet Transform.

Utilizing distance measures for use in anomaly detection requires understanding how precise the different measures are at differentiating between similar signal types. This required a test data suite made up of several distinct signal types, each with a varying set of parameters. The distances were examined for their ability to differentiate between

the different signal types, and their changes within each signal type as the parameters are varied. Within signal differentiation was measured using monotonicity and sensitivity as performance measures. Lastly, the best distance measures in this test suite were examined on a real-world multivariate time series dataset, a set of labeled Electroencephalographies. The chosen distance measures were examined for how they differentiate the known labels, and the clusters they create.

The results from the test bench showed that the use of CiDM with the PPM compression was the most effective of the compression based methods. CiDM was particularly robust when the signals in the test data were combined with Gaussian noise. When CiDM and PPM were compared to ApEn and the wavelets, the wavelets was more robust to noise in differentiating the different signals. The wavelet, however, lacked the ability to be as precise with varying parameters of an individual signal type as CiDM/PPM. Both the CiDM and wavelet measures were compared on the Electroencephalography dataset, where the wavelets were more accurate overall, and particularly for the control group. Previous results from other researchers on this data confirmed that the control group was harder to classify. These results indicate that the wavelets are clearly superior in terms of classification, while the test bench results for monotonicity indicate the CiDM and PPM would be preferable to identifying distances when the signals are similar, a distinction that makes it still useful for an anomaly detection task. When we clustered the Electroencephalography data using the distance matrix based on the wavelet transform, we found that the data broke into 7 clusters. Looking at these clusters show a balance for the labels, indicating that similarities for the labels exist more in the pairwise distances than as clusters of behaviors.

Previous studies on distance measures incorporate a large number of measures and explore their accuracy on supervised learning tasks with complex and often univariate signals. This research is meant to take a more focused view for the purpose of our research problem. This meant comparing the distances measures for their ability to help reduce the dimensionality of data and be effective for anomaly detection. This contribution is a series

of results that explore which measures may be more suitable to use for building anomaly detection model for complex systems that produce time series data where the majority of signals can be modeled in terms of dynamic functions. The results from these experiments carry into our research approach of using these measures to reduce our data for exploratory methods of anomaly detection in real world domains.

CHAPTER VI

ANOMALY DETECTION OF UNLABELED AIRCRAFT DATA FOR AVIATION SAFETY

After identifying complexity based distance measures appropriate for anomaly detection in Chapter V, we return to the primary goal of discovering anomalies in operational data collected from complex systems. In contrast to the approach in Chapter IV that exploits multiple sources of expert information, we have built an approach which is less reliant on expert-based knowledge and data.This anomaly detection approach is based on the complexity measures as a dimensionality reduction technique. In this chapter, we apply this approach to the domain of aircraft systems for improving aviation safety. The goal in this chapter is to isolate anomalies that are interesting safety related events. These anomalies would form the basis for building anomaly detection models for the takeoff of aircraft.

We discover these flight anomalies through modeling based on unsupervised learning techniques that are applied to search a large database of flight operations data. We make an assumption that most of the flight instances in this database will fall into a nominal range, but a small subset of flight will differentiate themselves from this nominal range. With a very large amount of data, we call this search for anomalies a "needles in a haystack" problem.

In our approach we begin by transforming our curated flight data from Chapter IV into a data cube based on the format specified in Section III.3.1. We consider the flights to be the instances, the sensors to be the features, and each sensor's measurements over time during the flight to be the signals. The data cube is contextualized according to the takeoff phase of flight in order to constrain the data and focus on specific types of anomalies. Before we can apply clustering algorithms to this data, we must first reduce the dimensionality of the data cube into a two dimensional data set of dissimilarities between each pair of flights

in our data. This is accomplished by applying one of the complexity measures explored in Chapter V. The use of complexity measures provides a method of reducing temporal signals of each instance and retaining important characteristics about the signal. After dimensionality reduction, we then apply hierarchical clustering to the reduced data. From these structures, we locate the larger clusters and label them as nominal. We examine each instance within the smaller clusters to characterize these instances and identify anomalies that are relevant to the expert. The characterization of an instance involves identifying "significant actors." We define a significant actor as a feature in the data cube that best differentiates the examined instance from the selected nominal set. When a cluster contains many more instances than can be examined manually, we produce a group characterization method using interactive techniques to sub-group the cluster by relevant features. These significant actors are ranked by significance and we present them for each instance to the expert through visualization of the anomalous signal against a nominal sample. The expert may then identify the likely cause of the flight anomaly and either flag it as interesting or not.

This chapter is organized as follows. Section VI.1 reviews previous approaches and limitations of previous work for anomaly detection in large amounts of unlabeled, high dimensional aircraft flight data. In Section VI.2, we formally describe our approach, first explaining the curation and transformation of the multivariate time series data into dissimilarity measures for exploration of anomalous instances using cluster analysis methods. Following clustering, we develop a scheme for identifying possible anomalous flights. We extract information from these anomalous flights to guide experts in characterizing them into equipment, environment, or human-related categories of anomalies. We conduct further study of these anomalies to determine if they should be flagged as aviation safety issues. Section VI.3 illustrates the approach applied to our transformed data by contrasting it with the previous work. Based on the results of our approach, we visualize the results and examine case studies about aviation safety based on our findings. Lastly, in Section VI.4,

we summarize and discuss our application of this approach to the aircraft flight system domain.

VI.1 Previous Work on Anomaly Detection of Aviation Data

Research into building systems for anomaly detection of aviation data has progressed from general methods designed for any domain, to approaches tailored for anomaly detection in aircraft flight system domains. Such methodologies are designed to handle large amount of raw flight data. We review some of the original work in this area, such as the state of the art methods of Principal Component Analysis with Density Based Clustering (PCA-DBSCAN) [94, 98], and Multiple Kernel Anomaly Detection (MKAD) [36, 37]. Both frameworks have been shown to be effective in discovering a variety of anomalies in aviation data. The assumptions these systems operate under reduce their generality in terms of detection in the problem domain, and our approach is designed to address these problems.

A theme that these techniques have in common is that starting with a data cube representation of the data, dimensionality reduction are applied to reduce the dimensions of the feature space, or to simplify the task of comparing pairs of flights. The choice of the approach and how it is applied to the raw data defines the nature of the overall methodology.

VI.1.1 From A General Approach to Approaches Using Limited Data

General approaches to exploring a feature space for identifying anomalous instances have employed a number of different learning algorithms, requiring highly tunable global models and error minimization procedures. Such methods include least-squares regression [12] to derive discriminative models from data. This leads to the development of robust algorithms to detect a number of additive faults through the use of receiver operating characteristic curves plotted to tune the detection algorithm and set the false alarm

rates [30]. Such approaches require large amounts of real and simulated data to derive general and robust solutions. Further, these methods are supervised approaches, since practitioners must understand the data and the results of experiments on the model(accuracy and false positives) in order to tune the system.

The domain of aviation flight data has produced a number of techniques for discovering anomalies, such as SequenceMiner [17], Orca [7], The Inductive Monitoring System [70], and Morning Report [28]. These methods are built with varying amounts of data, and are computationally expensive. For example, Morning Report, which was built to be run overnight on the previous day's flight data to generate a report to be examined in the morning.

SequenceMiner focuses on clustering methods for exploring a set of instances by reducing the features signals using a metric for measuring common sequences known as the normalized longest common subsequence [18]. This method's use of the metric across features, retains the original feature semantics, allowing an anomaly to be characterized from this model. This metric is similar to our complexity measure, but is targeted for analysis of symbolic sequences. Often to utilize a metric like the normalized common subsequence, a practitioner will preprocess numerical signals into a symbolic form. Transformation of numeric signals to symbolic sequences requires the use of a technique such as Symbolic Aggregate Approximation [97]. These techniques require signals of an adequate length (such as a few minutes) to produce symbolic sequences that are long enough to effectively leverage the similarity metrics. Since the focus of anomaly detection may be on a period of time such as takeoff or landing that may be as short as 20 seconds, the signals for those phases could be difficult to transform into symbolic sequences.

Orca uses a scalable k-nearest neighbor approach to detect anomalies in data with continuous and discrete features. Since each data point is a sample in time and treated as independent by the algorithm, Orca struggles to detect anomalies with temporal signatures.

The Inductive Monitoring System is a distance based anomaly detection method that

focuses on continuous parameters. The method uses incremental cluster analysis to build models of expected operation of the system, but also does not consider the temporal patterns in the data. The Euclidean distance from an anomalous data point to the nearest cluster center is reported as the anomaly score for that data point. This method was originally designed to deal with flight data, where new monitors in a diagnostic system could be built from the parameters of the clusters.

Morning Report builds a statistical signature across each feature of a sample to reduce it to a smaller dimension. This is then used with distance metrics such as Mahalanobis distance to find flights that are outliers from the majority of the data points. The use of a statistical signal makes characterization of the found anomalies difficult without the use of another method to investigate the original data of the anomalous instance.

SequenceMiner and Morning Report are designed to interact with temporal signals in the data. These methods make assumptions, such as SequenceMiner requiring a symbolic transformation and Morning Report requires a pass from another algorithm through the original data to help an expert characterize found anomalies.

VI.1.2 Principle Component Analysis and Density Based Clustering

The first method we look at in detail is one that combines Principal Component Analysis (PCA) and Density based clustering [94, 98]. The traditional method to PCA analysis is to generate the eigenvalues and eigenvectors of a covariance matrix, and retain the eigenvectors that correspond to the highest eigenvalues, such that at least 90% or higher of the variance is retained in the chosen features. This method relies on "unrolling" each instance's features in the data cube into a new set of features where each is a sample of time for each sensor. This transformation to make each sample independent is described in Section V.1. These "unrolled" instances are projected into a lower dimensional space that corresponds to the selected eigenvectors. This reduces the feature size and because of the eigenvectors, creates only orthogonal features which are uncorrelated. The next step is to

apply density-based clustering to this reduced feature space, which provides a number of advantages. It requires little domain knowledge to determine the input parameter of how many points create a cluster, and the threshold for similarity. The algorithm is relatively efficient for large numbers of instances, but not for large feature spaces. The clusters generated by the algorithm are of arbitrary shape, and the algorithm is robust to noise in the data. An advantage of the output of DBSCAN is that it does not require a sample to be affiliated with a cluster, so values that are sufficiently different will be labeled as outliers in the dataset. The output of this method produces clusters which are considered to be homogeneous in the chosen feature space, and a set of outlier data points that become the focus for further investigation.

There are two primary issues with this method. The first is that the "unrolling" of the instances in the data cube results in the removal of potentially important temporal information. This method isolates each sample in each signal as a unique feature, when the change over that time may be an important factor. The other issue with this approach is that the application of PCA which results in a transformed feature space does not allow for easy analysis and characterization of the abnormal nature of the outliers in terms of original features.

VI.1.3 Multiple Kernel Anomaly Detection

In contrast to the unsupervised approach of PCA-DBSCAN, Multiple Kernel Anomaly Detection (MKAD) [36, 37] represents a semi-supervised approach that operates on large data cubes such as those found in the aircraft flight system domain. Similarly to SequenceMiner, the algorithm first preprocesses all continuous sequential data into symbolic feature sequences. The preprocessing is necessary for measuring the similarity of these sequences between samples. The pairwise comparisons are organized for learning by building two kernels that combine the feature streams for either continuous or discrete values.

The kernel method for both types is based on the normalized Longest Common Subsequence [18], the same metric used in SequenceMiner for measuring common sequences. The kernel is built for a one-class SVM classifier [136]. This procedure is semi-supervised, so the data used for training should ideally contain only nominal samples. This method of isolating anomalies attempts to exploit common sequential information for two samples represented as a single value. When built over the entire set of samples, this technique can construct the model of nominal behavior. Analysis of flagged anomalies is examined post-SVM, since the SVM model based on kernel methods is difficult to interpret. MKAD is demonstrated with a combination of switching and continuous FOQA data for a fleet of aircraft [36]. The derived models find a number of interesting anomalies, such as a high energy approach landing, pilot responses to environmental disturbances, and high speed low altitude flights. MKAD uses a SequenceMiner routine on the group of anomalies flagged in the test set to better understand why these samples were detected.

MKAD is a robust algorithm for anomaly detection, but there exist issues that interfere with direct application to operational data. Due to the semi-supervised nature of the algorithm, MKAD requires knowledge of a nominal set for training. This keeps it from being applicable to large unlabeled data, until another method has been utilized to remove anomalies and isolate a set of instances as nominal. The implementation of MKAD attempts to deal with this issue but still struggles. The similarity to SequenceMiner in the use of symbolic sequences means that MKAD may also have trouble working with sequences in the data that are not very long. In the demonstrations of MKAD, the sequences used included significant amounts of the flight operations during landing to construct the training and testing sets. In our data, we may be focusing on a more precise phase that is of a short durtion, which would create difficulties for MKAD. Lastly, MKAD requires the need for a secondary technique such as SequenceMiner to identify the features that are anomalous for characterization. These issues interfere with this algorithm using a large collection of unlabeled data such as the one we have collected.

VI.2 Approach to Exploration and Characterization

Our approach is designed to address a number of the issues with both PCA-DBSCAN and MKAD, namely the temporal dependency that is ignored in PCA-DBSCAN, the lack of knowledge about nominal instances in the data for MKAD, and the ability to use the same data and clusters to identify and characterize anomalies in the data. Our approach to this exploratory anomaly detection for complex systems is broken into a series of steps. First, we transform and contextualize the curated data to produce an initial data cube for exploration. We then apply dimensionality reduction to the cube to produce a transformed two dimensional dissimilarity matrix, which is used with a hierarchical clustering approach to generate clusters from the data. We assume the results of the clustering produce some large cluster where significant number of the flight instances reside. These are labeled the nominal data. The rest of the instance can be divided into groups: (i) a very small number of outlier data points and (ii) some small clusters that distinct from the large nominal clusters. As a first step we study the outliers by selecting features that sufficiently distinguish the outliers from the nominal data. Then further analysis using group characterization is performed on the small clusters.

We illustrate the stages of this approach in Figures 20 and 21. These two figures break up the work into two general procedures. Figure 20 represents the transformation to clustering stages of the data. The input to begin this process is a curated data set, which is then contextualized and transformed into the standard data cube. We reduce this data cube by applying our complexity measure as a mechanism for dimensionality reduction. This produces a cube of distances matrices. We collapse the cube of distance matrices using the weighted euclidean distance and cluster the resulting dataset to identify clusters of anomalies and clusters of nominal ranges. The output at the end of the clusters should be preliminary groups of instances for further consideration.

These partitioned instances labeled as nominal or anomalous are the input for the procedures shown in Figure 21. These steps are responsible for the characterization and modeling

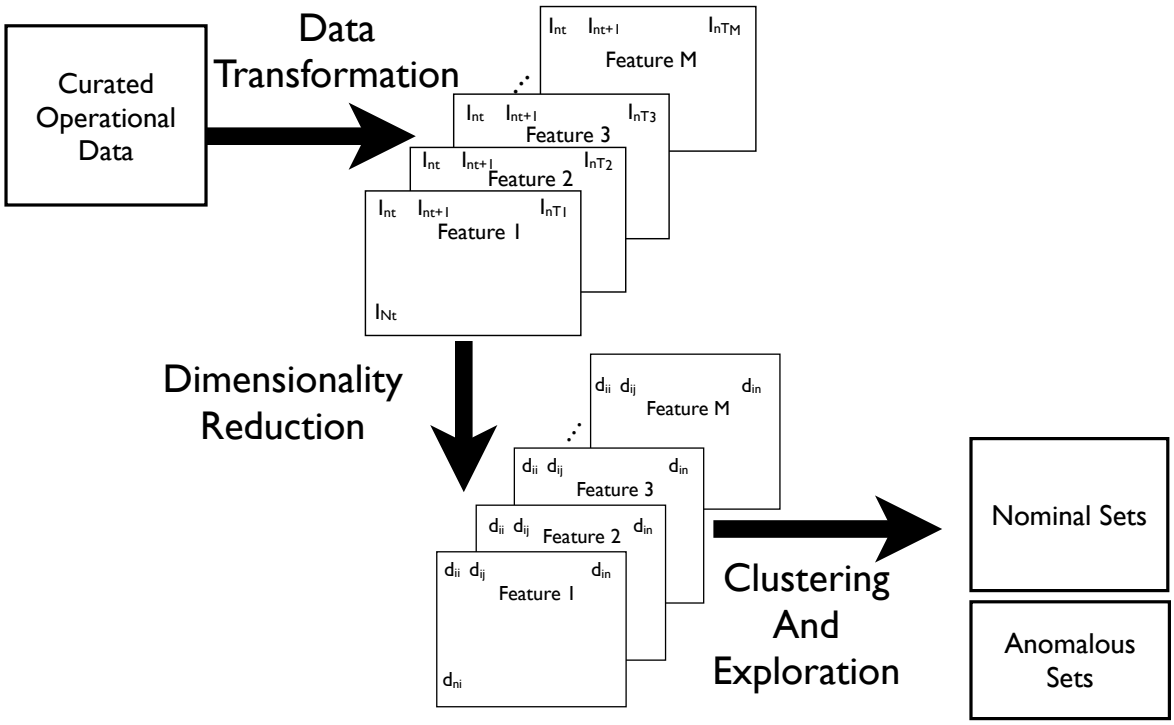

Figure 20: Transformation to Clustering of Unlabeled Data

of the data. With possible anomalous groups and a nominal base of clusters, feature selection is applied to identify the relevant features that differentiates each anomalous group. An expert will use these features to further characterize the anomalies for their level of failure. Coupled with the nominal sets, these groups can be used to produce new models of anomaly detection, to identify new anomalies in the incoming data.

VI.2.1 Transformation and Reduction

As a first steps we start with the aircraft data from the curated database described in Chapter IV and extract and transform the data from this database and produce a data cube representation. This step also narrows the focus of the anomaly detection. We then apply the dimensionality techniques discussed in Chapter V to reduce this to the dataset to a form that allows for the application of a standard clustering algorithm.

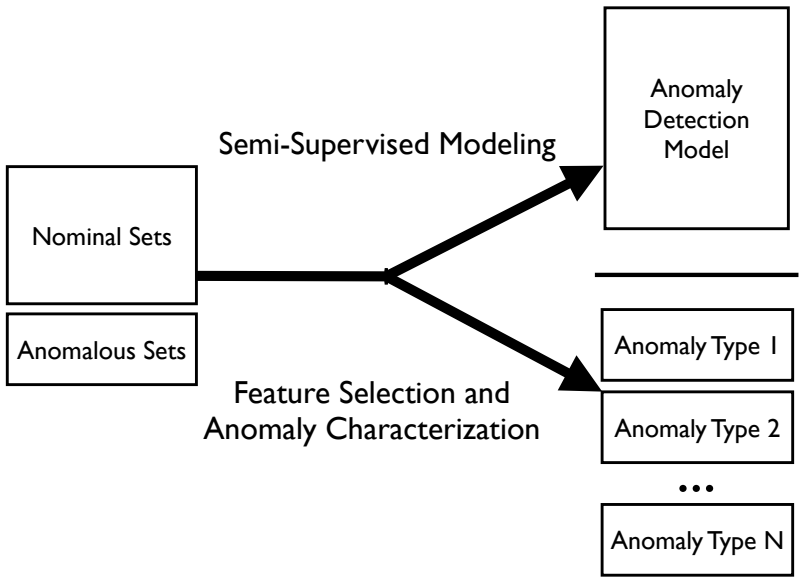

Figure 21: Characterization and Modeling of Anomalies

VI.2.1.1 Extraction and Transformation

The input for this approach is raw collected data organized and pre-processed to remove sensors, which are not germane to the operation of the aircraft. This removal helps improve the efficiency of the approach, and removes noise. This involves eliminating features not involved with describing the operation of the aircraft during flight such as meta-data about the internal storage on the aircraft. This removal based on expert input removes the number of possible features in our data from 145 sensors to 87. This dataset is further curated with location of each instance. The location is kept as a meta-data that can be used later to classify anomalies that can be attributed to environmental conditions. The location is responsible for variance in operations associated with different altitudes and geographical elements in the data, such as typical weather conditions and length of the runway. The location is identified by mining the latitude and longitude positions at takeoff to find common clusters of positions that coincide with general locations of airports.

After applying all of the curation steps, we are ready to build the relevant dataset. From the curation we select from 5333 possible flights to include in the data cube. This complete set covers flights of 12 different aircraft over a period of 5 years. This provides a broad

enough selection to encompass a variety of flight situations that include takeoff locations, and weather, reducing the need to perform the analysis only for restrictive contexts. We then extract data for a chosen phase of operation for the aircraft. We focus on a specific phase of flight in order to contextualize the instances for the same period of time during flight.

For this study, we further contextualize the data to takeoff, a situation when the aircraft equipment and pilots are most stressed. The calculation for this phase can be computed in a number of ways. In this work, we examine the two possible ways illustrated in Figure 22. The first method, known as the "phase computer method", relies on the computer of the aircraft to detect a takeoff situation based on pre-specified conditions. This generally begins when the pilot applies significant thrust to the engines, and then assuming a threshold of 90 seconds implying the aircraft will have completed its takeoff phase and begun the ascent phase within this time. The data we extracted by this method is down sampled, producing feature signals that have the same temporal length and are synchronized on the same points in time. This method was the first developed as a baseline for validating a variety of anomaly detection methods such as Principal Component Analysis with Density Based Clustering and specifically the Multiple Kernel Anomaly Detection. The implementation of Multiple Kernel Anomaly Detection makes demands on the structure of the data and that meant we needed as much information as possible, while retaining identical signal lengths for each feature. This method wasn't intended to be the final choice for transformation, but is included because a significant amount of work was done with this transformation, including the initial application of this approach. Also of note, due to issues discovered in the implementation of MKAD, this set contained 2116 flight instances extracted for a single aircraft.

The second method of calculating takeoff is to use a physical cue measured by the sensors. This is known as the "weight on wheels method." The data from this method is based on when the sensor recording the weight on the landing gear registers 0, meaning that the

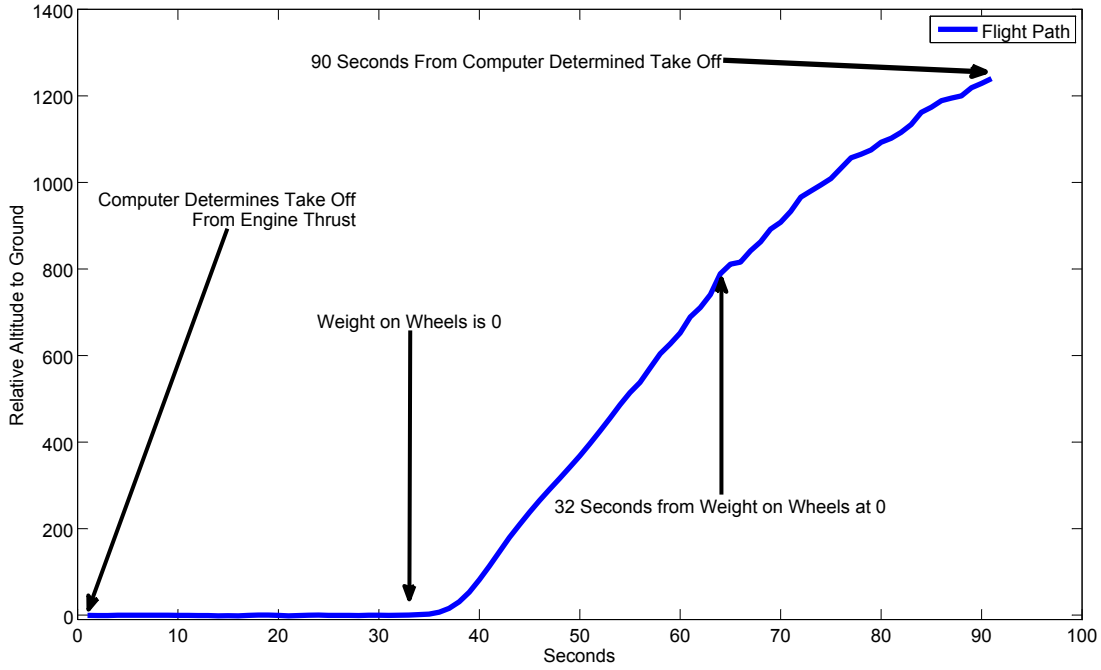

Figure 22: Illustration of Different Methods of Capturing Takeoff

plane has lifted off the ground. Normally, an expert would cut this off when the pilot retracted the landing gear, an indication that the plane is moving into full ascent mode since the aircraft's velocity and direction may increase the vertical and horizontal force and damage the landing gear. The retraction of landing gear is a variable timed procedure and in the data, this can take on average 30 seconds, and is usually between 25 seconds to 35 seconds. Due to the nature of the complexity based measures and to remove a possible variable in the time it takes to retract the gear, we take a 32 second sample after the weight on wheels has changed to zero. Unlike the first method, the features were not down-sampled, allowing the sensors original resolutions to vary the size of signal from feature to feature. This method is based on feedback from experts after first employing and experimenting with the phase computer method. This data cube contains the entire dataset of 5333 flights.

These two methods contain overlap of the same period of time, with the phase computer method detailing more of the aircraft's acceleration to lift off, and possibly containing some of the initial ascent. The weight on wheels method, however, contains more information

for sensors with a higher sampling rate for the same period of time. The weight on wheels method is also constraining in the amount of choices a pilot may make, thus reducing the overall possible anomaly size. Using both methods, we extract the signals of 87 sensors over the instances in the data.

VI.2.1.2 Dimensionality Reduction with Complexity Measures

The transformed data is reduced into distance matrix cube with the complexity based measures. These measures were selected based on the experiments in Chapter V. The distance based on the Haar Wavelet along with the complexity invariant distance measure using a prediction by partial matching compressor (CiDM/PPM) were the top choices from the experiments. Based on the experimental results of the real world data set the Haar Wavelet was the better of the two measures.

Early experiments of the phase computer based data cube utilized CiDM/PPM as the dimensionality reduction measure. As mentioned, this data was built as a test cube for a variety of methods, and the CiDM/PPM measure was applied to produce initial results. Utilizing the Haar wavelet for this cube would not be ideal since the features are all the same size of 90 samples which is not a power of two, and a requirement of the Haar Wavelet Transformation. The data cube with this method was built before the complexity experiments were finished, so the size was not an original consideration. While, the data could either be reduced to 64 samples, or padded with zeros to create 128 samples, the cube is left as designed, and the CiDM/PPM measure is applied to reduce the dimensionality for clustering.

