Naval Research Laboratory
Washington, DC 20375-5320

NRL/8120/MR—2022/1

Applying Machine Learning Anomaly Detection Techniques to U.S. Navy Space System Operations

ERIC J. PESOLA

Advanced Systems Technology Branch
Space Systems Development Division

January 31, 2021

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
31-01-2022	NRL Memorandum Report	July 20 2020 – July 20 2021

4. TITLE AND SUBTITLE

Applying Machine Learning Anomaly Detection Techniques to U.S. Navy Space System Operations

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
NISE

6. AUTHOR(S)

Eric J. Pesola

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
N2Y2

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/8120/MR--2022/1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

10. SPONSOR / MONITOR'S ACRONYM(S)

NRL-NISE

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Karles Fellowship

14. ABSTRACT

This report documents the first year of a Karle's Fellowship research project investigating applications of machine learning to enhanced spacecraft operations. The first year of the fellowship was primarily comprised of research scope determination, literature review, data collection, and algorithm selection and development. In recent years the United States (U.S.) Department of Defense (DoD) has placed an increased emphasis on the development of autonomous capabilities, and this has been echoed in U.S. Navy research and development strategy. Machine learning technology represents a near-term opportunity to incrementally improve autonomous capabilities through the augmentation of existing technology. In the longer term, it is an investment opportunity into new technology which may drastically improve the capabilities of DoD systems. Practical approaches to the autonomy problem must focus on removing the most significant barriers to autonomy before more sophisticated technology becomes realistic. In the context of space system operations, health monitoring and fault management has been identified by both government and commercial entities as one of the largest inhibitors to space system autonomy. The increasing size and complexity of space systems as well as the rapid adoption of satellite constellations has quickly made it impractical for traditional ground-based human monitoring to be sustainable. This work primarily investigates the use of machine learning for automated anomaly detection in satellite telemetry. Anomaly detection is one of the foundational responsibilities of autonomous health monitoring because the detection of off-nominal state is typically the first step in the operational fault detection and remediation process. Near-term automated anomaly detection can assist human operators by sorting through large amounts of telemetry and flagging only the data which requires investigation. In the long term, it may be used as part of an integrated autonomous health monitoring system. The field of machine learning for anomaly detection has been the subject of extensive research and as a result, the technology is mature enough to be applied to current health monitoring systems. Additionally, anomaly detection has a number of potential applications to other Naval interests including Maritime Domain Awareness (MDA) and Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) efforts. The second year of the fellowship will focus on conducting experiments, establishing a proof of concept, and integrating the technology into existing space systems.

15. SUBJECT TERMS

Autonomy, Anomaly detection, Health monitoring, Machine learning, Spacecraft, Fault Management

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Eric J. Pesola
U	U	U	U	41	**19b. TELEPHONE NUMBER** *(include area code)* (202) 279-4433

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank.

CONTENTS

FIGURES

TABLES

This page intentionally left blank

EXECUTIVE SUMMARY

This report documents the first year of a Karle's Fellowship research project investigating applications of machine learning to enhanced spacecraft operations. The first year of the fellowship was primarily comprised of research scope determination, literature review, data collection, and algorithm selection and development. In recent years the United States (U.S.) Department of Defense (DoD) has placed an increased emphasis on the development of autonomous capabilities, and this has been echoed in U.S. Navy research and development strategy. Machine learning technology represents a near-term opportunity to incrementally improve autonomous capabilities through the augmentation of existing technology. In the longer term, it is an investment opportunity into new technology which may drastically improve the capabilities of DoD systems. Practical approaches to the autonomy problem must focus on removing the most significant barriers to autonomy before more sophisticated technology becomes realistic. In the context of space system operations, health monitoring and fault management has been identified by both government and commercial entities as one of the largest inhibitors to space system autonomy. The increasing size and complexity of space systems as well as the rapid adoption of satellite constellations has quickly made it impractical for traditional ground-based human monitoring to be sustainable. This work primarily investigates the use of machine learning for automated anomaly detection in satellite telemetry. Anomaly detection is one of the foundational responsibilities of autonomous health monitoring because the detection of off-nominal state is typically the first step in the operational fault detection and remediation process. Near-term automated anomaly detection can assist human operators by sorting through large amounts of telemetry and flagging only the data which requires investigation. In the long term, it may be used as part of an integrated autonomous health monitoring system. The field of machine learning for anomaly detection has been the subject of extensive research and as a result, the technology is mature enough to be applied to current health monitoring systems. Additionally, anomaly detection has a number of potential applications to other Naval interests including Maritime Domain Awareness (MDA) and Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) efforts. The second year of the fellowship will focus on conducting experiments, establishing a proof of concept, and integrating the technology into existing space systems.