For the weight on wheels based data cube, we use the Haar Wavelet. The Haar Wavelet appeared as the best overall measure in our experiments, balancing the ability to identify significantly different signals, as well as remain sensitive to changes within signal type. The Haar wavelet decomposition is best applied to this cube without the need for padding or removal, since each feature signal is a power of two. This is due to the fact that each

sensor samples at a rate that is a power of two, so that a 32 second signal at full resolution will remain a power of two with 32, 64, 128, 256 or 512 samples.

VI.2.2 Clustering and Exploration

A hierarchical clustering algorithm using the complete link methodology is employed to construct dendrograms from the euclidean distance based dissimilarity matrix. Complete link clustering only joins two clusters together if the furthest distance between any two points in the clusters is the smallest distance value remaining in the adjacency matrix. This information is known as the linkage, and is stored for the links between single instances, as well as between the clustering of multiple instances. Complete link clustering usually yields clusters that are well separated and compact. Since we want to separate out the instances that are different from the majority in the data, this methodology should help build the clusters we are most interested in finding.

The structure of the generated dendrogram can be used as a mechanism for visualizing the unlabeled data. This means looking for clusters that break the data into a larger nominal cluster and producing a set of much smaller, possibly anomalous clusters. Beyond this visualization, the dendrogram serves the operator and expert as a marker for identifying the cutoff in the hierarchy where clusters should form for this data [71].Locating this cutoff can be done manually through this visualization, as well as searching through the a variety of cutoffs with a goodness of fit calculation such as the Cophenetic correlation coefficient [43] and the inconsistency coefficient [71], or directly from the linkage information about the dendrogram to determine a likely split. These utility measures all attempt to identify when clusters are too large and generalized, or still too small and specific by comparing the distances in the clustered sets with the overall distances across the clusters. We choose to operate a search that is interleaved with the use of the linkage criterion, and through human guidance to identify likely clusters. The work flow is illustrated in Figure 23.

Using the amount of total linkage contained in a cluster, we have the algorithm select

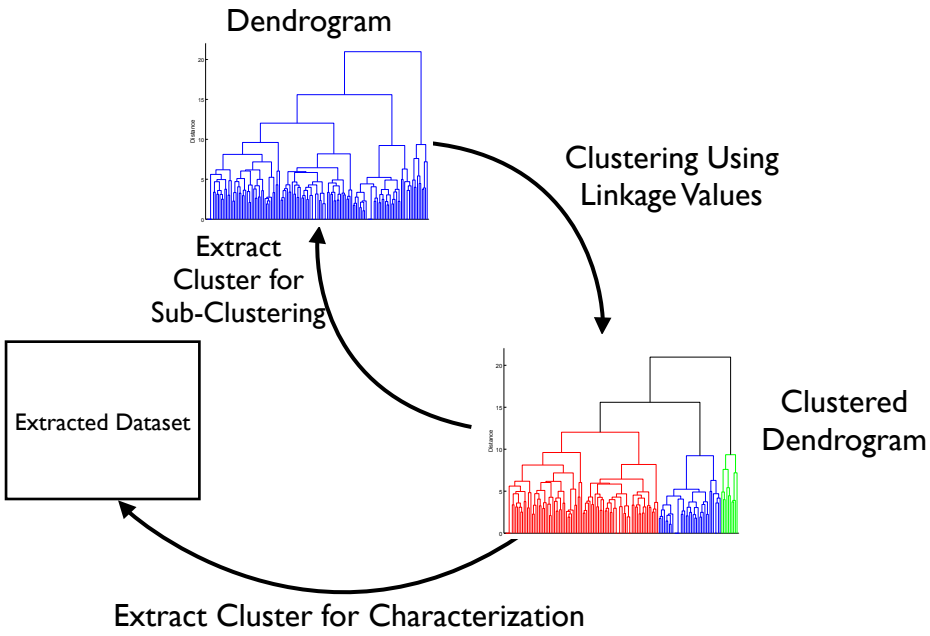

Figure 23: Clustering Work Flow

clusters based on how much of the linkage is contained at that point in the hierarchy. The base value we use is that a cluster may contain no more than 70% of the total linkage in the dendrogram. This initially skews towards larger clusters, but it will help identify very large and compact regions of data that may be identified as nominal. If one single cluster is identified, and thus no differentiation, we lower the threshold until a split has been found. The user will identify the likely nominal clusters, and look to subdivide the smaller clusters into possible sub-cluster themselves. Producing a smaller dendrogram from each original cluster, this procedure uses the same linkage criterion as above. This subdivision continues until a reasonable stopping point is identified by the user, or the linkage parameter used for clustering drops below a threshold. Based on these choices, the final cutoff is determined by identifying the overall distance in the original dendrogram that can partition the data according to these clusters.

The output at this stage is a series of datasets. Each dataset is labeled as either a likely

nominal set, or as a set to characterize by the expert. These smaller sets contain the instances that are worth characterizing for the differences from the majority in the entire dataset and the larger labeled nominal set will be used as the basis for finding the cause of these anomalies.

VI.2.3 Feature Selection and Characterization

Given a possible dataset of anomalies found through the clustering operation, the question is which features are separating these anomalies from the nominal set. Feature selection for these instances is governed by the signals where the anomalies have the greatest differences from the nominal set. It stands to reason that these differences can be found by examining the feature by feature distance matrices from the data cube produced by reducing the signal dimensionality through complexity measures. Finding the distances which are the greatest for the anomaly, or set of anomalies will produce an ordered set of features to examine for an expert. We refer to these features as "significant actors."

The features are selected by

1. For each feature, select the matrix made up of relevant distances for an anomaly, or set of anomalies and the selected nominal set. This is known as the anomalous distances.

2. Selecting the the distance matrix made up of only nominal flights. This set is referred to as the nominal distance matrix.

3. Removing the duplicates in the nominal matrix so it is a vector of unique distances for the nominal group. This is known as the nominal distances.

4. Performing a two-sample Kolmogorov-Smirnov test [96, 106] on anomalous distances and the nominal distances to test the null hypothesis that the samples come from the same continuous distribution.

5. If the test rejects the hypothesis, calculate the mean dissimilarity for the anomalous distances.

6. If the hypothesis is not rejected, skip that feature.

7. With a collection of average distances, sort them in descending order. This represents the list of significant actors from most offending to least.

The choice of an average dissimilarity for a the anomalous distances with the nominal set by itself would not take into account noisy sensors, which may have higher dissimilarity across all aircraft and flights, and thus the complexity measure itself is not suitable for ordering. We mitigate this issue by using a probability test to identify the likelihood that the anomalous distances would be drawn from the same distribution as the nominal distances. Since we do not want to make any assumptions about the distribution, and the fact that the distances represent continuous values, we decided to use the Kolmogorov-Smirnov test to adjudicate this matter. If the null hypothesis is rejected, we can feel confident that the distances are likely different between the two groups. If it is not, we can believe that the distances are all either very similar, in which case the average distance would have been quite low, or the values are quite spread out among nominal and anomalous distances meaning that the sensor is noisy and unreliable for identification.

Another choice in this matter is to either group the anomalies together, or look at the significant actors one by one. Ranking the significant actors as a group of anomalies will attempt to identify their unifying characteristics. If the anomalies only similarities are that they are sufficiently different from the nominal cluster, the distribution of possible significant actors will dilute the average feature distances. One anomaly against a nominal set will produce the significant actors for that anomaly. This could be burdensome for anomalous groups that are larger than a few instances. Also, it requires the expert to identify single anomalies, rather than being able to draw conclusions across a sample of possible anomalies.

VI.2.3.1 Characterizing Anomalies

Once ordered, the process of showing the top features to the expert for characterization of the anomaly is based on a tiered system. The ten highest significant actors are presented, followed by the next ten, until the distances drop below a threshold of normalized distance, such as 0.1. The expert recalls these tiers as they feel they are necessary to produce more information to characterize the anomaly. These tiers focus the expert, as well as provide information about their relative importance compared to the other features. This may help guide the expert as they focus on the possible anomalies.

The features are displayed to the expert through plots of the signal for that feature from the data cube. The plots clearly mark the anomalous signal, but also plot a random sample from the nominal set as well. This sample from the nominal set provides context for the anomalous signal, to show the expert how the aircraft typically responds. Identifying an anomalous takeoff will be easier through the lens of likely operations based on the samples from the nominal set.

VI.2.3.2 Characterizing Anomalous Groups

Using the single anomaly versus the nominal cluster to build significant actors, we developed a process to explore possible relationships in the data. Given a cluster of possible anomalies, the output should be a partitioning of the anomalous group in a manner than conveys a collection of significant actors that unify the different partitions.

The procedure for this is to:

1. Merge the top significant actors for each anomaly into a set.
2. For each anomaly, collect the distances for each feature in the set.
3. Cluster this data using a Targeted Projection Pursuit lgorithm [42].
4. From these clusters, identify the features Targeted Projection Pursuit finds most significant.

5. For each Partition found, associate that partition with a set of most significant features for visualization.

Targeted Projection Pursuit is a method of visualizing the process and output of using methods such as PCA, and Singular Value Decomposition [56] in identifying the features that most effectively separate the given data into partitions. This process is partially interactive, but the results of it are a series of features for each cluster. These features in turn can be used to identify possible relationships in the data.

VI.3 Results and Case Studies

We first present the basic results from applying PCA-DBSCAN and MKAD to the data cube built from the phase computer method. This provides a baseline for the how these approaches perform in data constrained to a specific phase, and without labeling. We follow this with the original application of CiDM/PPM to the phase computer based data cube, and identify and characterize the primary results. These anomalies indicate why we utilize the weight on wheels based method for the remainder of this work. We then use the Haar Wavelet based distance on the weight on wheels data. We examine the results of our approach in helping to characterize found anomalies. We present several anomalies found in our methods and assess their impact on aviation safety.

VI.3.1 Application of PCA-DBSCAN to Aviation Data

The implementation of PCA-DBSCAN comes from our own tested implementation of the PCA algorithm, and a DBSCAN implementation in the MATLAB exchange written by Michal Daszykowski. The application of PCA-DBSCAN starts by unrolling the data cube from the takeoffs calculated by the phase computer into a 2116 instance x 7830 feature data set. During the PCA step, after eigenvalue decomposition, we chose to retain 98% of the variance in the features. The application of PCA with 98% of the variance resulted in a reduced feature set of 13 orthogonal features.

This dataset was clustered with DBSCAN. There are two parameters, the threshold to declare to features similar to one another, and the minimum number of instances needed to produce a cluster. The threshold can be estimated from the data and the minimum number required for a cluster. We varied the minimum number to form a cluster parameter from 3 to 5 to 10 to 150 to 500. In each case, DBSCAN reported the same cluster assignments, identifying 7 instances that were considered outliers and marked as anomalous by the method and producing only one cluster containing all the rest of the instances.

Since this method is unable to identify the significant actors, we plotted every feature for each anomalous instance. We found that for 6 of these instances, the primary significant actor was the sensor for the second fuel tank, which was reporting zero. Inspection of other sensors that related to fuel quantities and fuel flow for the engines that utilize the fuel from that tank clearly indicate non zero values of fuel in tank two, contrary to the fuel quantity sensor reading. While interesting to identify, this anomaly does not directly impact aviation safety.

The final instance is where the phase computer method includes a recorded "flight" that was simply a ground test, meaning the mechanics fired the engines but the aircraft did not leave the ground. Since the computer doesn't use contextual clues such as altitude, and only the thrust of the engines, instances like this are included in this data cube of flight data. A ground run does imply some form of problem with the aircraft, resulting in the test, but the root problem associated with the test may be hard to discern.

While PCA-DBSCAN does identify a set of anomalies, the process to characterize them is arduous, since it requires at least another processing step on the original data to identify the significant actors. The quick and easy method applied above is due to the small size of outliers and would be very difficult in the case where there were more flights flagged as initially anomalous.

VI.3.2 Application of MKAD to the Aviation Data

The implementation of MKAD is from the data mining group NASA Ames. The base requirement is that the data cube be replaced by N flat comma separated value files, where N is the number of instances. Each line in the file is a sensor. This is loaded as a matrix into MATLAB. This requires each sensor to be the same length, thus the primary reason why the phase computer based method of the data cube down-samples the sensors that operate at greater than 1Hz. Due to this down sampling and the need for a rectangular structure for each file, this guided the 90 seconds captured by the phase computer method. Since the MKAD implementation resulted in loss of data through down sampling, we attempted to make this up by providing as much information about the takeoff phase, even if it included time past when a takeoff phase is completed.

These files are loaded, preprocessed to make the continuous sensors symbolic (the discrete sensors are considered to already be transformed). The parameters for this transformation, the window length, and the alphabet size are not made directly accessible to a user, but we managed to make them more transparent for our own experiments. The dataset is considered to be made up of nominal instances. Since we do not have a labeled nominal set the MKAD implementation attempts to find one through the use of distribution testing. With parameters to set a max sample size for the training set, and sigma values to use as the limit of the nominal range, MKAD attempts to shrink the training set down to a core group. A one-class SVM is then applied to this training set, and the pruned data is applied back as the test set. The flagged anomalies are then processed with SequenceMiner to identify the significant actors with relevant plots.

Starting with our entire data cube, we first used the default parameters for the symbolic preprocessing. We did, however, change the distribution testing parameters to encompass as much of the data as possible, leaving a single sigma bound and setting a max for the training set at the size of our data. With these set parameters, MKAD built a training set of 100 instances and in turn, produced results, where almost every single instance was flagged

as a possible anomaly. Furthermore, when Sequence Miner was employed by MKAD to characterize the anomalies, the large number caused the system to crash. This result, based on the number of possible anomalies was not unexpected, since SequenceMiner was not built with large data in mind. We increased the sigma parameter to 3 times the original sigma and reran the experiments. This time, 430 instances were used in the training set. This dropped the number of flagged anomalies to just under 500. This was still too many for SequenceMiner to handle without crashing MKAD.

The results caused us to write methods that made the symbolic transformation parameters more transparent. Since we were dealing with a small period of time, and had down sampled the signals, the symbolic transformation algorithms did not have sufficient data to reliably perform the signal to symbol transformation. As we expanded the alphabet size used for producing a finite domain of values (which the original implementation had limited to 20), and decreased the window size used for generating a new symbol from the signal, we were unable to find a tuned set of parameters that could find a more manageable set of anomalies that did not crash the system.

MKAD has high potential, but requires many choices that limit its generality and initial effectiveness. From the need to have large enough signals that symbolic transformation is effective, to the number of tunable parameters, to the need for a small enough number of anomalies to run SequenceMiner efficiently on a standard computer, the current implementation is brittle without significant curation of the data and some assumptions about where to look for anomalies.

VI.3.3 Application of Approach with CiDM/PPM and Phase Computer Based Data Cube

The application of PCA-DBSCAN and MKAD to this work showed mixed results. In general, we found anomalies that fell into either a set of flights with a broken fuel gauge, or ground tests. In the case of MKAD, we discovered that while the approach has a good

theoretical basis, the data must be carefully managed to accommodate the limitations and constraints of the implementation. As we ran the complexity experiments, we decided to run the best Compression-Based method on the same data to explore the data cube and compare our approach to the first two methods.

Using the CiDM distance metric with the PPM compression measure, we started with the $2116 \times 87 \times 90$ data cube and computed the pairwise single feature distance matrix. The results of the reduction with the complete-link agglomerative clustering algorithm to generate the dendrogram shown in Figure24. The solid rectangle to the right of Figure 24indicates a group of 3 flights that represent a possible anomalous situations for more detailed investigation.

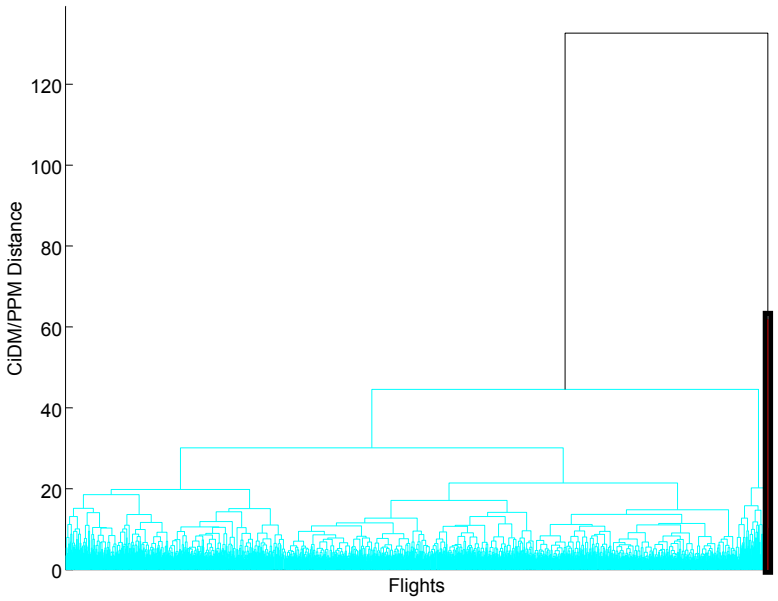

Figure 24: Dendrogram of the Agglomerative Clustering for CiDM/PPM Reduction

Of these three flights, two can be classified as ground runs of the aircraft. These match those found by PCA-DBSCAN. The last flight is one where the significant actors match an inoperative engine. This is found using weight on wheels data with the Haar Wavelet

transform and is discussed in more detail in Section VI.3.4.1. Certainly this is an important flight to find, which was missed by PCA-DBSCAN.

The results from this analysis show that CIDM/PPM is more conservative in identifying anomalies than PCA-DBSCAN. It does not identify the broken fuel gauge pattern of PCA-DBSCAN, but identifies a flight with inoperative engine, which is far more important. We understand that this phase computer based takeoff data used for detection is prone to issues with ground runs, and has been down-sampled, thus removing information that may help identify new anomalies. These results on the phase computer data motivate our use of weight on wheels data in the following section.

VI.3.4 Application of Approach with Haar Wavelet and Weight On Wheels Based Data Cube

Using the weight on wheels based data cube, we also now apply the Haar Wavelet based distance measure that was found to be successful in our earlier experiments. This method for calculating takeoffs should also remove the possibility of ground runs in the data, since to effectively have zero weight on the wheels, the aircraft sensors should record a wheel off the ground event. This data also contains a more diverse selection of instances, including all 5333 flights from our curated data set. This increases the possibility of characterizing nominal flight more accurately and also for finding additional anomalies in the data.

Using the combination of the Haar Wavelet transform and the Euclidean distance measure to reduce the data cube, the cube is reduced to a set of dissimilarity matrices, each corresponding to one of 87 features. This was transformed into a single distance matrix and we applied our clustering scheme to identify likely anomalies. Figure 25 shows the initial clustering results. The clusters are colored, but the anomalous cluster is indicated by a rectangle on the right of the figure. It is clear that the flight instances are broken into a large set which is considered to be nominal and a smaller set which are likely to be the anomalous flights. The small set contains 138 flights. The largest cluster is being left aside

as a nominal set. We extract the smaller set of 138 flights and cluster them separately. Figure 26 shows this sub-clustering which produced three clusters. These clusters are also colored, but the plot also contains the cutoff level applied in the dendrogram to determine how the three clusters are defined. From right to left, anomaly cluster 1 contains 9 flights, anomaly cluster 2 contains 39, and anomaly cluster 3 contains 90 flights respectively. All together these 138 flights make up just 2.5% of the total flight instances. In this work, we focus first on anomaly cluster 1, to show how the expert investigates these different anomalies. We then explore anomaly cluster 2, using our technique for characterizing a group of anomalies.

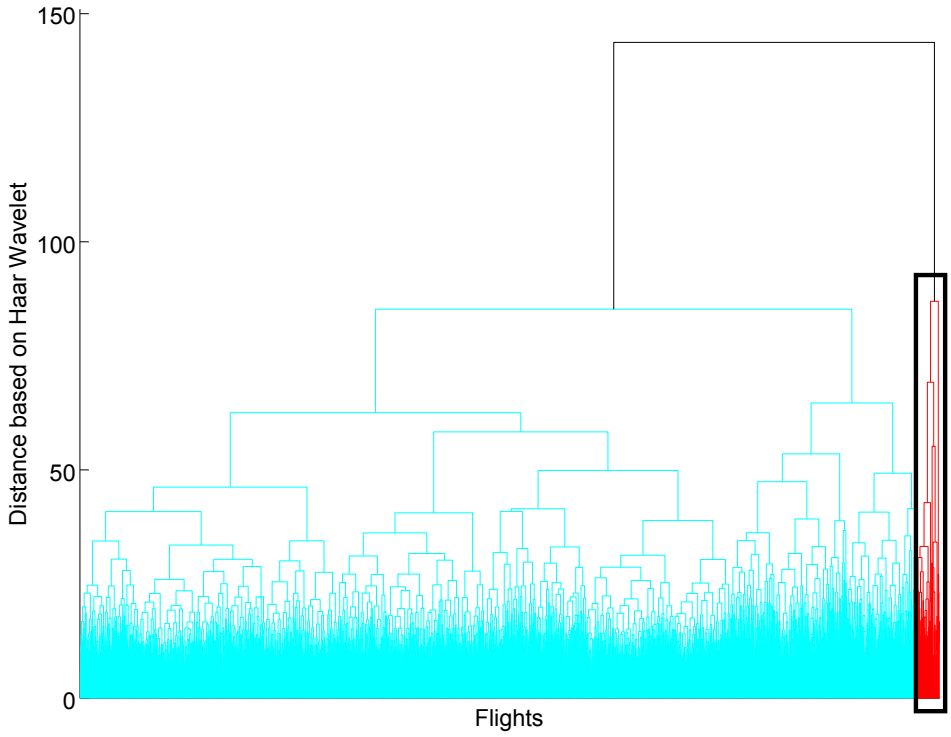

Figure 25: Full Dendrogram based on Haar Wavelet Transformation with Initial Clusters

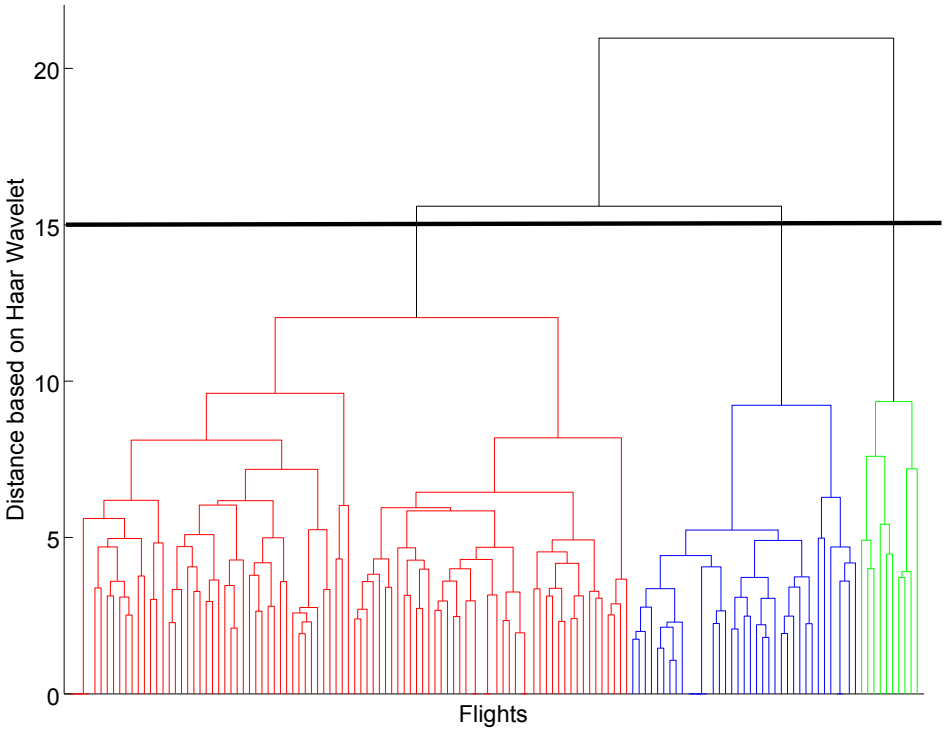

Figure 26: Sub Cluster Dendrogram based on Haar Wavelet Transformation with Cutoff for Three Anomalous Clusters

VI.3.4.1 Characterizing Single Anomalies in Cluster 1

Tables 29 and 30 shows for each anomaly in anomaly cluster 1, the ID for that flight in the data cube, the top significant actors, and a preliminary group ID for organizing our results and comparing these anomalies. From this table we can draw some general conclusions. There are three general groups of significant actors in these anomalies. The first group is the singleton of flight 5186. For this flight, the engine sensors for the second engine are the significant actors. This is in contrast to the anomalous flights of group 3 which show engine parameters for three of the engines as significant actors. The conclusion, therefore, is that the anomaly in flight 5186 is specifically associated with engine two. The second group of anomalies are ones that include environmental sensors such as total pressure and altitude as a contextual attribute. The last group, as mentioned earlier, contains several measurements for a variety of engines for the aircraft that appear to contribute to the anomaly. We examine these groups in more detail.

Group ID	Flight ID	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5
1	1256	N1.1	N1.3	N1.4	N2.1	N2.3
1	3316	N1.4	N1.3	N1.1	N2.3	FF.4
2	5006	EAI	BAL2	BAL1	ALT	VRTG
2	5007	BAL2	BAL1	ALT	PS	PSA
2	5148	BAL2	BAL1	ALT	PT	PS
2	5152	BAL2	BAL1	ALT	FQTY.2	PS
2	5153	BAL2	BAL1	ALT	FQTY.2	PS
2	5193	BAL2	BAL1	ALT	VRTG	PT
3	5186	N2.2	N1.2	ATEN	EGT.2	PLA.2

Table 29: First through Fifth Significant Actors for the Anomalies in Cluster 1

Group ID	Flight ID	Actor 6	Actor 7	Actor 8	Actor 9	Actor 10
1	1256	FF.4	N2.4	FF.1	VRTG	PLA.4
1	3316	N2.1	VRTG	PLA.4	N2.4	FF.1
2	5006	PS	PT	PSA	OIT.1	LATG
2	5007	PT	RUDP	OIT.4	VRTG	OIT.2
2	5148	PSA	LONG	BLAC	AOAI	AOAC
2	5152	PSA	PT	OIT.3	LATG	VRTG
2	5153	PSA	PT	OIT.3	LATG	VRTG
2	5193	PS	PSA	OIT.1	LATG	BLAC
3	5186	FF.2	LATG	VRTG	OIP.2	FQTY.2

Table 30: Sixth through Tenth Significant Actors for the Anomalies in Cluster 1

Flight 5186 is one of the most interesting anomalies in the entire dataset. The significant actors that relate to engine two, such as the one for Engine Temperature in Figure 27 show that the engine is not producing any power. All significant actors listed for engine two indicate that the engine appears to be nonfunctioning. Similar to the ground tests in the PCA-DBSCAN, the expert asked to look at a navigational sensor like altitude in Figure 28, as well as values for those sensors on other engines such as engine temperature for engine four in Figure 29, and core speed for engine one as illustrated in Figure 30. Together, these sensors indicate that this flight was indeed at full takeoff at a normal altitude, with the other engines registering a slightly higher than normal power. The expert examined these significant actors and came to the conclusion that engine 2 was not working during the flight. If this were a regular airline flight then it would represent a highly unusual situation, with strong safety implications.

While coming to this conclusion, the expert postulated that it was known that engine 2 was inoperative, and that this flight was the aircraft returning the from a remote airport to a hub for maintenance on engine two. This was verified by looking at the latitude and longitude of the raw data for the instance. The originating location for the flight was an airport in Ohio. The destination was to the hub airport of the airline. The expert came to the conclusion that this is a non-passenger flight but it would be difficult to verify this, since this type of aircraft is certified to fly with 3 engines for smaller distances. The expert believed the closest way to verify this is to check the engine serial number for the anomalous flight and compare it to subsequent flights for this aircraft. We found that after a number of ground runs after this flight, the second engine was replaced (different serial number) before the next full flight. The origin and destination of the anomalous flight coupled with the different serial number for the engine strongly suggests that the expert's hypothesis is correct.

The second group contains the collection of environmental sensors and the altitude as the significant actors. We choose one of these flights, 5006 as the representative sample.

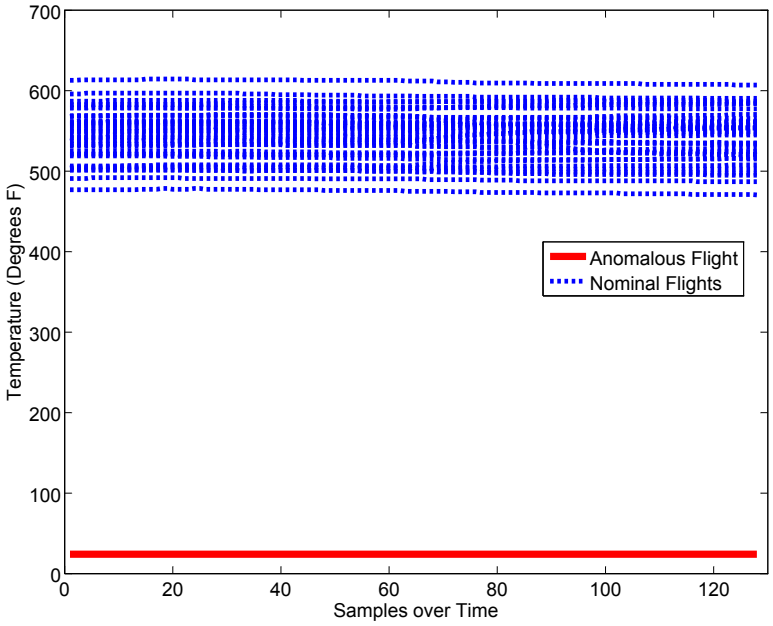

Figure 27: Temperature of Engine Two at Takeoff for Flight 5186

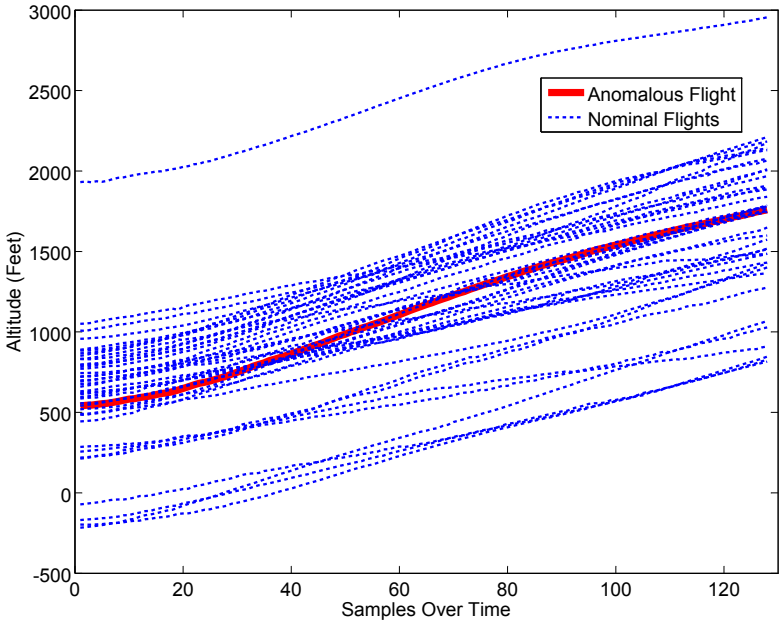

Figure 28: Altitude at Takeoff for Flight 5186

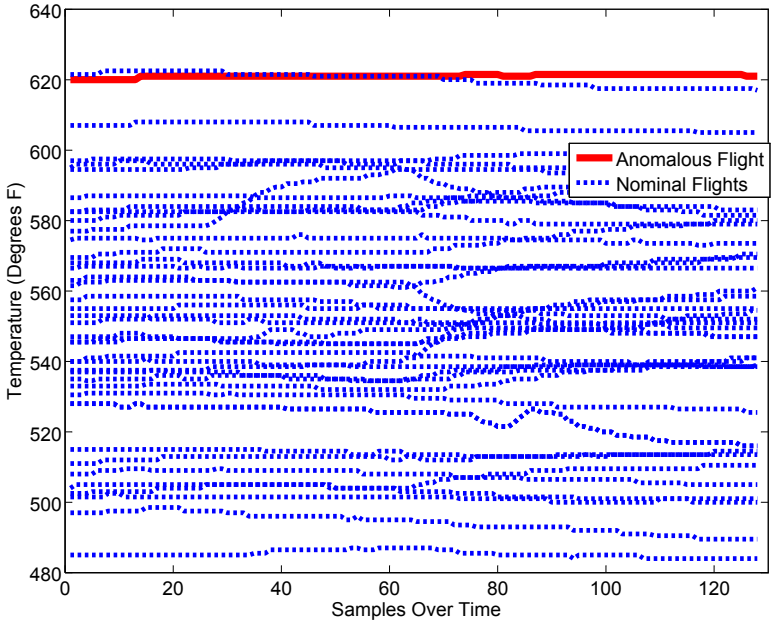

Figure 29: Temperature of Engine Three at Takeoff for Flight 5186

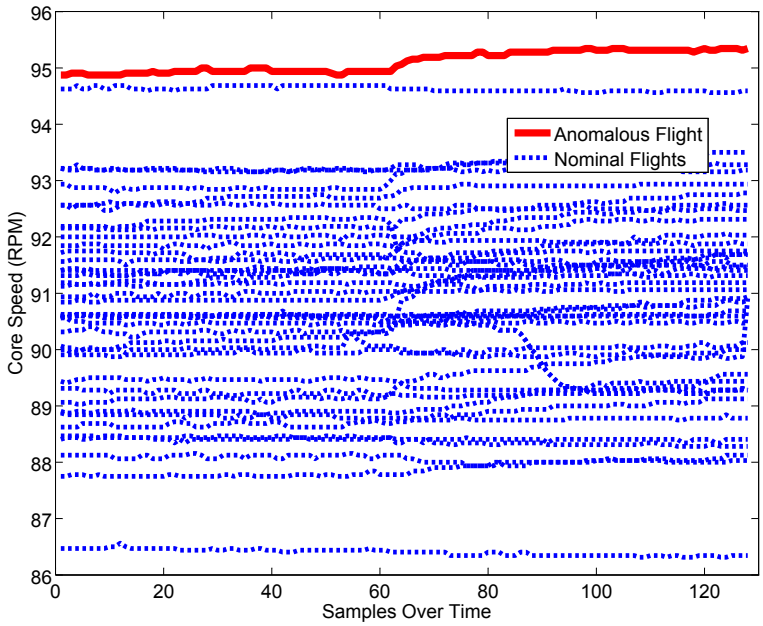

Figure 30: Core Speed of Engine 1 at Takeoff for Flight 5186

All flights in this group have the same basic altitude, 7900 feet at takeoff as shown in Figure 31. The total air pressure and other environmental variables are also related to the location of this takeoff. Corroborating this information was the examination of the takeoff location which is near a mountainous region of the United States. Since the radio altitude was not ranked in the top 10 significant actors, this would appear to eliminate the fact that this takeoff was otherwise anomalous compared to the nominal values, just that the location was rare for this airline. The expert asked to examine an engine parameter. We selected from the next ten ranked significant actors, an engine temperature for the third engine illustrated in the plot in Figure 32. The fact that the engine is running at a higher power confirmed the experts suspicions that these anomalies are similar to the results with CiDM/PPM and PCA-DBSCAN and are high energy takeoffs. Although we rediscovered this environmental anomaly, this approach was more helpful. The significant actors immediately point out that this is an environmental based anomaly by flagging environmentally sensitive measurements, rather than features that pointed to the performance of the aircraft or the pilot. This shows that these significant actors can potentially be used as contextual attributes such as the altitude sensor for anomaly detection. The expert again confirmed that these are high altitude takeoffs and would need to be filtered by this method in the future, as they constitute a false alarm to an expert who was aware of the possibility of flights from this environment. As with the previous analysis, we agree that this isn't a safety issue, but the method is effective at identifying rare operating environments.

A last observation about this anomalous group involves the sensitivity of the complexity distance based on the Haar Wavelet. Similar to our experiments, this is the situation where the slopes of these lines are similar, and instead there is a shift in the y-intercept. There is a range of about 1500 feet in the random sample of plotted nominal values. That provides quite a range of possible geographical locations that our method can encompass as a "normal" environment for the aircraft to be operating. For this anomaly, the fact that the location is also rare in the data, means that this is certainly worth catching. Our method

ranked significant actors that immediately let the expert know that the anomaly was likely to be environmental.

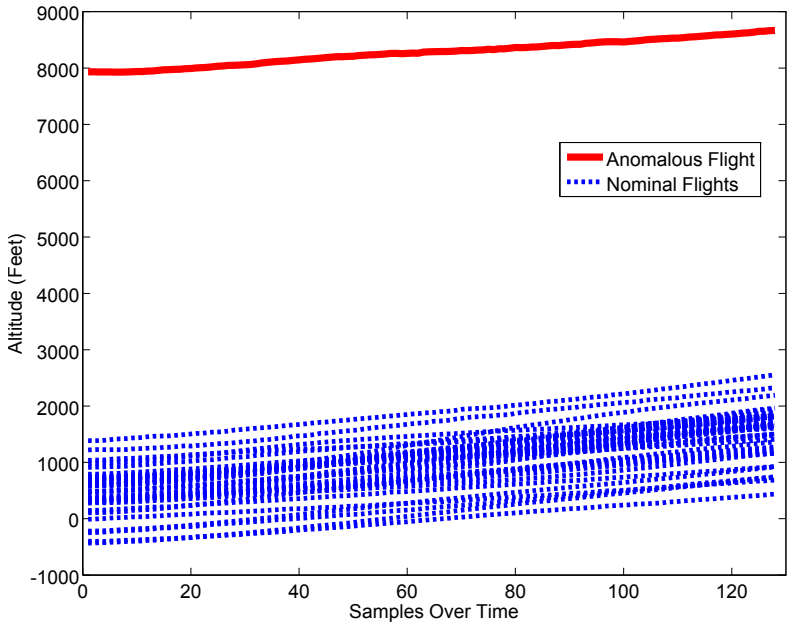

Figure 31: Altitude at Takeoff at Takeoff for Flight 5006

The third group contains two flights, and each has a selection of the engine parameters from the different engines relating to the core speed and the fan speed in the engine. This includes the fuel flow sensor for the first and fourth engine and the power level angle for the fourth engine. Figures 33 and 35 show examples of the fan speed for the same engine in both flights. Plotted with a selection of 50 flights from the nominal set, the dips in both sensors are quite large. Also of note, the fact that these are being clustered close to one another is reasonable considering the nature of the drops in both flights. After looking at a series of significant actors for each flight, the expert came to the conclusion that these two flights were quite different in terms of what they mean for a takeoff. The expert makes use of other significant actors such as flight path acceleration to place the change in engine parameters in context for each flight. The flight path acceleration for flight 1256 and shown in Figure 34 shows that the airplane slowed down off after takeoff. The expert postulated

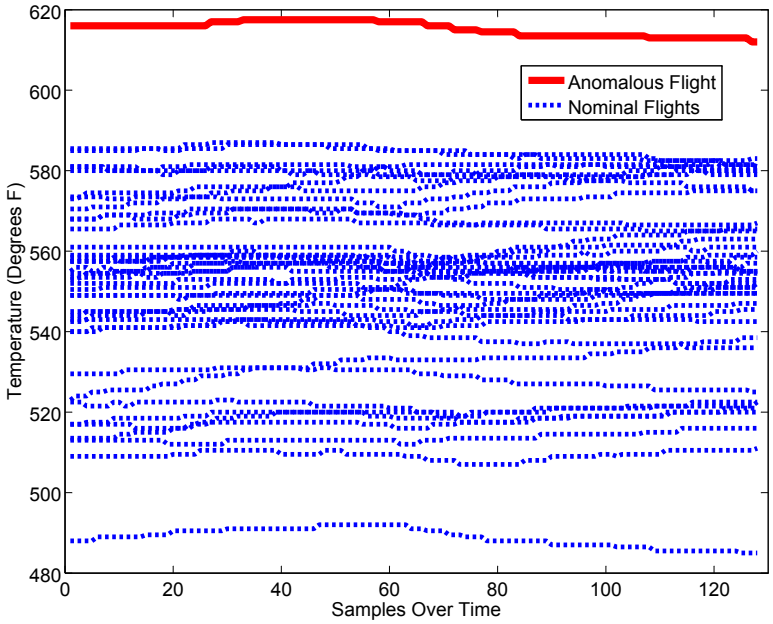

Figure 32: Temperature for the Third Engine at Takeoff for Flight 5006

that this could be a part of the flight plan, since all engines are consistent in their changes. The expert believed that there is nothing unusual about this type of flight and the final verification was that the the automatic throttle did not change mode as it should have, but that very likely an auto pilot decision.

Flight 3316 while initially appearing similar to flight 1256 is quite different. The expert believes that the auto throttle disengaged in the middle of the climb. The automatic throttle is designed to maintain either constant thrust from the engines, or as controller to maintain constant speed. The behavior in the significant actors is unusual because that means that the auto thruster decided to switch from maintaining speed for a takeoff to a setting that applied constant thrust. The change in setting in the auto thruster indicated that the plane is on the verge of a stall. This is verified by the flight path acceleration sensor shown in Figure 36. The sensor was trending up and if the plane continued to operate along that trajectory, there was a chance of a stall. The expert then explained that the automatic throttle would switch to a possibly lower thrust setting to compensate for this situation. By examining the engine parameters, the expert verified that all the engines responded in

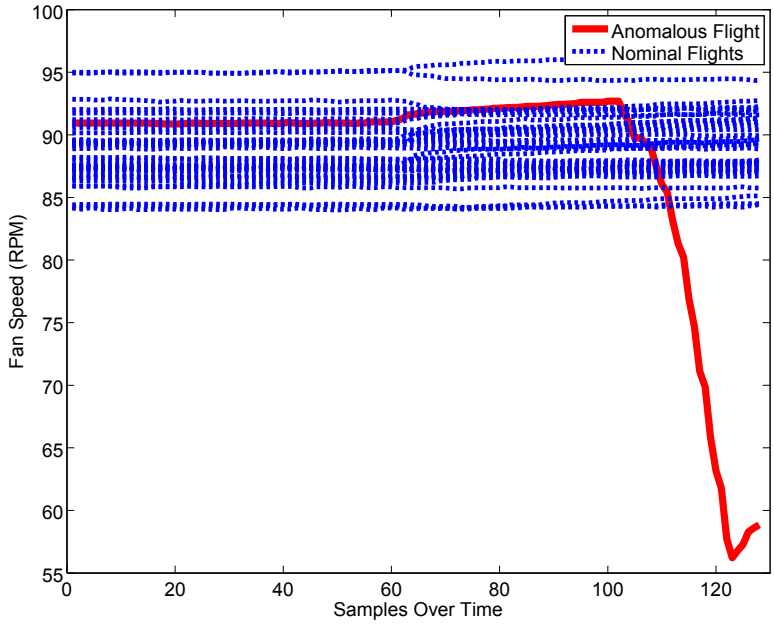

Figure 33: Fan Speed of Engine 3 at Takeoff for Flight 1256

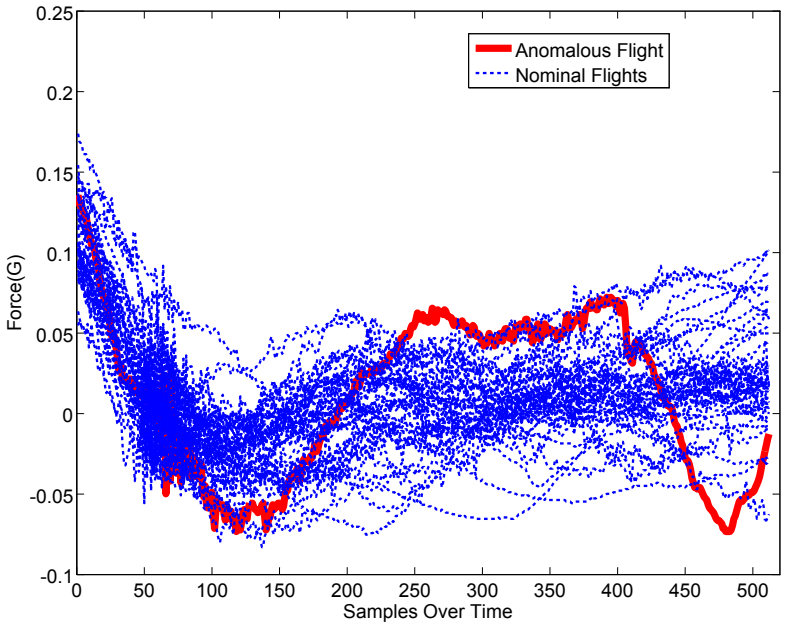

Figure 34: Flight Path Acceleration at Takeoff for Flight 1256

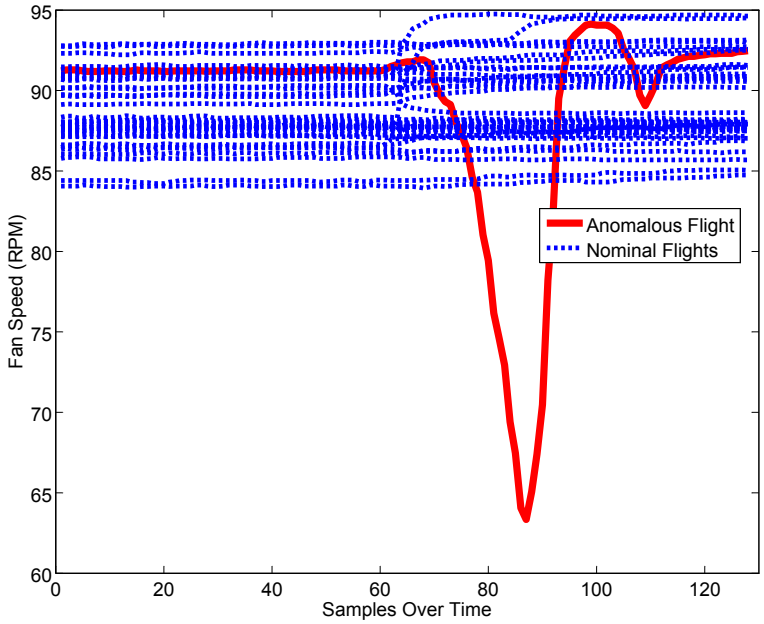

Figure 35: Fan Speed of Engine 3 at Takeoff for Flight 3316

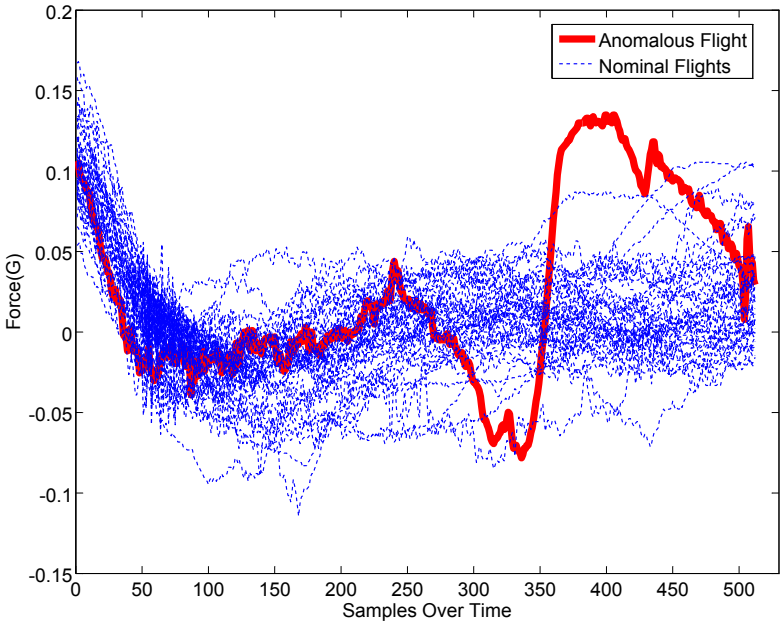

Figure 36: Flight Path Acceleration at Takeoff for Flight 3316

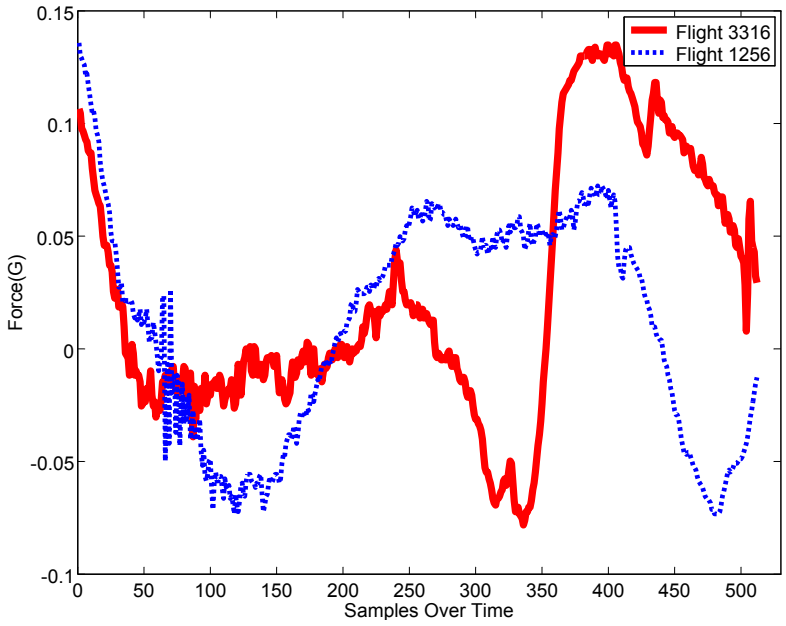

Figure 37: Flight Path Acceleration at Takeoff for Flight 3316 and Flight 1256

an appropriate fashion to this throttle command. This meant that the aircraft responded and slowed down (the acceleration drops at around 350 samples). Figure 37 shows the acceleration for Flight 1256 plotted on top of Flight 3316. This shows that while both were clustered together, the expert was better able to explain the two situations as different by examining a set of significant actors. While flight 3316 certainly does not demonstrate a flaw in the aircraft, the expert found the anomaly interesting and would ask "why did the airplane accelerate in such a fashion and come so close to a stall condition?" Since the expert could not determine the root cause, these incident would cause them to seek more information. The expert would also want to use this in future to guide other pilots away from taking action, and instead assure them that the aircraft autopilot would compensate sufficiently to correct for the situation.

VI.3.4.2 Characterizing the Group of Anomalies in Cluster 2

The previous cluster was a small collection of anomalies, and one that could be examined through a manual process of exploring the lists of significant actors. The second

cluster contains 39 flight instances. While this may be possible to be examined in the same manual manner, our method provides a way of guiding an expert through a high level characterization, and allowing them to prioritize flight and anomalies from the set they want to examine first.

Following our procedure for characterizing a group, we first found that there were 52 unique sensor listed as the top 10 significant actors for anomaly cluster 2. From this we created a dataset of 39 instances, each with 52 features, one for each sensor. Each feature is the average distance for that sensor of the instance from the nominal set. We then applied targeted projection pursuit. This method is partially interactive. The point of the targeted projection pursuit is to find a partitioning that splits the data effectively. This split provides the expert with guidance about what to look for in the cluster. Targeted projection pursuit also calculates significance of each feature. The interactivity of this process can isolate potential significant actors that may separate anomalies in a one of the partition groups.

A clustering of two was found and Table 31 shows a list of the relevant significant actors that are found during this interactive exploration, with their significance ranking in splitting the flights into the clusters. The feature with the largest differences for separating the two clusters was the altitude sensor. This was true for 31 of the 39 anomalous flights. This is quite helpful to the expert, as it already indicates a likely contextual issue with location. The next three sensors are all the Fan Speed, but for three of the 4 engines. These features appear to most closely group the remaining 8 anomalies. Targeted projection pursuit shows that this second cluster is less cohesive than the larger cluster based on altitude. This is further indicated by the sensors indicating the automatic throttle is engaged and the bleed valve position. The automatic throttle sensor indicates when the computer is set for a specific thrust and is a binary value. The bleed valve position sensor measures what position the actuator that bleeds the air from the turbines is set. The bleed valve is often used to produce compressed air to pressurize the cabin, or de-ice the wings. Targeted projection pursuit shows that these sensors each isolate one of the 8 flights in the smaller cluster. Based on

this information, we can begin to examine, the very large cluster based on altitude, and the other 8 we can examine similar to the approach used above, but looking at these sensors found through the use of targeted projection pursuit.