This page intentionally left blank

APPLYING MACHINE LEARNING ANOMALY DETECTION TECHNIQUES TO U.S. NAVY SPACE SYSTEM OPERATIONS

1. INTRODUCTION

This report documents the first year of a two-year Karle's Fellowship research project on the topic of machine learning (ML) for enhanced spacecraft operations. It is intended to provide a comprehensive review of all research thus far, including United States (U.S.) Department of Defense (DoD) and Navy need, research scope determination, background domain knowledge, literature review, data collection, algorithm selection and development, experiment design, and other findings. Further, it should serve as a motivator and foundation for the final year of research and subsequent follow-on projects.

1.1 The Jerome and Isabella Karle Fellowship Program

The U.S. Naval Research Laboratory (NRL) established the Jerome and Isabella Karle Distinguished Scholar Fellowship Program in honor of Drs. Jerome and Isabella Karle, whose outstanding contributions to the chemistry field earned them recognition within the United States and internationally, including a Nobel Prize award for Dr. Jerome Karle. The program provides researchers the opportunity to perform independent research projects lasting between 12 and 24 months which are funded internally by NRL. The Naval Center for Space Technology (NCST) at NRL accepted a Karle's Fellowship beginning July 2020 and ending July 2022, entitled "Machine Learning for Enhanced Spacecraft Operations." The primary focus of this fellowship is the identification, adaptation, and application of promising ML algorithms to spacecraft operations, and the creation of a proof of concept to evaluate how ML may improve space system performance. The provided stipulations for this proof of concept are as follows:

1. A ML algorithm
2. A data set for training the algorithm
3. A spacecraft simulation test bed for evaluating the algorithm
4. Analysis results quantifying any change in performance

There are many potential applications of ML to space missions, introducing the possibility for improvements to both operations and mission performance [1]. In order to align this research with U.S. Navy need and maintain a tractable scope for the proof of concept, the project began with a survey of the Navy's needs and goals in the space domain. This was succeeded by a review of general spacecraft operations and existing technological gaps to identify promising areas for ML application. The remaining introductory content and background outline this refinement in research scope and direction.

Manuscript approved January 21, 2022.

1.2 U.S. Navy Need for Enhanced Space System Autonomy

The U.S. Navy conducts numerous activities in the space domain, including Maritime Domain Awareness (MDA), Positioning Navigation and Timing (PNT), and Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) [2]. These activities provide the infrastructure for critical decision-making information to be collected, analyzed, and distributed worldwide. The importance of these systems is reflected in DoD investment strategy. For example, the 2018 National Defense Strategy (NDS) and the Naval Research and Development Framework each cite resilient and persistent C4ISR as an investment priority [2, 3]. As the systems which provide these capabilities continue to increase in size and complexity in response to increased operational demand, it becomes necessary to integrate more autonomous functionality in order to maintain the maximum possible level of system performance. The DoD has recognized this need, and as a result the 2018 NDS and a number of other strategic documents explicitly outline the advancement of autonomous capabilities as a key technological investment [2–5]. It is therefore necessary for the U.S. Navy to continue developing autonomous technology so that critical systems are able to scale with the needs of the force.

Developing autonomous capabilities in U.S. Navy space systems will help to improve the quantity, quality, and timeliness of actionable information. Autonomy can be applied to multiple aspects of space systems, including spacecraft operations and mission performance. From a purely operational perspective, greater autonomy allows space systems to scale up without overwhelming human operators. For example, autonomy becomes necessary when the time scale of decision-making or magnitude of information processing precludes manual control [6]. This is becoming increasingly commonplace as distributed multi-satellite missions become more prevalent. Additionally, autonomy offers the opportunity for greater system performance at a lower cost due to more robust and efficient operations [7]. From a missions perspective, increased autonomy can take the form of improvements in data collection, processing, analysis, and transmission, increasing mission efficiency and effectiveness. Improvements to operations and mission execution both result in reduced strain on human-in-the-loop systems and in some instances may eliminate the need for a human entirely, allowing operators to take on higher-level tasks [5, 8].