Rank	Sensor	Note
1	ALT	Main Sensor Separating Partitions
2	N1.1	Groups Remaining Non-Altitude Anomalies
3	N1.2	Groups Remaining Non-Altitude Anomalies
4	N1.3	Groups Remaining Non-Altitude Anomalies
5	Automatic Throttle Engaged	Significant for only one flight
6	Bleed Valve Position	Significant for only one flight

Table 31: Significant Sensors Found through the use of Targeted Projection Pursuit

Examining the altitude sensor for the larger partition of 31 flights, we discover more high altitude takeoffs due to location. As Figure 38 illustrates for flight 5332 from this set, these flights occurred at a high altitude. Like those found in anomaly cluster 1, these are quite a bit higher than the nominal range. The fact that these flights group together but not with the set found in cluster 1 would indicate that these flights possess a modest difference. In order to investigate why these flights were separated, we plotted the significant actors for the anomalies in this cluster with the high altitude takeoffs found in anomaly cluster 1.

Figure 39 shows a barometrically correct altitude plot for a the same sample in Figure 38 but compared to the high altitude takeoffs in anomaly cluster 1. These figures show that while both start in the same altitude range, there is a difference in the climb for this flight. Other significant actors include the engine parameters which when compared to cluster 1 are in the upper part of the range for that sample, and sometimes a bit higher. This is illustrated in the fan speed of engine 1 for the same sample used to show the altitude change and plotted in Figure 40. In general, these flights showed a likelihood of being even more aggressive at takeoff while at roughly the same altitude. The complexity measure derived from the Haar wavelet differentiates this set of high energy flights from those in

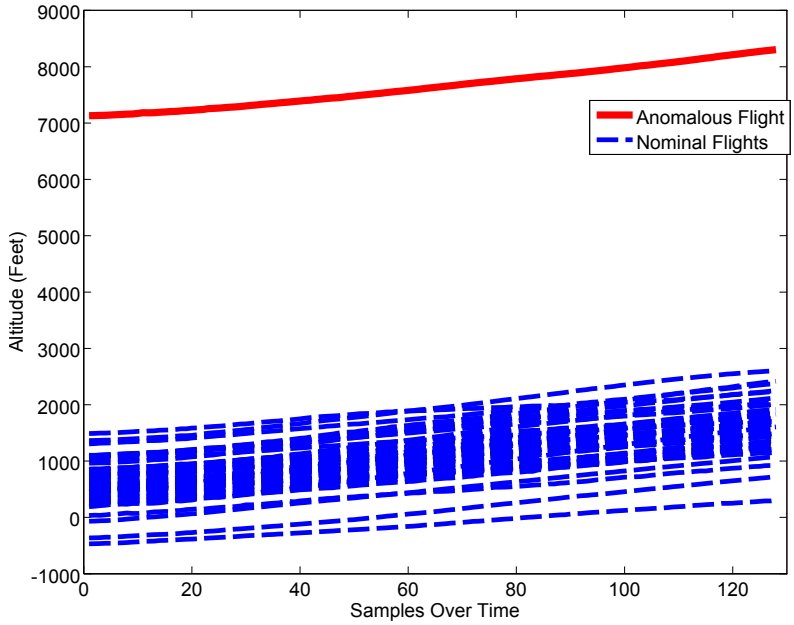

Figure 38: Altitude at Takeoff for Flight 5332

cluster 1, even if the location is the same. The expert agreed that these were an outlier, but maintained that there is no operational significance to these flights. Much like the previous set, we were the conclusion was that these type of flight should filtered out of the anomalous sets in the future.

Next we examined the smaller partition generated by the targeted projection pursuit. These eight flights can be broken into 1 group and two single flights. The group contains a series of flights containing significant actors that include engine sensors across the different engines. We illustrate with an example of one of these flights. Figure 41 shows a plot for the fan speed of one of the engines in flight 1370. All four engines for this feature as well as engine temperature, fuel flow and core speed are listed as significant actors with the same pattern. This bears some similarity to the anomaly found in flight 1256 in cluster 1 and plotted in Figure 33. Much like the differences in the high energy takeoffs, this anomaly and the flight 1256 both contain a drop, late in the takeoff, and as high altitude takeoffs could be differentiated by the aggressiveness of the climb, this behavior has a parallel, where flight 1370 has a much deeper and sustained drop in power. The expert looked at

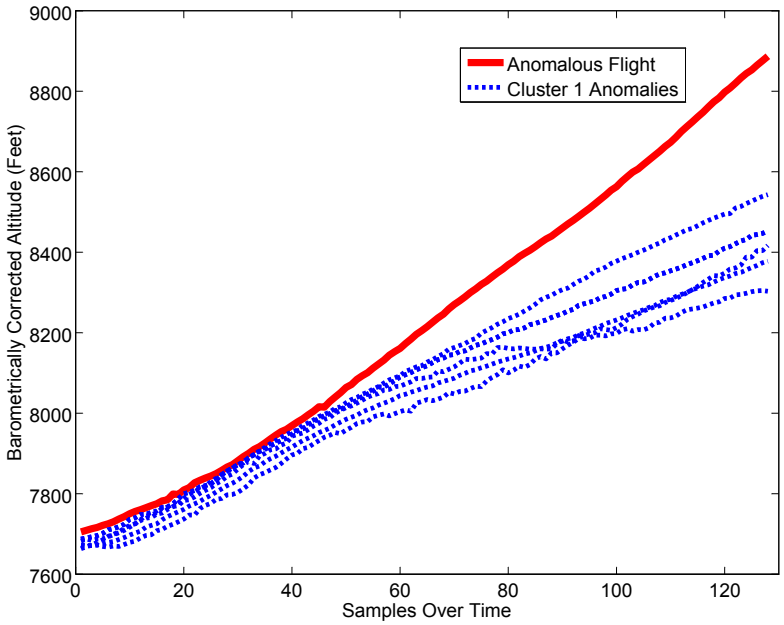

Figure 39: Barometrically Corrected Altitude at Takeoff for Flight 5332 Against High Altitude Takeoffs in Cluster 1

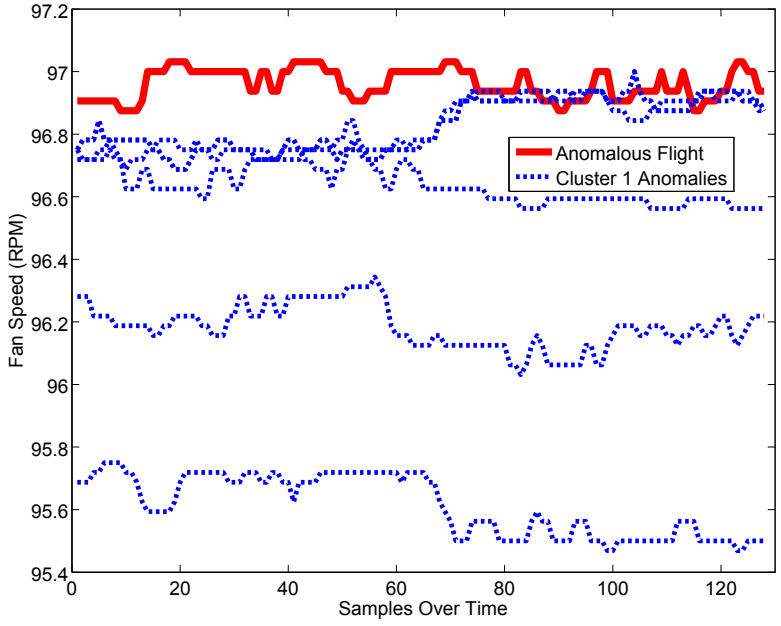

Figure 40: Fan Speed in Engine One at Takeoff for Flight 5332 Against High Altitude Takeoffs in Cluster 1

this anomaly and concluded that it was very similar to 1256 and not very interesting from an operational standpoint. The examination of the auto throttle for this anomaly found a similar behavior to flight 1256 indicating a normal slow down. The flight path acceleration for this anomaly was not a highly ranked significant actors but is presented in Figure 42 and in contrast to 3316. This shows that flight 3316 has a sharper change in acceleration, and more interesting than flight 1370.

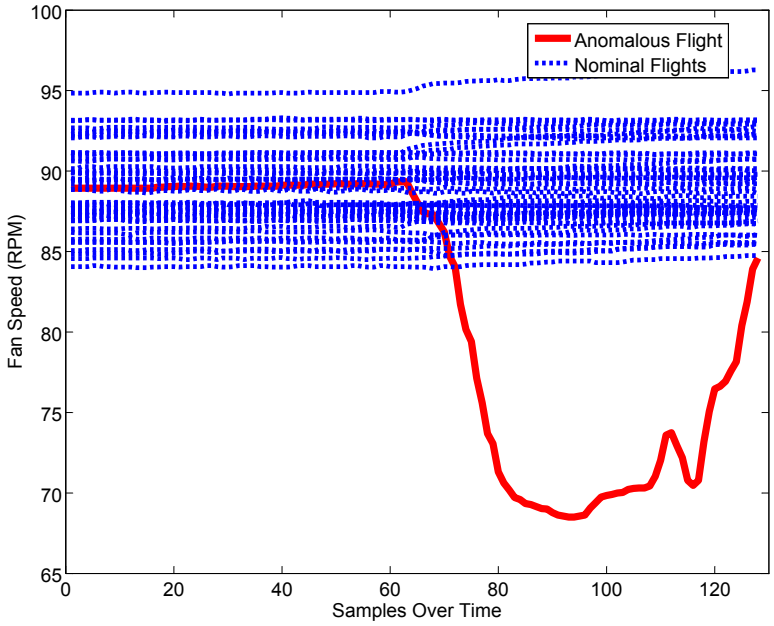

Figure 41: Fan Speed in Engine One at Takeoff for Flight 1370

As the targeted projection pursuit indicated, there was a flight in the smaller set that was the only one to be differentiated by the bleed valve sensor. The bleed valve sensor for flight 4893 is illustrated in Figure 43. This plot shows two types of nominal behavior, either a zero position meaning the valve is closed, or a position at 8 meaning partially open. In both cases, the signal remains flat. The anomaly shows a signal that starts at 8, but late in the takeoff, the bleed valve changes positions to 12 which is more open than 8. The expert was interested enough to request the entire flight instance for further examination. The expert explained that the bleed valve is normally open throughout the flight. There is one for each

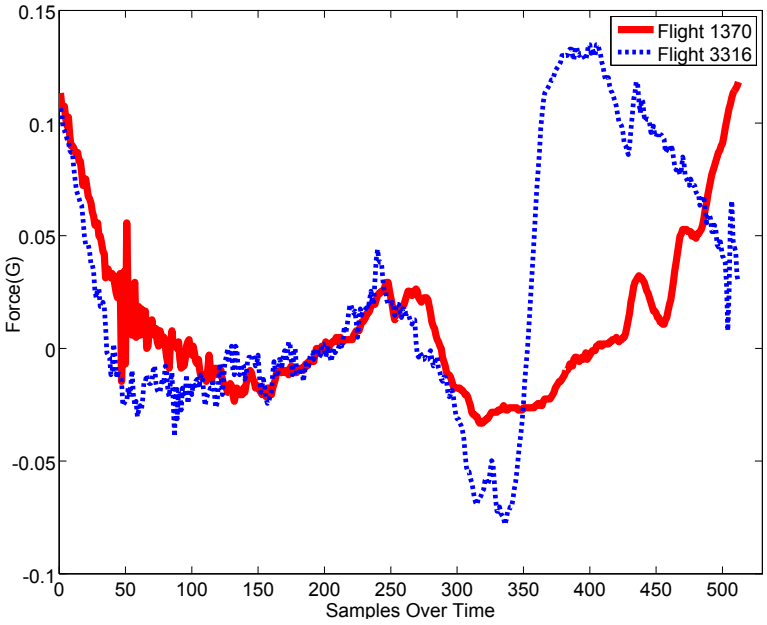

Figure 42: Flight Acceleration at Takeoff for Flight 1370

engine and they supply air to the cabin. Taking bleed from an engine is a parasitic load, meaning that this air is not available for propelling the airplane and thus reduces the power in the engines. When more engine power is needed, the bleed valves close so that more power is available. Hence it is common to see 2 out of 4 bleed valves close monetarily to make up the power.

Other significant actors, specifically those with the engines reflect an anomalous signal in general and a change that appears to correspond with the bleed valve. Figure 44 shows the temperature for engine one, but each engine has similar signals. The signals all show the engines as relatively low powered compared to the nominal. That signal also shows a drop when the bleed valve changes. The expert confirmed that this was coincidence that as the bleed valve is opened, and hot air leaves the turbines, the temperature would drop. While this is an unusual case, it had no safety implications. If the bleed valves remain closed for the entire flight or consecutive flights then it is a indication of degrading engine power. Since this is a one-off anomaly, and the expert verified that the bleed valve does not remain closed, this anomaly did not represent a safety issue.

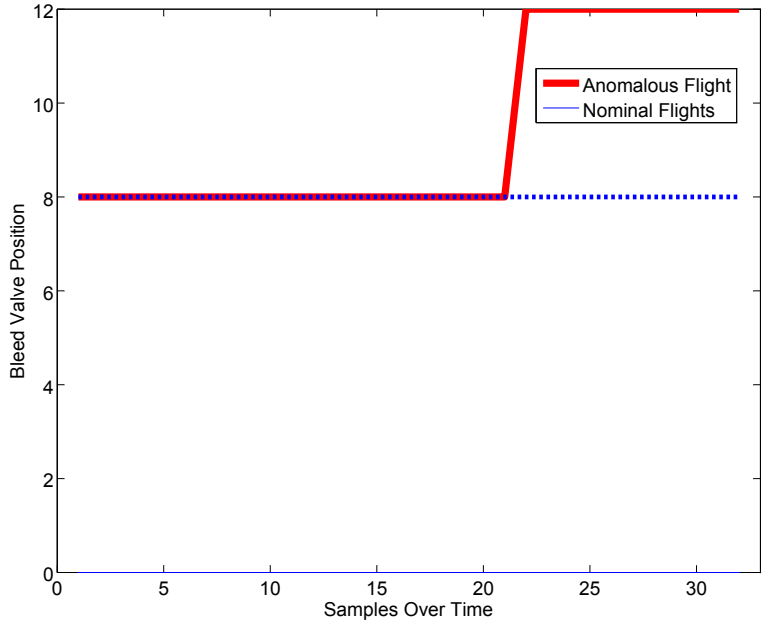

Figure 43: Bleed Valve Position at Takeoff for Flight 4893

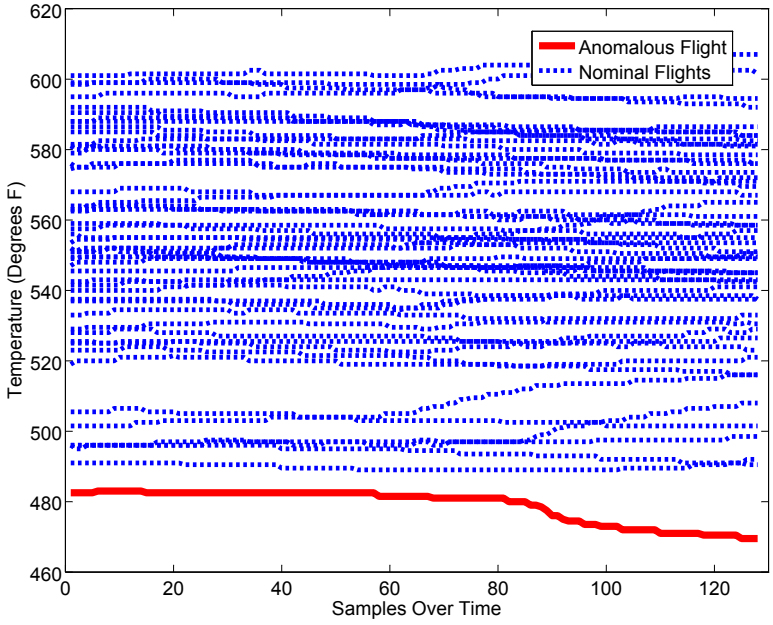

Figure 44: Temperature for Engine One at Takeoff for Flight 4893

The last flight we explore is the one that involves a significant actor for the sensor that detects when the automatic throttle computer has been engaged. Flight 222 contains an interesting set of significant actors. First, Figure 45 shows the automatic throttle sensor. The nominal data shows that this auto throttle is always engaged at takeoff. We can witness possible effects with significant actors from the engines such as Fan Speed. Figure 46 shows a plot of the fan speed for engine one during this flight. The engine is underpowered and the signal is definitely a different shape than those in the nominal sample as it rises twice during the takeoff. Lastly, the radio altitude was also identified as a significant actor. The plot for radio altitude in Figure 47 shows a flight that remains on a much shallower climb than the ones in the nominal set.

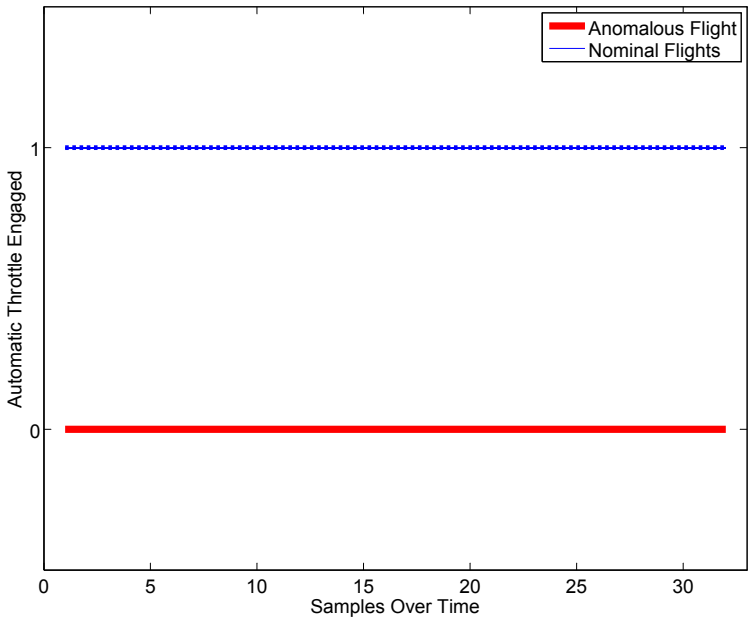

Figure 45: Automatic Thrust Engaged at Takeoff for Flight 222

After examination, the expert said it is rather atypical for a pilot not to engage the auto throttle during takeoff. Sometimes pilots may disengage the auto throttle because they feel they can handle the cross-winds better or have an unusual weight distribution on the aircraft. However, the data from the 2 engines from the significant actors indicate that the

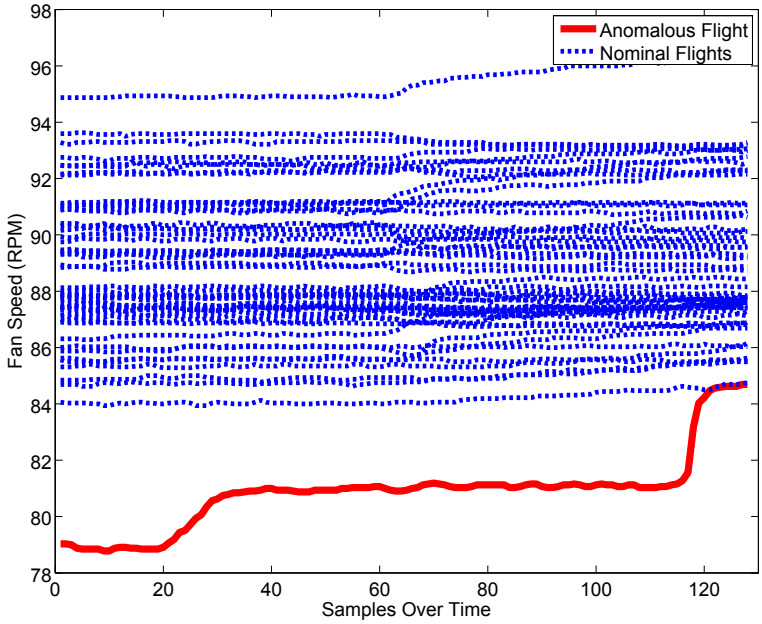

Figure 46: Fan Speed of Engine One at Takeoff for Flight 222

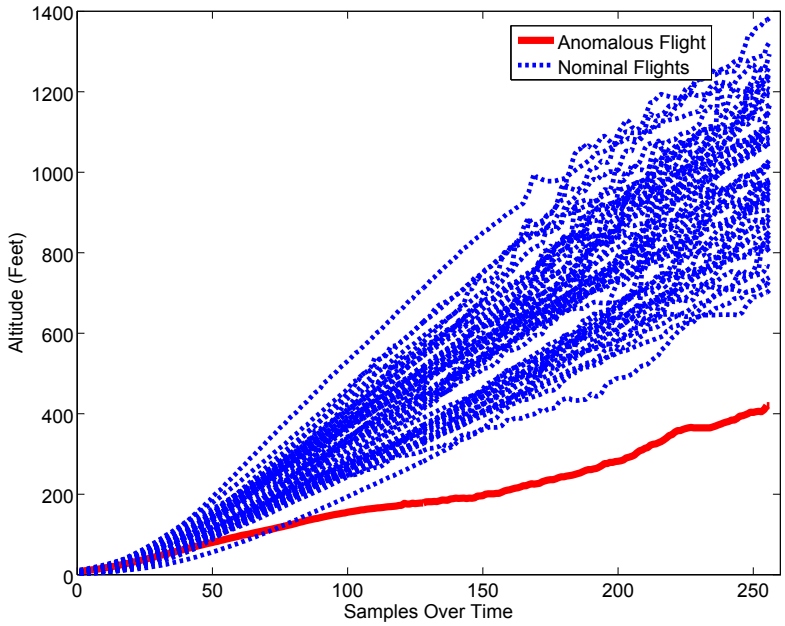

Figure 47: Radio Altitude at Takeoff for Flight 222

engines are synchronized and hence this data set may represent a change in command for the throttle. While nothing mechanical seems out of the ordinary here and the data shows the engines lining up correctly, this is an non-typical operating procedure. This anomaly clearly represents a pilot based decision.

This second cluster of anomalies contained quite a bit of variance in the significant actors, the possible causes and the impact on aviation safety. It would be difficult to explore each of these flights on a one by one basis. Our approach of using targeted projection pursuit helps induce possible places to start and groups of very similar anomalies. Together these results help cut down on the overhead an expert would be expected to contribute to characterize these anomalies.

VI.3.4.3 Comparison with CIDM/PPM on Weight on Wheels Based Data Cube

We briefly examined how the data clustered when we looked at the Weight on Wheels data with CIDM/PPM as the complexity measure. Figure 48 illustrates the dendrogram formed from the use of CIDM/PPM as the complexity measure. The rectangle on the right of the figure highlights the outlier data points. These are two clusters which are shown in more detail in Figure 49. The far right cluster (cluster 1) contains 37 flights and the left cluster (cluster 2) contains 62 flights.

Closer examination of these clusters is performed by comparing the flights found in these clusters to the anomalies found in the clustering that uses the Haar Wavelet. Cluster 1 contains only high altitude flight discovered in the two clusters we examined with the Haar Wavelet transform. It also contains every one of these flights, indicating that CIDM/PPM appears to more more likely identify their similarities and the fact that they are anomalous in the data (due to their rarity). This is contrast with the Haar Wavelet transform which found high altitude flights to be anomalous, but broke them into two groups, based on how aggressive the ascent at takeoff.

The second cluster contained 41% of the rest of the anomalies in the Haar Wavelet

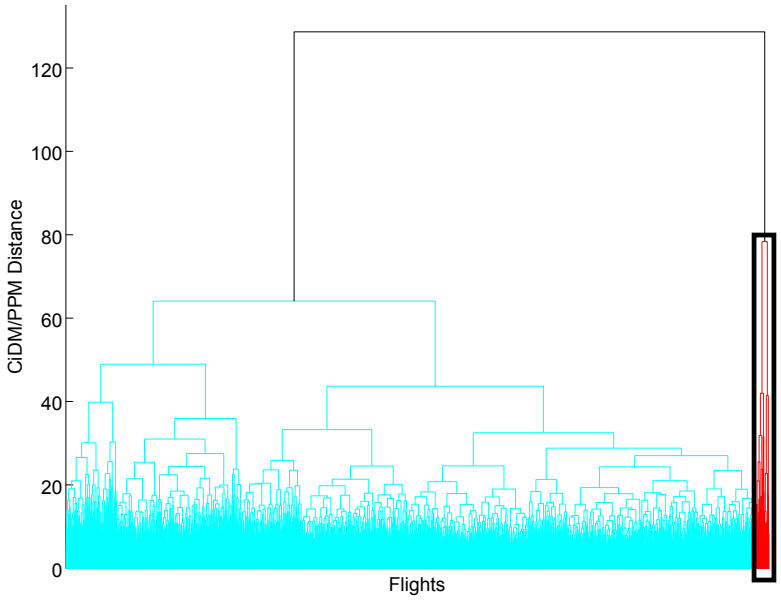

Figure 48: Full Dendrogram of the Weight on Wheels Data with CIDM/PPM

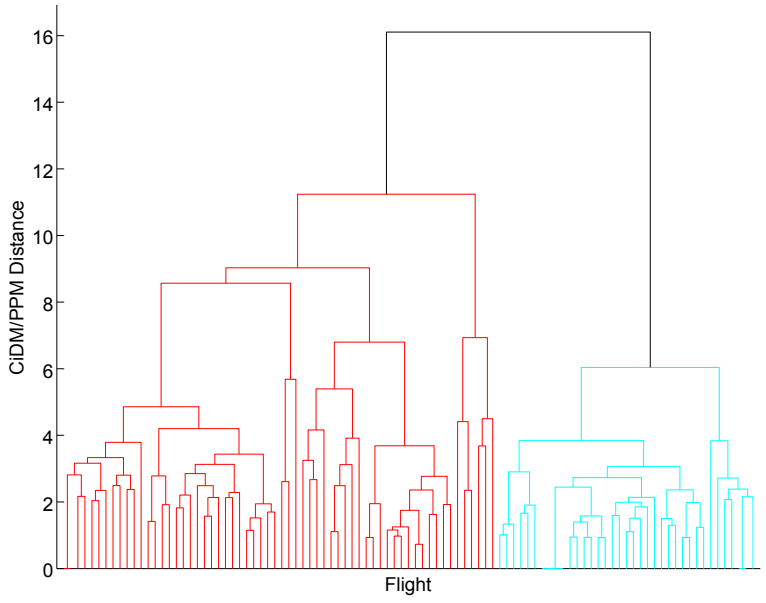

Figure 49: Enlarged Dendrogram of the Anomalous Clusters in Figure 48

based clusters. Among the missing anomalies was the near stall flight described earlier. A quick analysis showed that many of the flights in this second cluster contained issues where the flight path changes earlier in the data. The expert found this to be slightly anomalous but not an aviations safety issue.

In general, the CIDM/PPM clusters appear to be initially similar to the Haar Wavelet clusters, but the anomalies found in the Haar Wavelet clusters are more varied and contain more interesting flights for the expert to investigate. From this comparison, these results bolster the experiments in the previous chapter, showing that the Haar Wavelet transform is likely a better measure in the multivariate situations.

VI.4 Conclusion

In this chapter, we applied our knowledge of complexity measures derived from our review and experimental studies in Chapter V, and applied it to our exploratory approach for identifying and characterizing anomalies in a large multivariate signal dataset of flight segments. This approach starts from curated data, extracts an appropriate part of the data, uses the complexity measure to reduce the dimensionality and then applies hierarchical clustering to the dataset. The derived clusters can be broken into a large nominal set, and the rest into anomalies. Due to the way we reduce the data, we can utilize the same work to help characterize significant actor features that help explain the anomalies because they differ significantly from the nominal set. We also present a process for beginning the characterization process when there are more than a handful of anomalies to process. We present this in contrast to previous work, including the state of the art.

We explore our approach through the application of flight data to identify anomalies related to aviation safety. We focus on data related to aircraft takeoff, a phase that is strenuous on both the pilot and the aircraft, and takes the operating environment into account. We looked at two possible ways to extract this data for takeoffs. Examining other approached that have been applied, we found that this data was either too big and needed

added contextualization, or we found the current implementations lacking in flexibility for large unlabeled data.

Using our approach, we found that our first method for calculating takeoff was too broad and did not contextualize the takeoff appropriately. As a result, very few interesting anomalies were found. When we applied our approach using a much tighter definition of takeoff, we discovered more interesting anomalies, ranging from the environmental such as high altitude takeoffs, to ones in which the aircraft experiences changes in engine performance, to anomalies that indicate a pilot choice that would be worth investigating further. From the eight types of anomalies presented to our aircraft expert, three were flagged as very interesting for further study. This included a dead engine, an issue with a possible stall, and a pilot choice to not use the computer auto thrust. Only one type, the high altitude takeoff was considered unimportant, but it presented very clear significant actors that would allow it to be filtered in the future. In general, the expert found the method useful for identifying interesting anomalies from such a large dataset.

Through this last application of our approach we demonstrate a primary contribution of this work. Our approach is designed to handle unlabeled data, and make it easier for practitioner and expert to work with the data and isolate interesting cases for further exploration. This approach is shown to help identify possible nominal sets which would be useful for building semi-supervised models for further classification of new data. Our approach is also successful in showing how it may be helpful in the aircraft domain at isolating flights from very large unlabeled datasets that are worth exploring for possible fault causing behaviors. This approach is designed to be general, and in the next chapter we apply it to a different domain, i.e., pitcher data in professional baseball games.

CHAPTER VII

ANOMALY DETECTION OF UNLABELED PITCHER DATA FOR EVALUATION OF MECHANICS

Chapter VI presented our unsupervised exploratory approach to discovering anomalies in large segments of flight data, where most of the flight instances were nominal. However that data dealt with a number of identical aircraft, and their behaviors were defined by the aircraft state, physical laws, the manner of operation, and the environmental conditions. Anomalies in this domain correspond to signals that show different characteristics from the nominal behavior, which is derived by clustering all of the flight instances, and labeling the large groups of flights as nominal. An example, such as a pilot not using the automatic throttle to control the airspeed is detected because almost all nominal takeoffs utilize this controller.

However, our second problem domain, which involves the study and analysis of pitches thrown by a pitcher pitchers in Major League Baseball games is different, because, in this case, the pitcher's throw, once it leaves his hand, obeys the laws of physics, and is affected by environmental conditions. There are many subtle differences, however, in the way the ball leaves the pitcher's hand. A lot of these variations can be attributed to the pitcher's decision making and his mental and physical state, and all of these are much harder to analyze than an aircraft during taxiing, taking-off, cruising, or landing. Since the pitcher's decision making and mental states are truly latent, it makes exploring anomalies in baseball pitching much more challenging, primarily because there are not a finite state of well-defined physical laws that completely define the characteristics of a pitch.

Therefore, although this chapter uses the same exploratory approach for the same set of research challenges for complex systems and large data, the domain provides a new challenge from Chapter VI. This additional challenge in this chapter is to analyze baseball

pitchers' throws on a game by game basis, and try to characterize games where the pitcher's overall throws were anomalous as compared to their average pitching behaviors across the entire set of games that they pitched. Like the aircraft data, once we discover anomalous games, we analyze the details of the pitches in that game to determine what was different in their set of pitches for the game. The cumulative effect of the characteristics of a pitch includes parameters such as grip, release point, and shoulder and arm action. These are collectively referred to as a pitcher's mechanics. A number of environmental factors are carefully controlled during a game, however, the consequences to diversity in a pitcher's approach, and the his decision making can vary how one player's mechanics apply from game to game. As discussed earlier, we use our unsupervised learning approaches to classify the set of pitches the pitcher made in the game as nominal or anomalous, i.e., those that deviate significantly from the nominal. Analyzing the anomalous pitch patterns in more detail, should provide us with sufficient information for identifying good and bad patterns in the pitcher's throwing mechanics, and how they impacted his performance.

This chapter is organized as follows. Section VII.1 describes how our approach for organizing this data, and contrasts the approach with the aircraft flight system domain. Section VII.2 presents the results of the approach. We first describe the process of looking at all the data together, and the process of building a one pitcher model. We then contextualize the data into specific pitchers. From these models we examine case studies from selected pitchers and describe the anomalies found by our model. We summarize these results in Section VII.3 and briefly compare the results of using our approach with baseball data with the aviation data in the previous chapter.

VII.1 The Application of the Approach to Pitcher Data

Applying our approach follows the pitcher data involves the same steps described in Section VI.2 and illustrated in Figures 20 and 21. We first build the data cube described

in Section III.3.1. Using a complexity measure, we reduce the data cube to a set of dissimilarity matrices for each feature. This set is then reduced to a composite dissimilarity matrix and we build a hierarchical cluster and search for anomalous groups. Our approach then applies feature selection to help characterize these anomalies for further investigation and modeling. This application requires some modification in the implementation of these steps. We discuss the changes in data curation, contextualization, and how to interpret the output during characterization.

VII.1.1 Data Curation and Contextualization

The pitcher data for this study was put into a structured and interpretable form by Harry Pavlidis at Pitch Info LLC and made available to others in the form of a SQL database. Each record in the database is a pitch thrown since 2007. The structure used to transform the data from the database into the data cube is defined in Section III.3.3. Similar to focusing on the takeoff for the aviation data, to further curate the baseball data to make comparisons between pitcher games more equitable. We focus on games starting from the year 2009 where a pitcher throws 100 or more pitches for that game and we concentrate on pitchers that have at least 75 such games from 2009-2012. The reason for 100 pitches per game, is that all of these pitchers pitched for a sufficiently long time in the game for fatigue to set in on their bodies. The choice of 75 games was somewhat arbitrary, but chose to ensure that each of the pitchers had played a sufficient number of such games for the period of the study. Further, the choice of high number of pitches, implies that the pitcher was doing reasonably well, otherwise, he would have been replaced by a reliever earlier in the game due to poor performance. This selection results in our selecting 20 pitchers, and a total of 1818 instances of games that they played. These 1818 instances make up one dimension of the data cube. Table 32 presents the list of the 20 pitchers selected, whether they are right or left handed, and their top three pitches by overall usage.

Table 32: Pitchers Used in Data Cube

Name	Handedness	Pitch One	Pitch Two	Pitch Three
Ubaldo Jimenez	Right	Four Seam FB	Sinker Ball	Slider
Jered Weaver	Right	Four Seam FB	Sinker Ball	Slider
C.C. Sabathia	Left	Four Seam FB	Slider	Change Up
Roy Halladay	Right	Cut Fastball	Sinker Ball	Curve Ball
Jon Lester	Left	Four Seam FB	Cut Fastball	Curve Ball
Zack Greinke	Right	Four Seam FB	Slider	Sinker Ball
Clayton Kershaw	Left	Four Seam FB	Slider	Curve Ball
Matt Cain	Right	Four Seam FB	Slider	Change Up
Cliff Lee	Left	Sinker Ball	Four Seam FB	Cut Fastball
Felix Hernandez	Right	Sinker Ball	Four Seam FB	Change Up
Jeremy Guthrie	Right	Sinker Ball	Four Seam FB	Slider
Justin Verlander	Right	Four Seam FB	Curve Ball	Change Up
Yovani Gallardo	Right	Four Seam FB	Curve Ball	Slider
Max Scherzer	Right	Four Seam FB	Change Up	Slider
Dan Haren	Right	Sinker Ball	Cut Fastball	Curve Ball
James Shields	Right	Four Seam FB	Change Up	Cut Fastball
Tim Lincecum	Right	Four Seam FB	Sinker Ball	Change Up
Cole Hamels	Left	Four Seam FB	Change Up	Curve Ball
David Price	Left	Four Seam FB	Sinker Ball	Curve Ball
C.J. Wilson	Left	Four Seam FB	Sinker Ball	Cut Fastball

Like the aircraft domain, the anomaly detection task remains the same, but in this analysis there are two areas in which we could discover anomalies. First, we are looking for the games where the pitcher still went deep into the game but was not as effective as he was on average, i.e., we look for subpar performances. Second, we look for the opposite, i.e., games where he performed better than his average performance, and then analyze what changes in the mechanics resulted in the superior performance.

We employ the Haar Wavelet transform based distance to reduce the data cube, therefore, we need the signal lengths to be powers of two and we need an equal number of coefficients between pairs of signals. Unlike the aircraft flight system domain, there does not exist a contexualization of the baseball data that allows us to easily compare equal sized signals from the data without removing some of the signal. Therefore, to prevent loss of information by truncation of signal lengths, we padded each signal with zeros to next nearest power of two. We modify our implementation to make sure that when we compare two signals, if they are not already the same size, we further pad the smaller of the two signals with zeros to match the larger signal. The use of padding with zeros to help with signals that are not the right length has been advocated for a number of applications [138] and used by researchers to help explore the power spectrum [155] and compression [113]. In our case, when the two sequences are quite different in original length, this padding does not interfere with identifying these sequences as different. The consequence is that the approach can detect anomalies where one signal is not necessarily shaped different, but much smaller than normal, or missing altogether. In the context of baseball, this would mean that might detect games where a pitcher rarely uses one of their frequently used pitches. Given the contextualization of the data for a specific set of pitchers, finding pitcher-games where a pitcher does not use a normal pitch would be an interesting anomaly.

Given this data we explore two different contextualizations. We first examine the entire data cube. Similar to the aircraft flight system domain, we treat every pitcher-game instance as originating from the same type of generator. This use of the data cube is meant to provide

a baseline for how this human data responds to clustering. The second contextualization is to focus the approach on a specific pitcher. In this case, we remove any possible variation in terms of mechanics across all pitchers and focus on looking at finding a nominal set of games for that given pitcher. This would provide a nominal model for those mechanics and isolate anomalies where the mechanics were different. From this contextualization, we can focus on specific changes, rather than trends across all pitchers.

VII.1.2 Characterization of Anomalies in Pitcher-Games

The characterization of anomalies in this data set follows the same general approach, but to visualize the significant actors for this data and domain is slightly different. Since we deal with significant actors that indicate the lack of a signal when one is normally expected, we merely mark this for the expert. When the signal is present but different we rely on three types of plots to demonstrate the differences. We still utilize plots of several nominal signals against the anomalous signal. This is effective when we want to demonstrate that the pitcher's speed or spin on the baseball is general different (higher or lower) than normally expected. This provides a sample of the range for a pitcher as a game progresses. The second visualization is to produce a mean signal and illustrate a one sigma range around the signal. The plots of the nominal games can be so spread that they make it hard for an expert to track the nominal trends and see why the anomalous signal was different. A mean signal present the expert with a general trend and the sigma range shows a bit about where the pitchers normally fall during that trend. When the pitcher's mechanics are functioning at a level where he is not outside of his normal range, but is trending differently than expected, this plot can help characterize those instances. Lastly, our final plot is one that eschews the temporal sequencing in order to show a general change for the game. Sensors such as the starting location and ending location benefit from seeing the nominal locations for games, and comparing these to the anomalous set. This is especially true when the location is different but remains constant over time, making the signal less interesting. Visualizing

these sensors in the context of the game by displaying the Cartesian planes from which they are measured gives an expert the clearest idea of where the differences occurred for that game.

Lastly, this domain benefits from wealth of ancillary information collected about the game. These include news articles, as well as the game statistics for the pitcher. Unlike the aircraft flight system domain where secondary information such as pilot reports and mechanic notes may not exist or be possible to retrieve due to privacy concerns, the sport domain is about providing as much similar information to the fans as possible. For each anomaly we can examine the significant actors and then see if this can be corroborated with information provided by the players and coaches as well as scouts. In our results we utilize this information to provide context as well as to help explain the information we are receiving about these anomalies.

VII.2 Results and Case Studies with Pitcher Data

As described in our approach to this data, we utilize two contextualizations. We first describe the results of exploring the entire data cube of 1818 instances. We look for anomalous clusters and attempt to characterize the pitchers in that group. We then explore data cubes made up of a single pitcher's games. We explore three pitcher in particular. First we examine Roy Halladay, currently with the Philadelphia Phillies, and Tim Lincecum, currently with the San Francisco Giants. Both pitchers have won Cy Young awards for being the best pitcher during a season and both have won during the span of the data we currently have in the data cube. The third pitcher is Jon Lester, currently with the Boston Red Sox. Jon Lester has been a relatively successful starting pitcher, but over the last year has dropped in performance. All three pitchers provide an interesting exploration considering they all use slightly varying pitch types, and have very different mechanics.

VII.2.1 Exploring All Pitcher Data

The entire data set was reduced with the Haar Wavelet transform and distance, and then clustered. The results of the dendrogram and initial clustering are illustrated in Figure 50. Similar to previous applications of this approach, we find a very large cluster and then another smaller cluster as indicated by the rectangle. A quick analysis of this cluster showed that the data is comprised of only two pitchers, Roy Halladay and Dan Haren. Upon closer examination, the sub-tree was split into homogenous groups of the instances for these two pitchers. This would indicate that at the very least the mechanics for these two pitchers were different enough from the other 18 pitchers, but similar enough to be clustered near one another, but still separable.

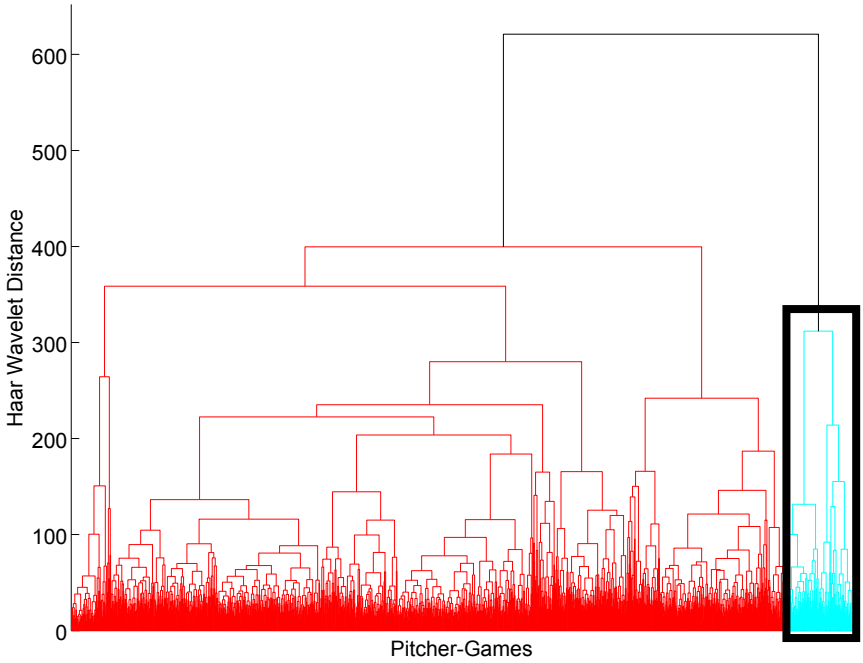

Figure 50: Dendrogram of All Pitchers with Anomaly in Rectangle

Further examination of these two pitchers from Table 32 indicates why they might be clustered. Roy Halladay and Dan Haren both thrown a rather unique set of three pitches. Nowhere to be found in their primary repertoire is the Four Seam fastball. Instead they

use another type of fastball, the cutter as the replacement for this pitch. Also of note, these two pitchers are right handed, and a quick look at their height and weight shows they have comparable body types. Based on this information, we might expect that they would provide a similar profile in terms of mechanics. That being said, their usage is moderately different with Dan Haren relying on his sinking fastball and Roy Halladay using his cut fastball as the primary pitch.

This clustering indicates that the individual mechanics and pitch repertoire of a pitcher are fairly dominant characteristics that determine the clustering results. This conclusion would line up with the expectation that even though each pitcher is over five feet and eleven inches, and they are all successful pitchers, they are a diverse group when it comes to measuring their abilities. Unlike the aircraft flight system domain where the aircraft is expected to be identical, these instances have repeatable variance that can separate out when grouped together.

VII.2.2 Exploring Single Pitcher Data

The other contextualization is clustering individual pitchers, an approach which would address the issues when clustering all the pitchers together. In this case, we are attempting to contextualize the mechanics and pitch repertoire that each pitcher brings, and hopefully build a group of nominal games where these attributes of the pitcher are relatively similar. Our goal is to then identify the anomalous games, and compare them to the expected mechanics and repertoire so we can see how a pitcher may have changed and the consequences of those adjustments. We examine three pitchers in detail, Roy Hallday, Tim Lincecum, and Jon Lester. As Table 32 shows, one is a right hander, and the other two are left handers. Their mechanics are also different starting from their size as Lincecum is only five feet and eleven inches tall, whereas the other two are over six and a half feet tall. The pitch repertoires for each are also diverse, providing breadth in our investigation.

VII.2.2.1 Roy Halladay

Roy Halladay is one of the two pitchers in our group that does not throw a normal four seam fastball, instead he prefers to throw a cut fastball which has more movement. By movement for this pitch and pitcher, we mean it has a higher spin rate, and when thrown, the ball moves to the left of the projected straight line. His other pitchers also involve a spin, including his fourth most thrown pitch, a split finger fastball, which is thrown slower than a cut fastball, and has spin on a different axis. Together these two pitches form a pair which when used strategically, are able to fool a batter into swinging at a pitch and in the case of contact, result in a weakly hit ball.

Using a data cube made only from the games where Roy Halladay pitched, we examined a total of 91 games. After dimensionality reduction and clustering, we can see a cluster of 10 games in the anomalous set shown in Figure 51. Similar to the case of the first cluster of anomalies in Chapter VI.3.4.1, we are able to rank the significant actors and look at each anomaly by itself. First, we lined up these anomalies to look for any temporal patterns in their occurence. Of the 10 anomalies, 7 of the games occurred in consecutive starts for Halladay. These games occur at the beginning of the 2011 season. Among their characteristics was the fact that Halladay did not throw his change up. While not one of his primary three pitches, the fact that he did not throw it all was part of this anomaly. Another pitch that is listed and shown to be thrown very infrequently in these games is his curveball, one of his top three pitches. Lastly, a pitch that shows up as thrown with a expected frequency but is different mechanically is his split finger fastball. Taken together, this would indicate that as Halladay was warming up for the season in his first games, he was not using, lightly using, or changing the mechanics of pitches in his normal selection.

When we examined each of the anomalous games in the context of the statistical results of how Roy Halladay pitched, we found two that were particularly interesting. We found a dominant game in his series of spring starts, and we found a poor start that lasted 100 pitches later in the season.

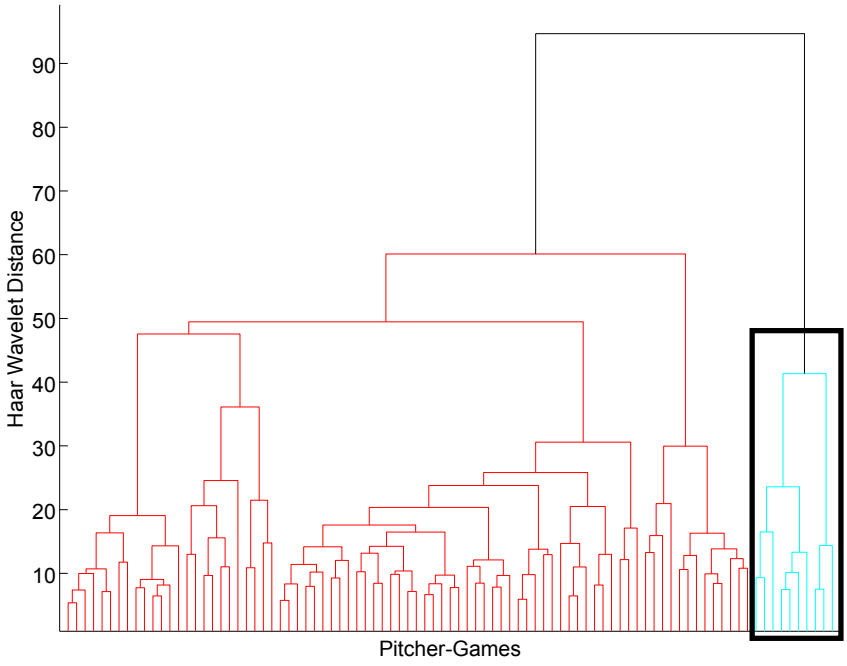

Figure 51: Dendrogram of Roy Halladay Games with Anomaly in Rectangle

The first anomaly is a complete game against the New York Mets where he allows a single run, 7 hits, and struck out 8 batters. The news reports for this start consider it a dominant performance [131]. When we examine the significant actors, we first notice the fact that he did not throw a single change up. The telling significant actors however are the start and ending speed of his split finger fastball as well as the spin rate of the same pitch. Figures 52 and 53 show the ending speed and spin rate, respectively. The ending speed is plotted against his normal trend. From this graph, one can notice that his typical trend is to lose speed on his pitches as the game goes on, but in this case, his ending speed starts up slower then steadily improves over time before dropping again. Couple this with spin rate in Figure 53, also plotted against the mean signal. Here the spin rates are much higher. Taken together, it seems that the pitches are coming at the batter faster as the game goes on, and with more spin, they are moving more than expected. With this sort of movement, and since Halladay isn't throwing one of his pitches, he appears to find a strategy for inducing outs against the Mets. The article goes on to say, that Halladay did not feel like he was

pitching his best that day, but he was being aggressive. The significant actors for the split finger fastball would help corroborate this statement.

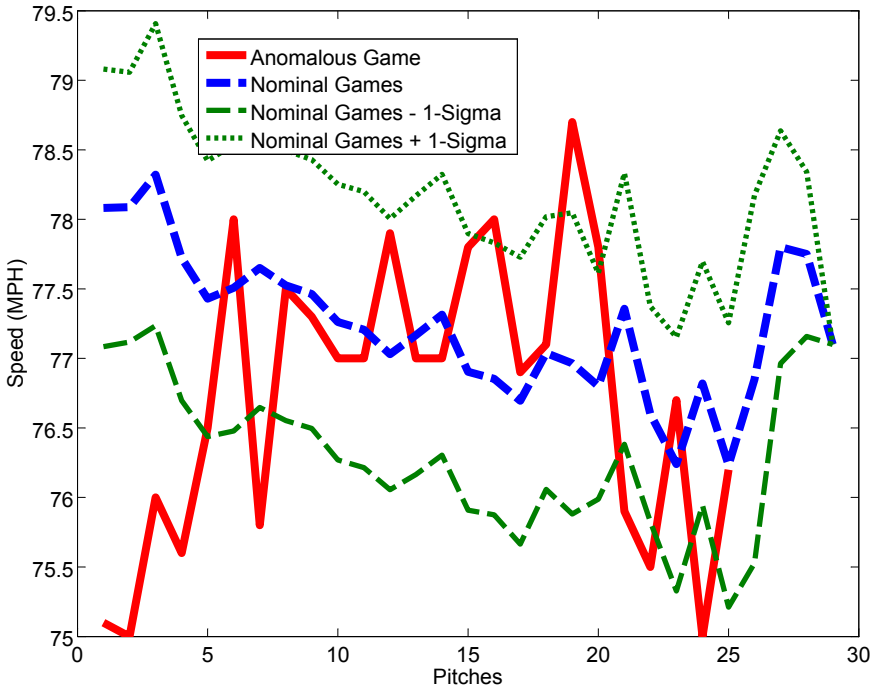

Figure 52: Ending Speed of Roy Halladay's Split Finger Fastball Against the Mets

The second game we analyze was not as successful. Although Halladay lasted over 100 pitches, he gave up more runs and was less effective [130] than at any point in the season. The significant actors for this instances include some of the same sensors as the first anomaly, as well as new significant actors. Figures 54 and 55 show the spin rate of the split finger and the speed of the curve ball. Unlike the dominant game, here the spin rate, plotted against the mean signal, is much lower. With that little spin, there was also likely less movement, meaning that it was just a slow pitch that would be easier for the batter to make solid contact. This significant actor also shows that he did not throw as many of these pitches as he would normally. This is likely due to the fact that he was ineffective with the pitch. The results are similar for his curveball, where Figure 55 shows that his starting speed for his curveball was also lower than average. With both pitches ineffective, he was

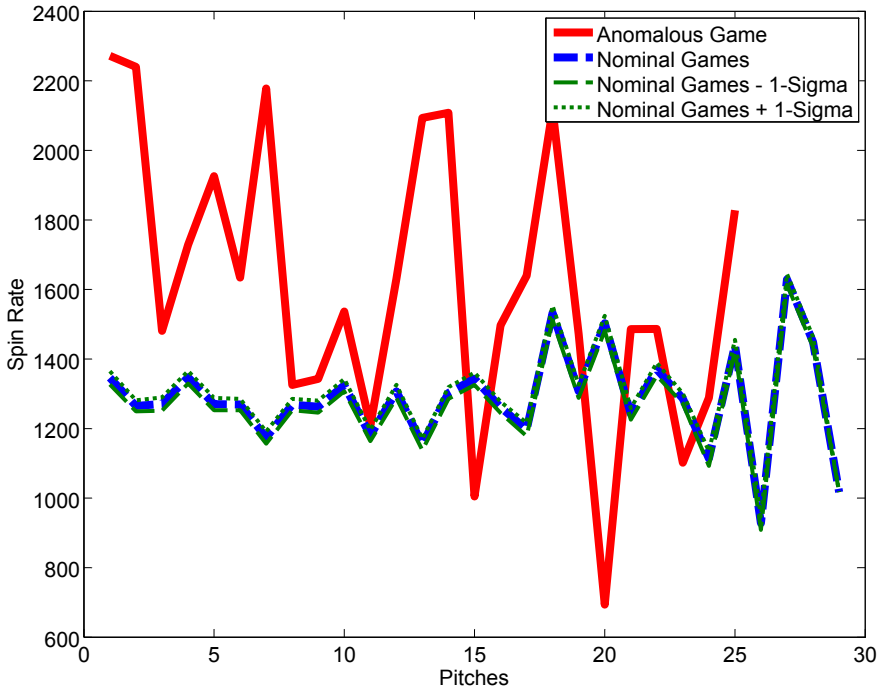

Figure 53: Spin Rate of Roy Halladay's Split Finger Fastball Against the Mets

relying on smaller repertoire. Overall, this game was a struggle because he was unable to get the typical movement on one of his pitches, and another was not at the normal velocity.

Both games show that Halladay's pitching results could hinge on his split finger fastball. A pitch like the split finger that relies on a lot of spin to generate movement and confuse a hitter is certainly going to help generate a lot of strike outs. When it isn't working, it would be more likely to cause a pitcher to have to rely on a smaller set of pitches to survive a game. Information generated for these type of games would be useful for two sets of people. First, for the team that is currently employing the pitcher, these results may help identify before his next start, what the problem was, and see if it can be corrected through extra practice. The second group is opposing teams, who could use this information, such as the lack of change ups being used in the spring, to scout the pitcher, and help their own hitters narrow down what to look for when the hit against the pitcher. These two applications are enhanced by detecting changes in the pitcher's normal approach, and being able to characterize them given the normal mechanics.

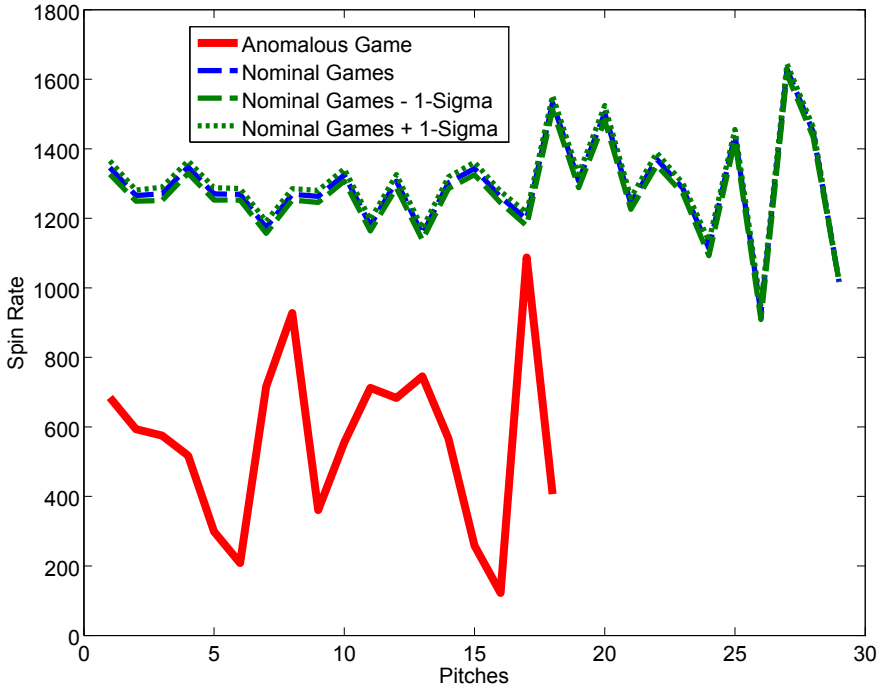

Figure 54: Spin Rate of Roy Halladay's Split Finger Fastball Against the Rockies

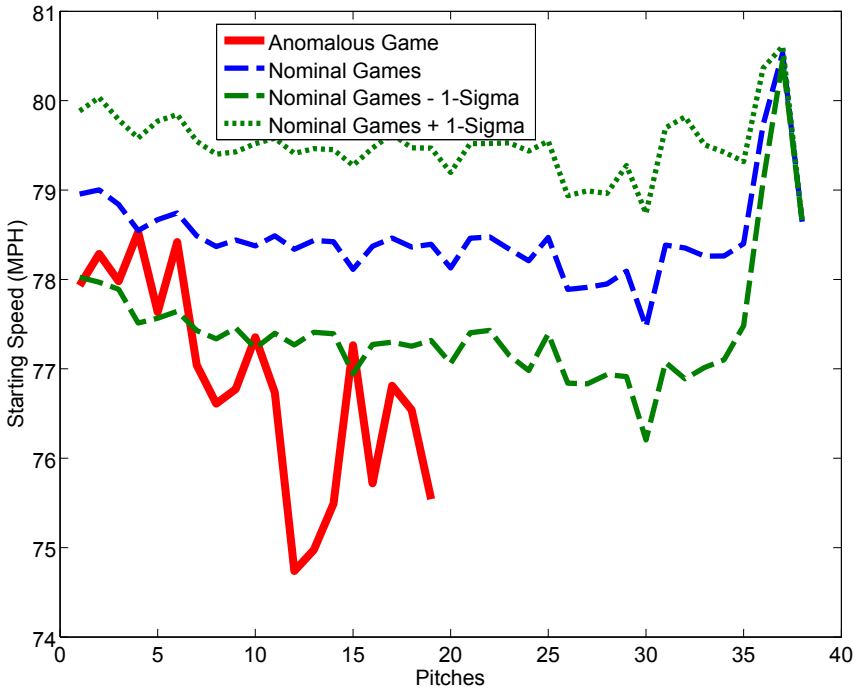

Figure 55: Starting Speed of Roy Halladay's Curveball Against the Rockies

VII.2.2.2 Tim Lincecum

The next pitcher we examine is Tim Lincecum. In contrast to Roy Halladay's mechanics, Lincecum is a different type of pitcher. Shorter than Hallday by 7 inches and also left handed, Lincecum has the nickname of "The Freak" due to a unique pitch delivery for his frame. He has also been very successful, winning several season awards for his performance.

Contextualizing the data cube for only the games that Lincecum pitched reduced the data cube to 88 instances. After reduction, and clustering, the dendrogram shown in Figure 56 helped identify one large cluster and one small cluster containing 8 anomalies. Same as with Halladay, we looked at each of these anomalies one on one, since there were so few. Looking at them temporally, only two of the games were sequential. Similar to Halladay, both starts were from the beginning of a season, and in this case, the season was 2009. In both games, it appears that the reason for the anomaly was that the pitcher did not utilize his slider. This makes sense, because sliders are one of the more physically straining pitch types, thus the need to slowly work into throwing that pitch during the season so as to not risk injury. The remaining 6 anomalies, however, demonstrate how this approach on the data is good at identifying anomalies which are due to performance above the norm for the pitcher. We present one of the anomalies as a case study of the 6.

This anomaly was a complete game shutout, where the pitcher gave up 3 hits and had 6 strike outs in a win over the Oakland Athletics. It was such a great game, that advanced stats scored it as one of the three best games for Lincecum up to that point in his career [132]. Complete games are commendable by themselves, but ones where the pitcher has a shutout are considered exceptional. Examining the significant actors for this game, we find an interesting set. First, in Figure 57, we show the starting speed of Lincecum's four seam fastball over the course of the game. A four seam fastball is the most common fastball in Major League Baseball. It is often referred to as a "rising" fastball due to the fact that the spin is placed on the ball to give it the illusion that it is moving upwards towards the

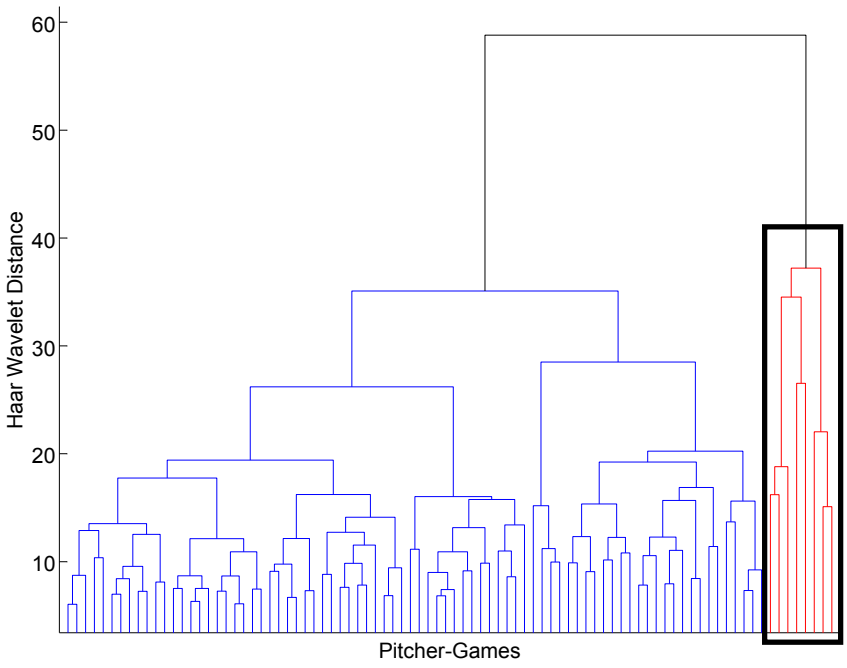

Figure 56: Dendrogram of Tim Lincecum Games with the Anomalous Cluster in a Rectangle

batter. In this plot we can see that the speed is not exceptional, but rather it is the lack of a pattern of degradation in the speed over time, as shown by the mean signal. Lincecum was able to maintain his top velocity over the course of the game, and was able to vary it when necessary. One of the many subtle levels of strategy is a pitcher's ability to vary their own speed and balance their control over the location of the pitch with overwhelming a batter with higher than expected velocity. This is backed up by the spin rate on his fastball which also maintained a consistency shown in Figure 58. In the case of his four seam fastball, Lincecum was able to produce this kind of variation over the course of the game, and even threw it harder at the end. The final significant actor to examine is in Figure 59, where we show the release location of the four seam fastball. The release point is an x and y coordinate based significant actor. The plot shows the anomalous pitches against a random sample from the nominal games. This is visualized as facing the pitcher. The origin refers to a spot in the middle of the pitcher's mound and a ball thrown directly in front of the pitcher. The x-axis shows the release point in relation to the middle of the pitcher's mound

and the y-axis refers to the release from the height of the pitcher (these values have been normalized for the given pitcher and the height of the mound). The originating significant actor in this case is the x-axis. In general we see that his release point varies on this axis, moving at times further from the mound and at other time closer. Also of note, the release point is consistently low, especially compared to the spread shown in the normal data. This lower release point may have helped him maintain his velocity over the course of the game. The varying nature of the x-axis may also have helped confuse the batters about the type of pitch being thrown as well as obscuring the pitchers intended velocity.

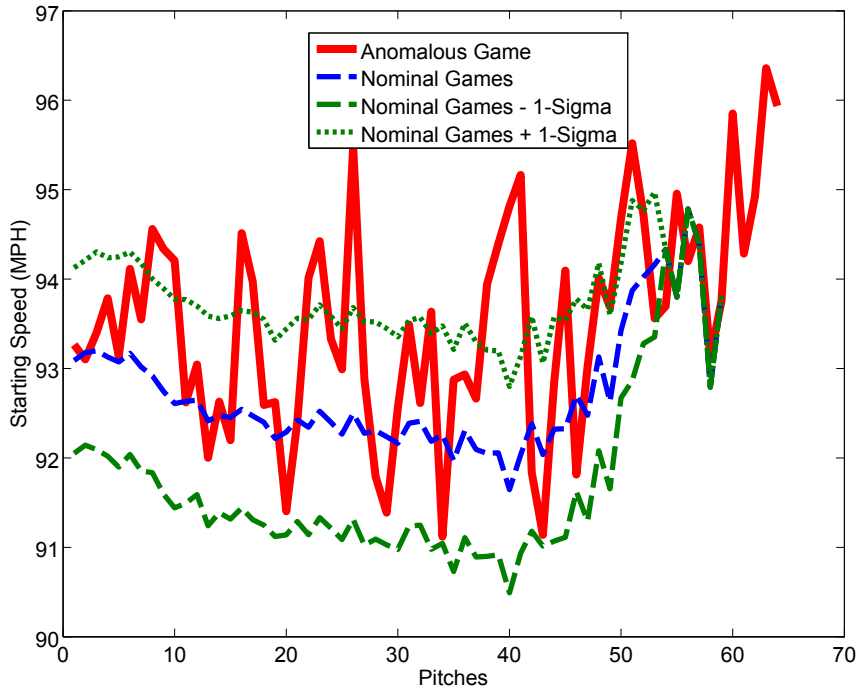

Figure 57: Starting Speed of Tim Lincecum's Four Seam Fastball Against the Athletics

Similar to Halladay, we find that for Tim Lincecum we are able to identify some of the anomalies from his lack of throwing certain pitches. Our ability to catch when a pitcher is not throwing a pitch is important to potentially identifying whether there is an injury explanation, or if it's conditioning the throwing arm for a long season, especially when

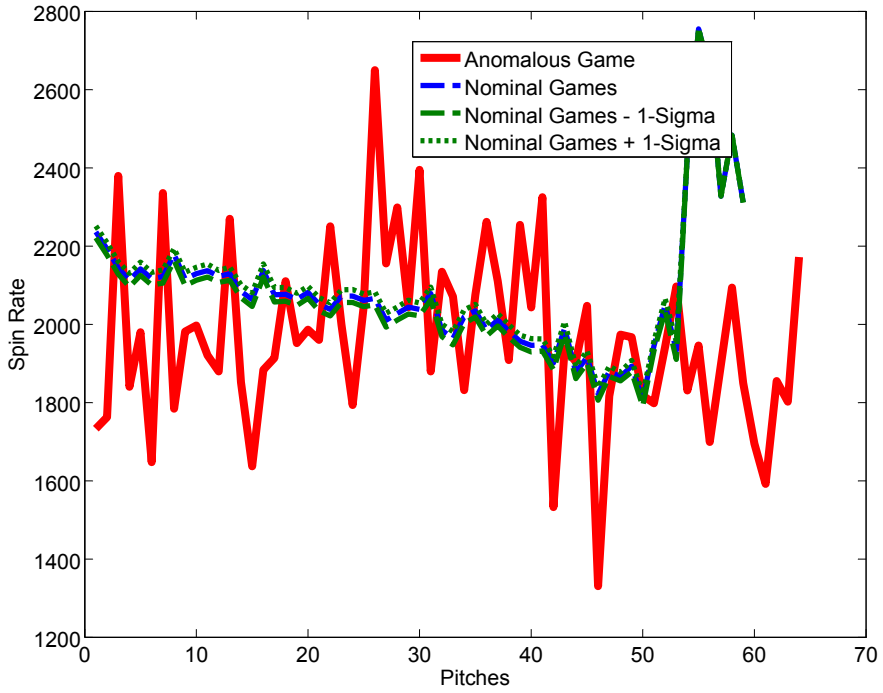

Figure 58: Spin Rate of Tim Lincecum's Four Seam Fastball Against the Athletics

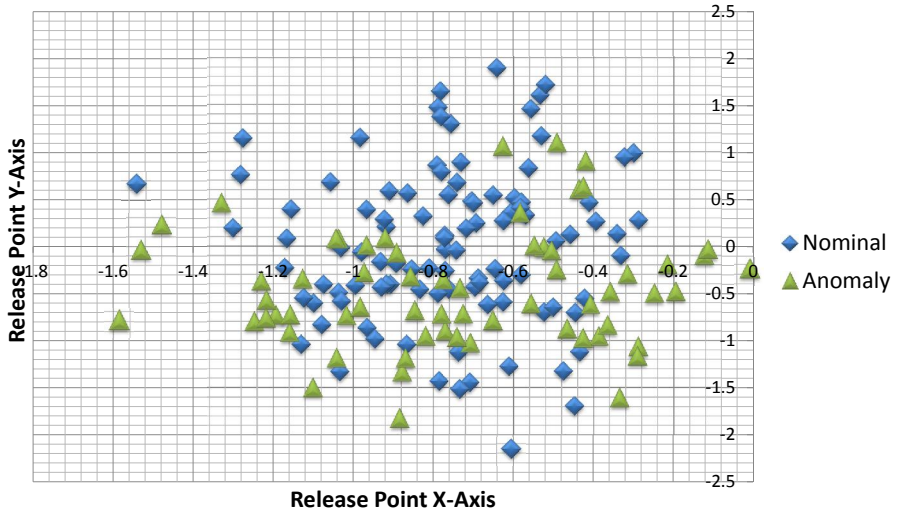

Figure 59: Release point of Tim Lincecum's Four Seam Fastball Against the Athletics Compared to Nominal Games

the pitch like a slider can put a lot of stress on the arm. Lastly, this approach helps catch what makes a start special, such as the shut out we examined for Lincecum. In contrast to Halladay, it may not be a spike in the velocity during the game, but rather maintaining the performance and varying the release point of the pitch that is effective.

VII.2.2.3 Jon Lester

The last two pitchers presented two very different approaches to pitching, from handedness, to size, to the overall repertoire. The last pitcher we examine, Jon Lester, is a cross between the two pitchers. Also left handed like Lincecum, the body type of Jon Lester is similar to Roy Halladay. Lester's pitch selection is a cross between the two, with an overlap of two pitches out of the top three for the other pitchers. This provides an interesting final study, as we examine a great start, and a really poor start where the success and failure of the mechanics are clearly understood and even recognized by coaches before they even examined the video tape of the game.

Contextualization of Jon Lester's games from the data produces a data cube with 91 instances. Figure 60 shows the dendrogram after performing the dimensionality reduction on the data. Unlike the two previous examples, there is one large cluster and two smaller anomalous looking groups of games. From right to left in Figure 60 we refer to these smaller clusters as anomaly cluster 1 with three games and anomaly cluster 2 with 8 games. This provides an interesting chance to look at these clusters and to look across them.

The first cluster contains three games. Their unifying theme when examining the significant actors is that these are starts where Lester did not throw a sinker ball. While not in his top 3 pitches, Lester does use the pitch a moderate amount during games and these three did not include a single instance. Of these three, two were excellent starts, one during 2010, and another in 2011. The final game in the cluster was from 2012 and was not a performance on the same level as the other two games in the cluster. In fact, Lester was mediocre, but it was considered an improvement from his previous starts[29].

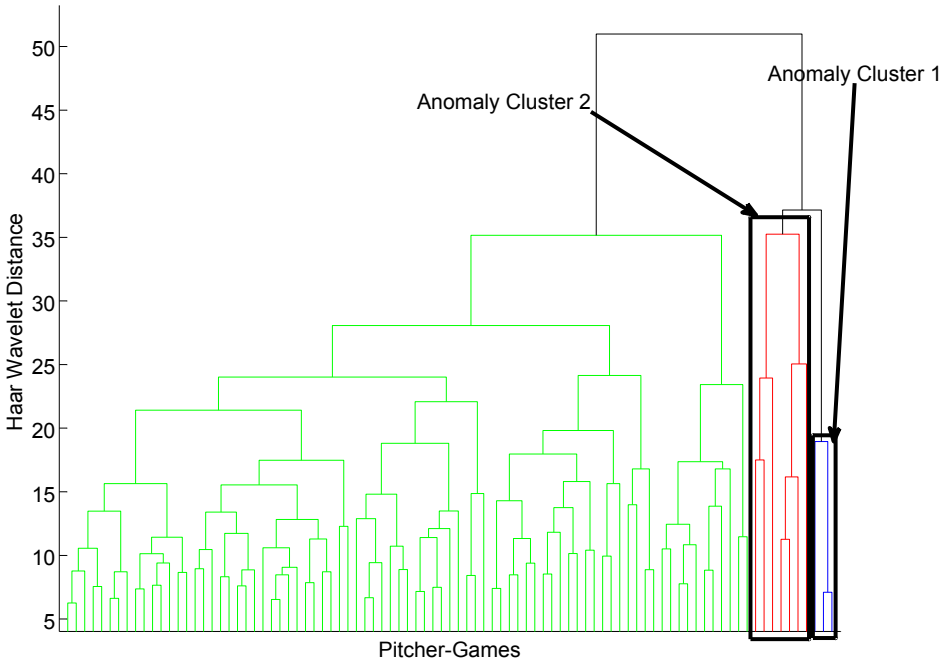

Figure 60: Dendrogram of Jon Lester Games with Anomalous Clusters in Rectangles

We focus on the two top games, which include a complete game against the Minnesota Twins [116] and a strong performance against the Kansas City Royals [129]. In both games, the significant actors are demonstrative about what elevated the performances. Figure 61 shows the starting speed for Lester's four seam fastball against the Twins. Plotted against the mean, it shows that Lester really elevated his velocity against the Twins and attempted to overwhelm them with the speed. This is coupled with his release point shown in Figure 62. The release point is not vastly different in terms of the Y-Axis, but Lester is more consistent in releasing a bit closer to the middle of the mound. This results in more of an over head motion which would allow him to increase the overall velocity of this pitches. The consistency in the location of the x-axis means that he was likely throwing with more command.

The game against the Kansas City Royals only has two significant actors: the spin rates of the four seam fastball and the cut fastball. Figure 63 and 64 show these rates for the four seam and cut fastballs respectively. In both cases, plotted against the mean value,

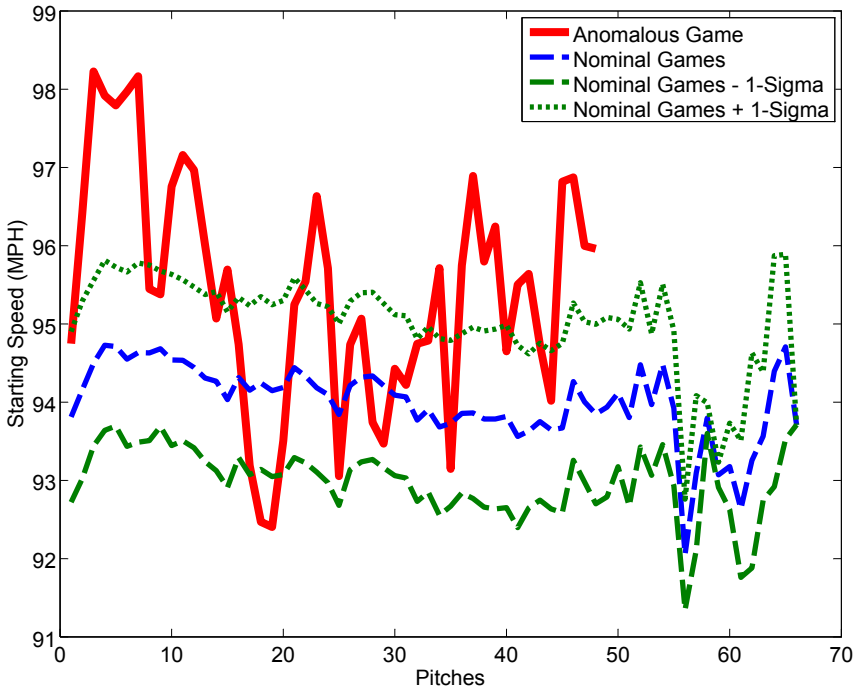

Figure 61: Starting Speed of Jon Lester's Four Seam Fastball Against the Twins

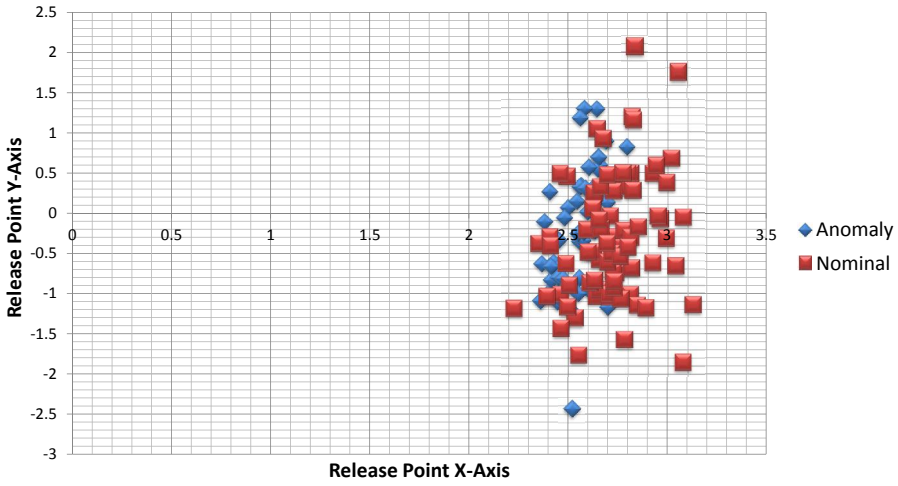

Figure 62: Release Point of Jon Lester's Four Seam Fastball Against the Twins compared to Nominal Set

we see that these spin rates are higher the the nominal signal and with greater variance. The spin rates for both pitches are still sustained at a higher level than normal. Since no other significant actors are selected, the pitches were likely at the normal velocity and release point, but contained better movement and thus were sharper in terms of location and more effective. In this case, we see that better mechanics does not necessarily mean an improvement in terms of speed, but rather, it may indicate that the grip has improved, producing better spin and thus improving the deceptive nature of the pitch's movement.

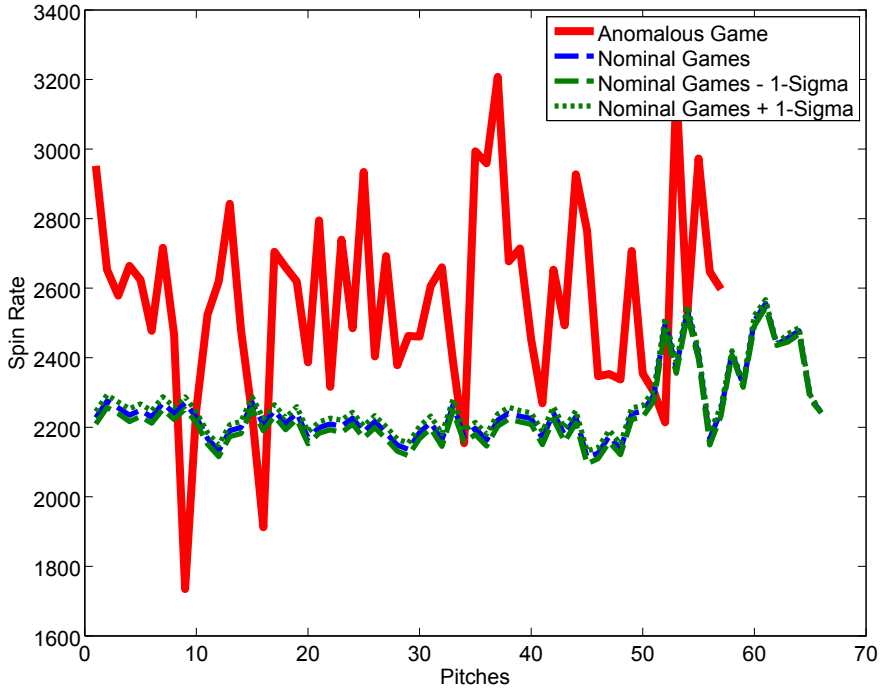

Figure 63: Spin Rate of Jon Lester's Four Seam Fastball Against the Royals

The second cluster contains 8 games. With the exception of one game where it listed the significant actor for no change ups thrown, the rest of the games were not flagged for lacking in an expected pitch type. Instead these games seemed to indicate a fair amount of variability in the pitching of Jon Lester. We focus on one game in this cluster which contrasts with the two great games examined above and contains expert testimony that corroborates the significant actors. The game we chose from this cluster was a year to the

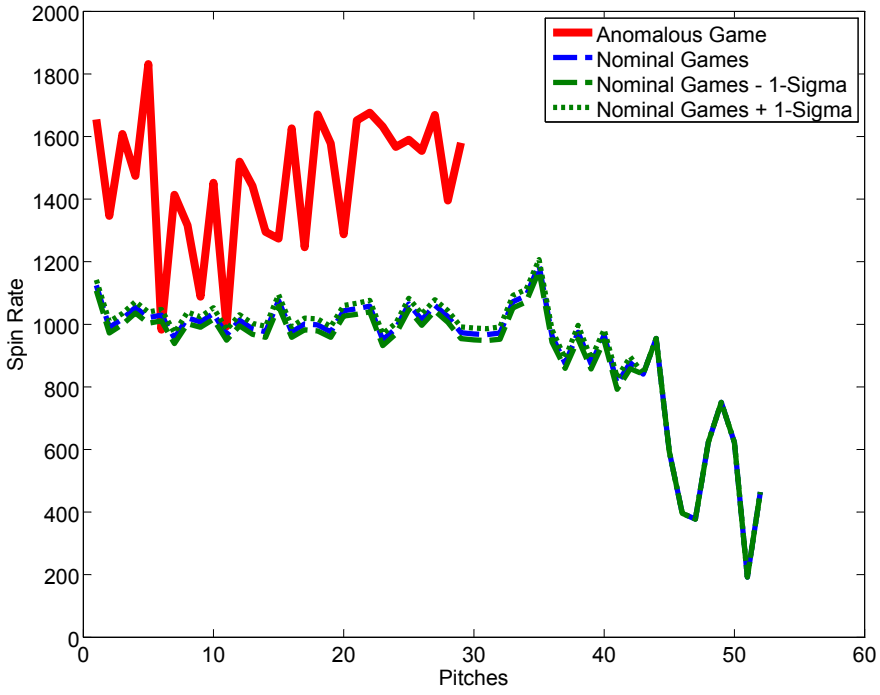

Figure 64: Spin Rate of Jon Lester's Cut Fastball Against the Royals

day after the complete game against the Twins and was a tough outing for Lester against the Chicago Cubs [160]. In the report of this game, the manager for the Red Sox, Terry Francona says "I thought because of the [ineffective cut fastball] he had to work harder and gave up some hits." When we look at the significant actors for this game we identify the spin rate for the cut fastball as well as the spin rate for the four seam fastball. Shown in Figures 65 and 66, the spin rates are certainly down and more erratic compared to the mean signal for both pitches and with neither fastball containing the normal movement, it would be hard to be strategic with when to use either pitch, since the opposing team has less movement on the pitches to keep them off balance.

These anomalous games for Jon Lester provide another lesson in characterizing anomalies for this domain. It is not always the case that if a pitch is unsuccessful, it is the velocity, spin and release point all failing at the same time, but rather, each component has a degree of autonomy in making up the mechanics of the pitch. If one part of the mechanics is off, it can make the difference between a strike out and solid hit. Identifying for Lester that his

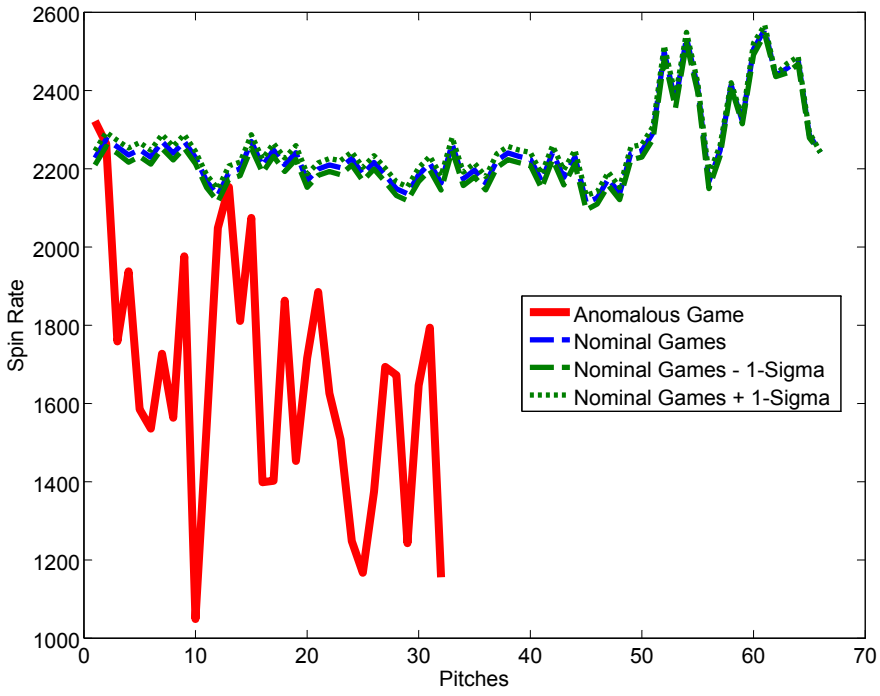

Figure 65: Spin Rate of Jon Lester's Four Seam Fastball Against the Cubs

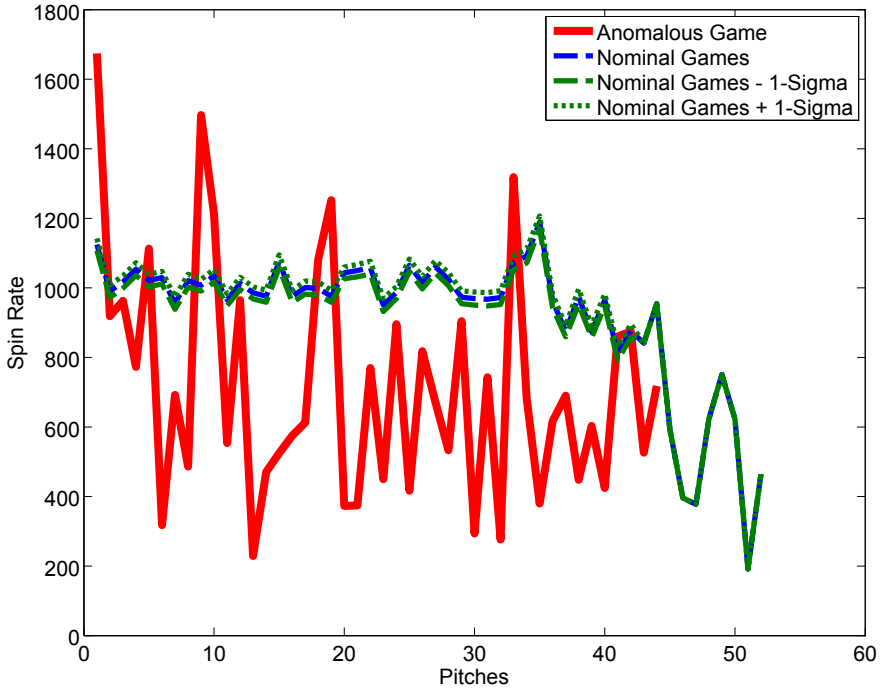

Figure 66: Spin Rate of Jon Lester's Cut Fastball Against the Cubs

spin for his cut fastballs is a primary mechanical issue means that the experts on the team, such as the pitching coach, can attempt to work with Jon Lester to improve the consistency of his cut fastball.

VII.3 Conclusion and Discussion

In this chapter, we took our exploratory approach to anomaly detection in large multivariate signal datasets and applied it to a domain that is primarily about physical human interaction. In this case we examined pitcher data from Major League Baseball. This data is sparser in terms of the signals than the aircraft flight system domain. It also has more variance from instance to instance. While the environment is more tightly controlled because it is a sport with many rules, the nature of human bodies, the variation in technique of throwing a baseball, and psychology produces many instances, even for the same pitcher that can be different. The goal of this work was to test whether our approach would be able to find a nominal set of operational instances and isolate interesting anomalies that could be used for further investigation and modeling.

We specifically applied the approach to pitchers who have been known to thrown many pitches during a game and have a proven track record of being successful. The trade off for this choice is the gathering of longer signals for the features in the data cube, versus picking instances that are more likely to be successful instances in the game. We first clustered all the data together and found that the individual mechanics and pitch selections for a pitcher could differentiate certain pitcher's entire set of games from the data cube. We then applied our approach to contextualized data cubes of individual pitchers. We produced case studies of a few pitchers and found that our approach could identify some poor games, but was more successful at identifying effective performances for the pitcher above their standard. Our feature selection allowed us to characterize both the poorer games and the excellent performances in terms of the mechanics for the specific pitchers. We showed anomalies where the pitcher was able to either improve an aspect for a pitch or even just remain more

consistent across the entire game. We also showed for the same pitcher how we could identify mechanics that could make the difference between a great start and one that was worse. Lastly, we showed that this could be effective not just for the team that the pitcher being analyzed was currently playing for, but that this could be used for opponent scouting to identify current flaws.

When compared to the aircraft flight system domain, one of the greatest contrasts is that the anomalies found in the aircraft were never improvements over the nominal operation. Even when they were not safety incidents, they still represented something that was unexpected about the takeoff of the airplane. In the baseball domain, the baseline, even for a successful pitcher leaves anomalies in two directions. There is the poor outing, where the anomaly is why the mechanics of the pitcher lead to worse performance than normal. In this domain, there are anomalies where even a successful pitcher has a better game than expected, and is in more control. As much as the experts would want to avoid the poor mechanics, understanding what helped produce such a great game would be very useful in this domain.

CHAPTER VIII

RESEARCH CONTRIBUTIONS AND FUTURE WORK

Detection of anomalies in complex systems using large data requires approaches that can accommodate the complex nature of the systems, and be efficient enough to leverage the large amount of operational data that is produced in these systems. Section I.1 identified specific research challenges for performing anomaly detection for systems that are both complex in the number of relationships and also produce large amounts of data in terms of instances, features, and the temporal sequences for each feature. Section III.3 details the problem domains that we focused on in this research and were emblematic of the research challenges.

This dissertation has described our two approaches we developed for performing anomaly detection on these complex systems. The first approach is a supervised learning methodology for improving diagnostic reference models and thus improving the accuracy and early detection times of diagnostic reasoners. It is an approach that exploits expert information to constrain both the problem and the data to discover specific knowledge that is used to update diagnostic models. The second approach is exploratory method that uses dimensionality reduction to transform the data cube described in Section III.3.1 into a two dimensional dissimilarity matrix for clustering analysis. We use the structure generated by clustering to identify anomalous instances in the domain, and then perform feature selection to help an expert characterize the nature of the anomalies. These approaches form the core of our research and are demonstrated to show their effectiveness in improving models of detection for specific anomalies, and for discovering and characterizing previously unknown anomalies for future detection.

VIII.1 Summary and Research Contributions

Chapter II provided a background on the current state of the art in anomaly detection. We first explained the different types of anomalies and then described the different data considerations when building models for anomaly detection. We then explored discriminative and generative types of data mining algorithms that can be used to construct the anomaly detection models. We explained the difference between supervised, unsupervised and semi-supervised models and review the different anomaly detection approaches that have been constructed with each level of supervision of data mining.

Chapter III formally described our research approach to this problem. We started by detailing the nature of the data that is collected from complex systems and that we use to look for anomaly detection. We then explained the problem we are solving, starting with data curation. This curation led to data transformation where the data may have dimensionality reduction applied to make it computationally feasible for a data mining. We either applied supervised learning techniques or unsupervised learning depending on the approach and the goal of the detection task. We summarized the problem domains, including formally describing the data cube representation that use to frame the data in the rest of the dissertation. Lastly, we explained our research problems in more detail with high level descriptions of the two approaches and what the goal of these approaches are when applied to the data in our problem domains.

Chapter IV presented our supervised approach to anomaly detection as a knowledge engineering task. We applied our approach to the case of improving the reference models in diagnostic systems. This improvement is designed to make the models more accurate for a specific anomaly which in turn improved the early detection performance of the diagnostic reasoner. We first described the reasoner and the reference model we targeted with this approach. We then described the Bayesian framework we developed for building models from data, and then incorporated information from these data-driven models into the reference model structure. We then presented our overall approach for targeted

anomaly detection, starting with curation of data, and using expert knowledge to target relevant segments of the data to transform the curated data into a specific data set. We then explained how we build and validate classifier models using Tree Augmented Naïve Bayes model built from the targeted dataset. Our final step was to take the information from the Tree Augmented Naïve Bayes model and suggest improvements to the reference model. We combined this approach with the aircraft flight system domain and data, along with the reasoner and reference model described earlier. Using this specific implementation of our approach, we demonstrated its effectiveness through a number of case studies of specific engine failures in the aircraft in our domain. We showed that our technique does improve the early detection times of the reasoner, as well as alert an expert when a failure being explored is not germane to the subsystem on the aircraft that we targeted.

Our major contributions in the chapter include:

- A general framework for applying targeted supervised anomaly detection with expert guidance: This framework allows a practitioner to utilize different reasoner and reference models and different domains of physical systems to start from raw data, and produce specific anomaly detection models which can be reapplied to a reasoner for targeted improvements.

- A successful implementation of this framework on an industry designed aircraft reasoner: This implementation of our general approach was able to take operation data from an aircraft, a specific aviation safety incident in the data and produce a Bayesian model that could accurately detect the failure. With the help of a human expert, and using our framework, aspects of this model could be reapplied back to the reference model. After these applications it was shown that the reference model was quicker in detecting the correct anomaly.

Chapter V described the dimensionality reduction techniques we utilized in our second approach for using unsupervised learning methods for anomaly detection. One of the main

components of our unsupervised approach is the use of dimensionality reduction of the temporal sequences to reduce the data cube into a two dimensional dissimilarity matrix. Since this reduction is so important to the overall approach, we focused on the possible measures available to us. Since we wanted the temporal signal to be compressed but the semantic information about the features to remain, we reviewed the possible complexity measures that can meet our requirements. We examined compression based measures, the information theoretic measure of approximate entropy, and then investigate the use signal analysis techniques, such as the Haar Wavelet transform. In order to compare these different complexity measures we built a series of experiments. These experiments used a combination of artificial data of many dynamic signals built specifically to test the sensitivity and monotonicity of these measures as the signals change. We measured across the different signals, as well as measuring the changes as the parameters of each signal vary. We then took the two best measures, the Compression Invariant Distance Measure using the Prediction by Partial Matching compression algorithm and the Haar Wavelet transform and examined them in the scope of a very specific real world example of EEG data. The data came from the readings of alcoholics and a control group and was a multivariate time signal. Using clustering and an N-Nearest Neighbor classifier, we settled on the use of the Haar Wavelet transform as the best complexity measure.

Our major contribution in the chapter included:

- The building of a experimental test suite for exploring the different complexity measures: While others have run tests of their own, our experiments were focused on how these measures change for two conditions, when the signals are different, and when the parameters vary for the same signal type. Both of these are important in our problem domains used for anomaly detection. It is important to recognize two signals are different if they vary drastically, but also to detect when two signals with

the same general shape have different enough parameters. Our experiments help dictate the use of the Haar Wavelet transform as the main complexity measure to use in our dimensionality reduction.

In Chapter VI we presented our unsupervised learning approach for anomaly detection. Using the dimensionality reduction techniques explored in Chapter V, we applied our approach to the domain of aircraft flight systems. We first reviewed the previous work for anomaly detection, especially other techniques that were designed specifically with aircraft flight systems. We described two of the more recent methodologies in depth, the use of principal component analysis with density based clustering and multiple kernel anomaly detection. With this previous work as the baseline, we provided a description of our second approach, starting with curation of the data cube. We then described the process of contextualization of the flight data into the specific phase to limit the anomalies we attempt to discover. From this contextualized data cube we performed dimensionality reduction, clustered the dissimilarity matrix and looked for nominal clusters and anomalous clusters. We then explained our approach for characterizing the anomalies using feature selection that looks for the features known as significant actors in the data cube that best differentiates an anomaly from the nominal set. We described how this process works when the number of anomalies is small enough for examining them one by one. We also suggested a process of characterizing groups of anomalies using an interactive approach using Targeted Projection Pursuit. Lastly, we compared our approach to the previous work. In our approach we identified the anomalies and showed how we can characterize their behavior with the significant actors. Using a domain expert for the aircraft flight domain, we used these characterization to identify if any of the anomalies were aviation safety issues. We found that a number of our anomalies are quite interesting to the expert including a flight that only used three of the four engines, and another flight that came very close to stalling before the computer controller took over. In both cases the expert would want to flag these anomalies for further investigation, and to look for these kind of anomalies in the future.

When compared to the previous work, our approach is quite successful at both identifying a reasonable number of anomalies, and easily characterizing their significant actors.

Our major contributions in the chapter include:

- An exploratory approach to discovering previously unknown anomalies in large data from complex systems: We provide an end to end approach for taking raw data and producing at the end, a likely nominal data set as well as set of characterized anomalies. This approach uses complexity based measures to perform dimensionality reduction on the data that renders the data more efficient for clustering while retaining information about the original features to identify later. The approach uses hierarchical clustering to build a rich structure that can identify high level trends and be examined deeper in the tree for more specific relationships. We present in our approach a method for producing the significant actor features that can help an expert explore the potential anomalies more closely.

- The successfully exploration of a very large flight dataset with our approach and interesting anomalies: Using our approach we were able to apply it to a flight dataset that has yet to be truly explored. In our maiden use of this data of 5333 flight we already started to discover interesting flights such as one where it used only 3 of the 4 engines. Finding these flights in the data are useful as annotations for other researchers to then apply that information to better contextualize which data is relevant. For example, this flight may likely implicate the flights before it as indicating an engine anomaly. Since there are no incident reports for the flight, an expert now has a smaller set of flights to investigate. We also found flights that implicate pilot behavior and the use of automatic throttle. These can be used to help model pilot behavior at takeoff.

Chapter VII maintained the general approach we described in Chapter VI and explored

the use of our unsupervised learning approach in the baseball domain. Using the same approach, this time we chose a domain that contrasted with the rigid nature of aircraft design. Whereas all aircraft of the same type are built to a specification, data that comes from the measurement of physical human interaction contains a variety of variation, between humans, and even from instances measures from the same human. In this chapter, we used this domain to test the flexibility of our approach. The data we selected was pitching data from Major League Baseball. We treated each instance as a game that a pitcher threw, and we selected from the data, pitchers who threw quite a few pitchers during their careers. This was an attempt to find data rich instances, but also predisposes the data to more successful games. Our goal was to look for anomalous games and then use the characterization method from our approach to identify the significant actors from the sensors that measure the pitcher's approach to throwing the ball. These sensors then provide insight in the mechanics of the pitcher. The results from our approach for a selection of pitcher found two results. First, when we grouped all the selected pitchers together, we found that they merely clump based on their pitch repertoire and their mechanics vary enough to separate them. When we applied the method to a specific pitcher in the dataset, the results helped us identify games where the pitchers perform better than usual, including games where they are near perfect according the rules of the game. We identified that for each pitcher we ran, we saw parts of their mechanics which vary more, and which impact their performance the most. Mechanics of the pitcher including where they release the ball, and how much spin they put on the pitches were two of the more common significant actors we found.

Our major contribution in the chapter includes:

- The successfully exploration of a novel data set for a relatively new purpose: Using our approach we were able to apply it to a baseball dataset. This dataset has not been used previously by researchers and specifically for the use of signal based methodologies. We found a number of interesting anomalies, and were able to relate them back to notable performances, as well as link the significant actors to information

provided by the experts in the game. This application is a novel use of the data, and showed the generality of the approach with a domain that contrasts in interesting ways with the aircraft flight system domain.

VIII.2 Future Research Directions

Given the combinatorial ways in which one can contextualize or structure specific data from different domains, and explore approaches to anomaly detection, there are many opportunities for future research. Promising research directions for our approach and problem domains include:

Studying Different Phases of Flight. The examination of takeoff is interesting because of the stress placed on the pilot and the aircraft during that phase of operation. Choosing another phase such as landing would be interesting since it also includes relatively high stress on both actors in the domain. In fact, running this exploratory analysis on several phases and then overlaying the anomalies on a time line would provide a first pass of annotations for each aircraft. These annotations could be used to identify when an aircraft suffers from multiple anomalies in a row, even if at different phases of flight. These anomalies could be contextualized across a single aircraft or across a fleet to identify likely environmental anomalies, mechanical issues, or even with added data, look for similar pilot profiles. Automatic annotation of such data would be a great first step in putting unlabeled large data into perspective.

Different Structure to the Pitcher Data. Similar to the different contextualizations for different phases of flight, the pitcher data offers a number of different ways to structure the data. Our breakdown used a subset of the available sensors, and grouped them by each pitch type. Another structure would be to produce single signals that combine all the pitches together regardless of type for each sensor. One could add a new symbolic signal of the pitch types. This advantage is that using the data extracted in this research, the padding would likely only go to 128 for each sensor, increasing their resolution and ability to look

for subtle changes. In this restructuring of the data, the clustering may identify whether different pitchers with similar mechanics approach the same teams in similar ways.

The structure may also be modified to try and reduce the amount of padding that occurs in the data. The effect here is that we remove the tangible differences in how long a pitcher goes. With this structure we may be able to relax the extraction criteria away from only pitchers that throw over 100 games. This relaxation may allow us to build more general models of pitchers from this data.

Different Wavelets and Transformations for Dimensionality Reduction. The choice of the Haar wavelet was predicated on the simplicity of the approach and diversity of applications that use the Haar Wavelet for analysis. There are however, other choices that might be interesting to examine that may also allow us to relax the need for the same size signals between instances, or at least requiring the signal to be a power of two. Shannon entropy based wavelets could be a potentially interesting choice. The application and success of information theoretic measures such as approximate entropy in other domains suggests that an information theoretic wavelet would be an interesting experiment. Using another transformation such as the iterative function systems used with fractals is another measure that may be interesting. Together, these changes to the approach would make for an interesting comparison on either of the problem domains.

APPENDIX A

LIST OF PUBLICATIONS

Our research has lead to the following journal, conference, and workshop publications.

A.1 Refereed Journal Publications

J-1 **Daniel L.C. Mack**, Gautam Biswas, Xenofon Koutsoukos, and Dinkar Mylarswarmy, "Learning Bayesian Structures to Augment Diagnostic Reference Models", *Journal of Engineering Applications of Artificial Intelligence*, 2013 *Submitted*.

J-2 Joseph W. Hoffert, **Daniel L.C. Mack**, and Douglas Schmidt, "Integrating Machine Learning Techniques to Adapt Protocols for QoS-enabled Distributed Real-time and Embedded Publish/Subscribe Middleware", *International Journal of Network Protocols and Algorithms (NPA): Special Issue on Data Dissemination for Large-scale Complex Critical Infrastructures*, Vol 2, No 3 2010

A.2 Refereed Conference Publications

C-1 John S. Kinnebrew, **Daniel L.C. Mack**, and Gautam Biswas, "Mining Temporally-Interesting and Characteristic Learning Behavior Patterns", *The Sixth International Conference on Educational Data Mining*. Memphis, TN. July 2013 *Submitted*.

C-2 Josh D. Carl, **Daniel L.C. Mack**, Ashraf Tantawy, Gautam Biswas, and Xenofon Koutsoukos, "Fault Detection and Isolation for Spacecraft Systems: An Application to a Power Distribution Testbed", *SAFEPROCESS-2012*. Mexico City, Mexico. 2012.

C-3 **Daniel L.C. Mack**, Gautam Biswas, Xenofon Koutsoukos, Dinkar Mylarswarmy

and George Hadden,"Deriving Bayesian Classifiers from Flight Data to Enhance Aircraft Diagnosis Models", *Annual Conference of the Prognostics and Health Management Society*, Montreal, Canada. October 2011.

C-4 John S. Kinnebrew, **Daniel L.C. Mack**, Gautam Biswas, and Douglas C. Schmidt, "Coordination of Planning and Scheduling Techniques for a Distributed, Multi-level, Multi-agent System", *The International Conference on Agents and Artificial Intelligence (ICAART 2010)*. Vallencia, Spain. January 2010.

A.3 Refereed Workshop Publications

W-1 **Daniel L.C. Mack**, Gautam Biswas, Xenofon Koutsoukos, and Dinkar Mylarswarmy,"Using Tree Augmented Naive Bayes Classifiers to Improve Engine Fault Models", *Uncertainty in Artificial Intelligence: Bayesian Modeling Applications Workshop*, Barcelona, Spain. July 2011.

W-2 Joseph W. Hoffert, **Daniel L.C. Mack**, and Douglas Schmidt, "Using Machine Learning to Maintain Pub/Sub System QoS in Dynamic Environments", *The 8th Workshop on Adaptive and Reflective Middleware (ARM) 2009*, Urbana Champaign, IL. December 2009

A.4 Other Publications

O-1 **Daniel L.C. Mack**, Dan Brooks, and Gautam Biswas "Baseball Prospectus News: Introducing Pitch Sequence Visualizations", *Baseball Prospectus*. September, 2012.

APPENDIX B

LIST OF ACRONYMS

ALT	Altimeter
AOAC	Corrected Angle of Attack
AOAI	Indicated Angle of Attack
ApEn	Approximate Entropy Measure
ASIAS	Aviation Safety Information Analysis and Sharing Database
ATEN	Automatic Throttle Engaged
BALX	Barometrically Adjusted Altitude
BLAC	Body Latitudinal Acceleration
BWT	Burrows-Wheeler transform
CDM	Compression-based Dissimilarity Measure
CI	Conditional Indicators
CiDM	Complexity-Invariant Distance Measure
CPT	Conditional Probability Table
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Networks
EAI	Engine De-Ice Activated
EGT.X	Engine Temperature of Engine X

EM Expectation-Maximization algorithm

FAA Federal Aviation Administration

FF.X Fuel Flow for Engine X

FQTY.X Fuel Quantity in Tank X

GBN General Bayesian Networks

HMM Hidden Markov Model

LATG Lateral Axis Acceleration

LCD Local Causal Discovery

LONG Longitudinal Axis Acceleration

LZW Lempel-Ziv Welch Algorithm

MKAD Multiple Kernel Anomaly Detection

NCD Normalized Compression Distance

N1.X Fan Speed for Engine X

N2.X Core Speed for Engine X

NB Naïve Bayesian network

PCA Principal Component Analysis

PCA-DBSCAN Principal Component Analysis with Density Based Clustering

PBN Partial Bayesian Networks

PLA.X Power Level Angle of Engine X

PPM Prediction by Partial Matching

PS Static Pressure

PSA Average Static Pressure

PT Total Pressure

RUDP Position of the Rudder

SBAC Similarity-based Agglomerative Clustering

SVM Support Vector Machine

TAN Tree Augmented Naïve Bayesian networks

VRTG Vertical Acceleration

REFERENCES

[1] C. D. A. Banerjee, P. Burlina. A support vector method for anomaly detection in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(8), Aug 2006.

[2] R. Acharya, V. S. Sree, S. Chattopadhay, and S. J. Suri. Automated diagnosis of normal and alcoholic eeg signals. *International Journal of Neural Systems*, 22(03), 2012.

[3] E. Aleskerov, B. Freisleben, and B. Rao. Cardwatch: a neural network based database mining system for credit card fraud detection. In *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pages 220 –226, mar 1997.

[4] C. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *The Journal of Machine Learning Research*, 11:171–234, 2010.

[5] P. G. J. Aslin, Matthew J. Central maintenance computer system and fault data handling method, 07 1990.

[6] G. Batista, X. Wang, and E. Keogh. A complexity-invariant distance measure for time series. In *SDM-2011: Proceedings of SIAM International Conference on Data Mining, Philadelphia, PA, USA*, 2011.

[7] S. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 29–38, 2003.

[8] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 407–416. ACM, 2000.

[9] L. Bettencourt, R. Ribeiro, G. Chowell, T. Lant, and C. Castillo-Chavez. Towards real time epidemiology: data assimilation, modeling and anomaly detection of health surveillance data streams. In *Proceedings of the 2nd NSF conference on Intelligence and security informatics: BioSurveillance*, pages 79–90. Springer-Verlag, 2007.

[10] A. K. Bhaskar, C. Ramakrishna, and H. Aggarwal. Data compression techniques. 1985.

[11] J. R. Binkley and S. Singh. An algorithm for anomaly-based botnet detection. In *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the*

Internet - Volume 2, pages 7–7, Berkeley, CA, USA, 2006. USENIX Association.

[12] C. M. Bishop. *Pattern Recognition and Machine Learning*. 2007.

[13] G. Biswas, J. B. Weinberg, and D. H. Fisher. Iterate: a conceptual clustering algorithm for data mining. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 28(2):219 –230, may 1998.

[14] R. J. Bolton and D. J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):pp. 235–249, 2002.

[15] E. Brigham and R. Morrow. The fast fourier transform. *Spectrum, IEEE*, 4(12):63–70, 1967.

[16] D. Broomhead and G. Kind. Extracting qualitative dynamics from experimental data. *Physica D*, 20:217–236, 1986.

[17] S. Budalakoti, S. Budalakoti, A. Srivastava, M. Otey, and M. Otey. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(1):101 –113, 01 2009.

[18] S. Budalakoti, A. Srivastava, R. Akella, and E. Turkov. Anomaly detection in large sets of high-dimensional symbol sequences. *NASA Ames Research Center, Tech. Rep. NASA TM-2006-214553*, 2006.

[19] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. 1994.

[20] Y. Cai, H. Wang, and W. Zhang. Learning patterns of motion trajectories using real-time tracking. *Advanced Materials Research*, 403:2768–2771, 2012.

[21] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58, July 2009.

[22] V. Chandola, S. Boriah, and V. Kumar. A reference based analysis framework for analyzing system call traces. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, CSIIRW '10, pages 33:1–33:3, 2010.

[23] K. Chaudhary, J. Yadav, and B. Mallick. A review of fraud detection techniques: Credit card. *International Journal of Computer Applications*, 45(1):39–44, 2012.

[24] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137(1-2):43 – 90, 2002.

[25] D. Chickering. Learning bayesian networks is np-complete. *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG-*, pages 121–130, 1996.

[26] D. M. Chickering, D. Heckerman, and C. Meek. A bayesian approach to learning bayesian networks with local structure. In *In Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1997.

[27] K. Chidananda Gowda and T. Ravi. Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity. *Pattern Recognition*, 28(8):1277–1282, 1995.

[28] T. Chidester. Understanding normal and atypical operations through analysis of flight data. In *Proceedings of the 12th International Symposium on Aviation Psychology*, 2003.

[29] R. Chimelis. A better Jon Lester doesn't get the win, but the Red Sox do. http://www.masslive.com/redsox/index.ssf/2012/07/dont_bury_him_yet_red_sox_left.html.

[30] E. Chu, D. Gorinevsky, and S. Boyd. Detecting aircraft performance anomalies from cruise flight data. In *AIAA Infotech @ Aerospace Conference*, pages 29–38, 2010.

[31] M. Chuah and F. Fu. Ecg anomaly detection via time series analysis. In *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*, pages 123–135. Springer, 2007.

[32] R. Cilibrasi and P. Vitányi. Clustering by compression. *Information Theory, IEEE Transactions on*, 51(4):1523–1545, 2005.

[33] J. Cleary and I. Witten. Data compression using adaptive coding and partial string matching. *Communications, IEEE Transactions on*, 32(4):396–402, 1984.

[34] I. Cohen, M. Goldszmidt, T. Kelly, J. Symons, and J. S. Chase. Correlating instrumentation data to system states: a building block for automated diagnosis and control. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, pages 16–16, Berkeley, CA, USA, 2004. USENIX Association.

[35] M. Das and S. Parthasarathy. Anomaly detection and spatio-temporal analysis of global climate system. In *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data*, pages 142–150. ACM, 2009.

[36] S. Das, B. Matthews, and R. Lawrence. Fleet level anomaly detection of aviation safety data. In *Prognostics and Health Management (PHM), 2011 IEEE Conference on*, pages 1 –10, june 2011.

[37] S. Das, B. Matthews, A. Srivastava, and N. Oza. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 47–56. ACM, 2010.

[38] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *Information Theory, IEEE Transactions on*, 36(5):961–1005, 1990.

[39] R. Dearden and D. Clancy. Particle filters for real-time fault detection in planetary rovers. In *Proc. of the 12th International Workshop on Principles of Diagnosis*, DX 2001, pages 1–6, 2001.

[40] O. Depren, M. Topallar, E. Anarim, and M. Ciliz. An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert systems with Applications*, 29(4):713–722, 2005.

[41] B. Efron. The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, pages 892–898, 1975.

[42] J. Faith. Targeted projection pursuit for interactive exploration of high-dimensional data sets. In *Information Visualization, 2007. IV'07. 11th International Conference*, pages 286–292. IEEE, 2007.

[43] J. S. Farris. On the cophenetic correlation coefficient. *Systematic Biology*, 18(3):279–285, 1969.

[44] O. Faust, R. Acharya, A. Allen, and C. Lin. Analysis of eeg signals during epileptic and alcoholic states using ar modeling techniques. *IRBM*, 29(1):44 – 52, 2008.

[45] T. Fawcett and F. Provost. Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3):291–316, 1997.

[46] T. Felke. Application of model-based diagnostic technology on the boeing 777 airplane. In *Digital Avionics Systems Conference, 1994. 13th DASC., AIAA/IEEE*, pages 1 –5, 1994.

[47] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[48] D. H. Fisher. Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research*, 4:147–179, 1996.

[49] D. H. Fisher. Data mining tasks and methods: Clustering: conceptual clustering. In *Handbook of data mining and knowledge discovery*, pages 388–396. Oxford University Press, Inc., 2002.

[50] J. E. Fowler and R. Yagel. Lossless compression of volume data. In *Proceedings*

of the 1994 symposium on Volume visualization, VVS '94, pages 43–50, New York, NY, USA, 1994. ACM.

[51] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[52] Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–602. Ieee, 2005.

[53] S. Gaddam, V. Phoha, and K. Balagani. K-means+ id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):345–354, 2007.

[54] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.

[55] P. Gogoi, B. Borah, and D. Bhattacharyya. Anomaly detection analysis of intrusion data using supervised & unsupervised approach. *Journal of Convergence Information Technology*, 5(1):95–110, 2010.

[56] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

[57] K. Gowda and G. Krishnan. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978.

[58] D. Grossman and P. Domingos. Learning bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 46–, New York, NY, USA, 2004. ACM.

[59] M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1):pp. 10–18, 2009.

[60] G. Hamerly and C. Elkan. Bayesian approaches to failure prediction for disk drives. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 202–209, 2001.

[61] J. Hartigan and M. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[62] G. Hazel. Multivariate gaussian mrf for multispectral scene segmentation and anomaly detection. *Geoscience and Remote Sensing, IEEE Transactions on*, 38(3):1199–1211, 2000.

[63] D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[64] M. Hejazi and Y. Singh. Credit data fraud detection using kernel methods with support vector machine. *Journal of Advanced Computer Science and Technology Research*, 2(1), 2012.

[65] R. Hilborn. *Chaos and Nonlinear Dyanamics*. Oxford Univ. Press, 1994.

[66] Honeywell. Vehicle integrated prognostic reasoner. *NASA Contractor Report to appear*, NNL09AD44T, 2010.

[67] W. Hu, Y. Liao, and V. Vemuri. Robust support vector machines for anomaly detection in computer security. In *Proc. 2003 International Conference on Machine Learning and Applications (ICMLAâĂŹ03)*, 2003.

[68] Z. Huang and M. Ng. A fuzzy k-modes algorithm for clustering categorical data. *Fuzzy Systems, IEEE Transactions on*, 7(4):446–452, 1999.

[69] S. R. I.A. Rezek. Stochastic complexity measures for physiological signal analysis. *Biomedical Engineering, IEEE Transactions on*, 45(9), Sept 1998.

[70] D. Iverson. Inductive system health monitoring. In *Proceedings of The 2004 International Conference on Artificial Intelligence*, 2004.

[71] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[72] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, UAI'95, pages 338–345, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[73] A. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841, 2002.

[74] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

[75] S. Kay and S. Marple. Spectrum analysis- a modern perspective. 69, 1981.

[76] M. Kearns and Y. Mansour. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In *Proceedings of the 15th International Conference on Machine Learning*, pages 269–277. Citeseer, 1998.

[77] E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S.-H. Lee, and J. Handley. Compression-based data mining of sequential data. *Data Min. Knowl. Discov.*, 14(1), Feb. 2007.

[78] S. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in bioinformatics*, 4(3):228–235, 2003.

[79] S. Knerr, E. Augustin, O. Baret, and D. Price. Hidden markov model based word recognition and its application to legal amount reading on french checks. *Computer Vision and Image Understanding*, 70(3):404–419, 1998.

[80] D. E. Knuth. Dynamic huffman coding. *Journal of algorithms*, 6(2):163–180, 1985.

[81] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[82] A. Kolmogorov. Three approaches to the definition of the concept âĂIJquantity of informationâĂİ. *Problemy peredachi informatsii*, 1(1):3–11, 1965.

[83] M. Kousarrizi, A. Ghanbari, A. Gharaviri, M. Teshnehlab, and M. Aliyari. Classification of alcoholics and non-alcoholics via eeg using svm and neural networks. In *Bioinformatics and Biomedical Engineering , 2009. ICBBE 2009. 3rd International Conference on*, pages 1 –4, june 2009.

[84] J. Kruskal, Joseph B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):pp. 48–50, 1956.

[85] R. Kwitt and U. Hofmann. Unsupervised anomaly detection in network traffic by means of robust pca. In *Computing in the Global Information Technology, 2007. ICCGI 2007. International Multi-Conference on*, pages 37–37. IEEE, 2007.

[86] S. P. L.A. Fleisher and S. Rosenbaum. Approximate entropy of heart rate as a correlate of postoperative ventricular dysfunction. 78:683–692, 1993.

[87] T. Lane. A decision-theoritic, semi-supervised model for intrusion detection. *Machine Learning and Data Mining for Computer Security*, pages 157–177, 2006.

[88] P. Larrañaga, M. Poza, Y. Yurramendi, R. Murga, and C. Kuijpers. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(9):912–926, 1996.

[89] R. Laxhammar. Anomaly detection for sea surveillance. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008.

[90] K. Lee, L. Guillemot, Y. Yue, M. Kramer, and D. Champion. Application of the gaussian mixture model in pulsar astronomy–pulsar classification and candidates ranking for {\ it Fermi} 2fgl catalog. *Arxiv preprint arXiv:1205.6221*, 2012.

[91] U. Lerner, R. Parr, D. Koller, and G. Biswas. Bayesian fault detection and diagnosis in dynamic systems. In *Proc. of the 17th Nat. Conf. on Artificial Intelligence*, AAAI 2000, pages 531–537, San Mateo, CA, USA, 2000. AAAI Press.

[92] K. Leung and C. Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342. Australian Computer Society, Inc., 2005.

[93] D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. *Machine Learning: ECML-98*, pages 4–15, 1998.

[94] L. Li, M. Gariel, R. Hansman, and R. Palacios. Anomaly detection in onboard-recorded flight data using cluster analysis. In *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, pages 4A4–1 –4A4–11, oct. 2011.

[95] M. Li, M. Ng, Y. Cheung, and J. Huang. Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters. *Knowledge and Data Engineering, IEEE Transactions on*, 20(11):1519–1534, 2008.

[96] H. W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, 1967.

[97] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003.

[98] J. S. X. M. Ester, H. Kriegel. A density-based algorithm for discovering clusters in large spatial databases with noise. *Knowledge Discover in Databases*, pages 226–231, 1996.

[99] D. L. C. Mack, G. Biswas, X. Koutsoukos, and D. Mylaraswamy. Using tree augmented naive bayes classifiers to improve engine fault models. In *Uncertainty in Artificial Intelligence: Bayesian Modeling Applications Workshop*, Barcelona, Spain, 07 2011.

[100] D. L. C. Mack, G. Biswas, X. Koutsoukos, and D. Mylaraswamy. Learning bayesian structures to augment diagnostic reference models. 2013.

[101] D. L. C. Mack, G. Biswas, X. Koutsoukos, D. Mylaraswamy, and G. Hadden. Deriving bayesian classifiers from flight data to enhance aircraft diagnosis models. *Annual Conference of the Prognostics and Health Management Society*, 2011.

[102] M. Madden. A new bayesian network structure for classification tasks. *Artificial Intelligence and Cognitive Science*, pages 183–197, 2002.

[103] S. Manganaris and D. H. Fisher. Learning time series for intelligent monitoring. 1994.

[104] G. Manzini. An analysis of the burrowsĂŤwheeler transform. *J. ACM*, 48(3):407–430, May 2001.

[105] S. Mascaro, K. B. Korb, and A. E. Nicholson. Anomaly detection in vessel tracks using bayesian networks. In *Proceedings of the 8th Bayesian Modeling Applications Workshop*, 2011.

[106] F. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, pages 68–78, 1951.

[107] D. Mazel and M. Hayes. Using iterated function systems to model discrete sequences. *Signal Processing, IEEE Transactions on*, 40(7):1724–1734, 1992.

[108] M. Mehta, J. Rissanen, R. Agrawal, et al. Mdl-based decision tree pruning. In *Proc. 1st Intl. Conf. Knowledge Discovery and Data Mining (KDD95), Montreal, Canada*, 1995.

[109] N. Merhav, M. Gutman, and J. Ziv. On the estimation of the order of a markov chain and universal data compression. *Information Theory, IEEE Transactions on*, 35(5):1014–1019, 1989.

[110] J. Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine learning*, 4(2):227–243, 1989.

[111] T. Mitchell. *Machine learning. 1997*. McGraw Hill, 1997.

[112] S. Mukkamala, G. Janoski, and A. Sung. Intrusion detection using neural networks and support vector machines. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 2, pages 1702–1707. Ieee, 2002.

[113] C. Mulcahy. Image compression using the haar wavelet transform. *Spelman Science and Mathematics Journal*, 1(1):22–31, 1997.

[114] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *Neural Networks, IEEE Transactions on*, 12(2):181 –201, mar 2001.

[115] K. Murphy. Dynamic bayesian networks. *Probabilistic Graphical Models, M. Jordan*, 2002.

[116] S. Nation. Jon Lester dominates Twins with complete game gem. http://www.sbnation.com/2010/5/20/1481284/red-sox-twins-score-recap-may-20.

[117] J. Nelson and N. Kingsbury. Fractal dimension, wavelet shrinkage, and anomaly detection for mine hunting. *IET Signal Processing*, 2012.

[118] D. Ormoneit and V. Tresp. *Improved Gaussian mixture density estimates using Bayesian penalty terms and network averaging*. 1995.

[119] N. Pal and J. Bezdek. On cluster validity for the fuzzy c-means model. *Fuzzy Systems, IEEE Transactions on*, 3(3):370–379, 1995.

[120] Z. Pan, S. Chen, G. Hu, and D. Zhang. Hybrid neural network and c4. 5 for misuse detection. In *Machine Learning and Cybernetics, 2003 International Conference on*, volume 4, pages 2463–2467. IEEE, 2003.

[121] S. Parthasarathy, M. Zaki, M. Ogihara, and S. Dwarkadas. Incremental and interactive sequence mining. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 251–258. ACM, 1999.

[122] A. Pawling, P. Yan, J. Candia, T. Schoenharl, and G. Madey. Anomaly Detection in Streaming Sensor Data. *ArXiv e-prints*, Oct. 2008.

[123] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.

[124] R. Perdisci, G. Gu, and W. Lee. Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 488–498. IEEE, 2006.

[125] J. Phipps. Dendrogram topology. *Systematic Biology*, 20(3):306–308, 1971.

[126] S. Pincus. Approximate entropy (apen) as a complexity measure. 5(1), 1995.

[127] D. Poole. Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[128] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001*. Citeseer, 2001.

[129] A. Press. Jon Lester gem, Jason Varitek triple lift Red Sox by Royals. http://scores.espn.go.com/mlb/recap?gameId=310821107.

[130] A. Press. Phillies' Roy Halladay struggles but beats Rockies for 14th win. http://scores.espn.go.com/mlb/recap?gameId=310803127.

[131] A. Press. Roy Halladay's complete game shuts down Mets as Phils win. http://usatoday30.usatoday.com/sports/baseball/nl/2011-04-30-phillies-mets_N.htm.

[132] A. Press. Tim Lincecum retires 21 straight batters to help Giants beat A's again. http://scores.espn.go.com/mlb/recap?gameId=310521126.

[133] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[134] S. Rajasegarar, C. Leckie, and M. Palaniswami. Anomaly detection in wireless sensor networks. *Wireless Communications, IEEE*, 15(4):34 –40, aug 2008.

[135] C. Rao. The use and interpretation of principal component analysis in applied research. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 329–358, 1964.

[136] G. Rätsch, B. Schölkopf, S. Mika, and K. Müller. Svm and boosting: One class. *Submitted to NIPS 2000*, 2000.

[137] I. Roychoudhury, G. Biswas, and X. Koutsoukos. Comprehensive diagnosis of continuous systems using dynamic bayes nets. In *Proc. of the 19th International Workshop on Principles of Diagnosis*, DX 2008, pages 151–158, 2008.

[138] D. K. Ruch and P. J. Van Fleet. *Wavelet theory: an elementary approach with applications*. Wiley-Interscience, 2011.

[139] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 3 edition, 2010.

[140] J. Ryan, M. Lin, and R. Miikkulainen. Intrusion detection with neural networks. In *Advances in neural information processing systems*, pages 943–949. MORGAN KAUFMANN PUBLISHERS, 1998.

[141] J. Sander, M. Ester, H. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998.

[142] M. Sardana and R. Agrawal. A comparative study of clustering methods for relevant gene selection in microarray data. *Advances in Computer Science, Engineering & Applications*, pages 789–797, 2012.

[143] K. Sayood. *Introduction to data compression*. Morgan Kaufmann, 2000.

[144] D. Sculley and C. Brodley. Compression and machine learning: A new perspective on feature space vectors. In *Data Compression Conference, 2006. DCC 2006.*

Proceedings, pages 332–341. IEEE, 2006.

[145] B. Shah and B. Trivedi. Artificial neural network based intrusion detection system: A survey. *International Journal of Computer Applications*, 39(6), 2012.

[146] A. Sharma and P. Panigrahi. A review of financial accounting fraud detection based on data mining techniques. *International Journal of Computer Applications*, 39(1), 2012.

[147] D. Sharp, A. Bell, J. Gold, K. Gibbar, D. Gvillo, V. Knight, K. Murphy, W. Roll, R. Sampigethaya, V. Santhanam, and S. Weismuller. Challenges and solutions for embedded and networked aerospace software systems. *Proceedings of the IEEE*, 98(4):621 –634, april 2010.

[148] I. G. S.M. Pincus and R. Ehrenkranz. Heart rate control in normal and aborted sids infants. 264, 1993.

[149] P. Smyth. Hidden markov models for fault detection in dynamic systems. *Pattern Recognition*, 27(1):pp. 149–164, 1994.

[150] S. Söyler, E. Kurt, and O. DaÇğ. Optimization of the magnetic anomaly signals from a new land mine detection device. In *Proceedings of the 11th international conference on Applications of Electrical and Computer Engineering*, pages 178–183. World Scientific and Engineering Academy and Society (WSEAS), 2012.

[151] C. Spitzer. Honeywell primus epic aircraft diagnostic and maintenance system. *Digital Avionics Handbook*, (2):pp. 22–23, 2007.

[152] A. Statnikov and C. Aliferis. Analysis and computational dissection of molecular signature multiplicity. *PLoS computational biology*, 6(5):e1000790, 2010.

[153] D. Stein, S. Beaven, L. Hoff, E. Winter, A. Schaum, and A. Stocker. Anomaly detection from hyperspectral imagery. *Signal Processing Magazine, IEEE*, 19(1):58–69, 2002.

[154] G. Stein, B. Chen, A. Wu, and K. Hua. Decision tree classifier for network intrusion detection with ga-based feature selection. In *Proceedings of the 43rd annual Southeast regional conference-Volume 2*, pages 136–141. ACM, 2005.

[155] P. Subramani, R. Sahu, and S. Verma. Feature selection using haar wavelet power spectrum. *BMC bioinformatics*, 7(1):432, 2006.

[156] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Data Mining, Fifth IEEE International Conference on*, page 8 pp., nov. 2005.

[157] N. N. Taleb. *The black swan: The impact of the highly improbable*. Random House

Trade Paperbacks, 2010.

[158] Q. Tran, H. Duan, and X. Li. One-class support vector machine for anomaly network traffic detection. In *The 2nd Network Research Workshop of the 18th APAN*, 2004.

[159] I. Tsamardinos, L. Brown, and C. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[160] N. Underhill. Boston bats cover lackluster performance from Jon Lester in win over the Cubs. http://www.masslive.com/sports/index.ssf/2011/05/boston_bats_cover_lackluster_p.html.

[161] H. Vaccaro and G. Liepins. Detection of anomalous computer session activity. In *Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on*, pages 280–289. IEEE, 1989.

[162] V. Vapnik. The nature of statistical learning theory. 1995.

[163] A. Varga and R. Moore. Hidden markov model decomposition of speech and noise. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 845–848. IEEE, 1990.

[164] V. Verma, G. Gordon, R. Simmons, and S. Thrun. Real-time fault diagnosis. *IEEE Robotics and Automation Magazine*, 11(2):pp. 56–66, 2004.

[165] M. Vetterli and C. Herley. Wavelets and filter banks: Theory and design. *Signal Processing, IEEE Transactions on*, 40(9):2207–2232, 1992.

[166] C. S. Wallace. *Statistical and inductive inference by minimum message length*. Springer Verlag, 2005.

[167] C. S. Wallace and D. L. Dowe. Mml clustering of multi-state, poisson, von mises circular and gaussian distributions. *Statistics and Computing*, 10:73–83, 2000.

[168] K. Wang, J. Parekh, and S. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Advances in Intrusion Detection*, pages 226–248. Springer, 2006.

[169] C. Whitrow, D. Hand, P. Juszczak, D. Weston, and N. Adams. Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, 18(1):30–55, 2009.

[170] I. Witten and E. Frank. *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations*. 1999.

[171] J. Xue and D. Titterington. Comment on âĂIJon discriminative vs. generative classifiers: A comparison of logistic regression and naive bayesâĂİ. *Neural processing*

letters, 28(3):169–187, 2008.

[172] J. H. Y. He and B. Zhang. Approximate entropy as a nonlinear feature parameter for fault diagnosis in rotating machinery. 23(4), 2012.

[173] N. Ye, M. Xu, and S. Emran. Probabilistic networks with undirected links for anomaly detection. In *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, pages 175–179, 2000.

[174] C. Yoo and E. Brilz. The five-gene-network data analysis with local causal discovery algorithm using causal bayesian networks. *Annals of the New York Academy of Sciences*, 1158(1):93–101, 2009.

[175] M. Zaki. Sequence mining in categorical domains: incorporating constraints. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 422–429. ACM, 2000.

[176] Y. Zhang and D. Adjeroh. Prediction by partial approximate matching for lossless image compression. *Image Processing, IEEE Transactions on*, 17(6):924–935, 2008.

[177] Y. Zhao, B. Li, X. Li, W. Liu, and S. Ren. Customer churn prediction using improved one-class support vector machine. *Advanced Data Mining and Applications*, pages 731–731, 2005.

[178] S. Zhong and J. Ghosh. Hmms and coupled hmms for multi-channel eeg classification. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, volume 2, pages 1154 –1159, 2002.