While the Navy should continue to invest in a variety of autonomous capabilities in the long-term, near-term efforts must focus on removing the most significant barriers to autonomy. Multiple authorities have identified spacecraft fault management (FM) as one of the largest space system autonomy inhibitors, including the Defense Science Board (DSB) and National Aeronautics and Space Administration (NASA) [4, 9–11]. In general, FM is an enabling technology for autonomous systems because it allows a system to continue operating in the event of off-nominal conditions [12]. For the Navy, autonomous FM, and more broadly health monitoring, offers several benefits including increased resiliency and reliability. Systems with the ability to predict, prevent, isolate, and recover from faults are inherently more likely to remain operational during critical periods. Additionally, autonomous health monitoring can help alleviate the difficulty of assessing the health state of space systems which are becoming increasingly large and complex.

1.3 Research Questions

There is a clear U.S. Navy need for continued investment in autonomous space system capabilities. For this reason, the current scope of work is concerned with determining how ML may be applied to space system autonomy. Specific consideration is given to spacecraft FM systems because of the significant challenge and opportunity they pose as an autonomy enabler. This research seeks to obtain answers to the following questions:

1. What are the most promising applications of ML to autonomous space system operations?
2. How can ML be integrated into spacecraft FM systems?
3. What FM tasks are appropriate for ML?

2. BACKGROUND

To motivate and provide context for subsequent chapters, this chapter includes information on autonomy and automation, ML fundamentals, space system operation, and spacecraft FM. Each section is intended to provide sufficient background information in each relevant topic such that its role may be understood within the larger context of the report.

2.1 Automation, Autonomy, and Artificial Intelligence

Automation, autonomy, and artificial intelligence (AI) are three distinct concepts which are often conflated due to their conceptual similarities. When creating a system which operates at least partially outside of human control, care must be taken to consider which of these concepts should be employed. For example, a process which must be strictly repeatable and deterministic is a much stronger candidate for automation than autonomy or AI. Conversely, a system which must be able to adapt to unforeseen circumstances and make decisions must employ some measure of autonomy. These concepts can also be leveraged to work together; for example, an autonomous system may rely on automated components to perform repetitive functions and AI-based components to assist with decision-making tasks. It is important to emphasize that a given system may employ one or more of these concepts while still utilizing some form of human-in-the-loop control. To provide a clear delineation, this report adopts the following definitions from [6]:

Automation is the automatically-controlled operation of an apparatus, process, or system which takes the place of human labor. Though automated processes may be sophisticated, they are strictly deterministic and all actions taken by the system are chosen through predetermined decision criteria. In other words, automated processes function by following explicit instructions and cannot operate outside of those instructions.

Autonomy is the ability of a system to achieve goals while operating independently of external control. Two key characteristics of an autonomous system are self-direction and self-sufficiency. Note that the operational boundaries - limits to what the system may and may not do - can and should be implemented by the system designers. Further, specific autonomous capabilities can exist in a larger system which operates under human control. For a more in-depth treatment of this topic, see [4].

Artificial Intelligence is the capability of computer systems to perform tasks that normally require human intelligence. A system which uses AI may use it to aid in decision-making at a number of different levels of autonomy, and can be implemented to work alongside human operators within well-defined operational bounds. While AI is a common component of systems which have some measure of autonomy, it is important to note that an autonomous system does not necessarily require an AI component.

2.2 Fundamental Machine Learning Concepts

Machine learning can be defined as a subset of AI in which computer algorithms automatically improve at a given task given repeated exposure to data. Modern deep learning techniques which utilize large artificial neural networks are among the most prevalent ML methods. Figure 1 shows the relationships between AI, ML, and deep learning. This report uses the term machine learning in reference to present-day practices employing algorithms which work best at targeted tasks; this is known as "weak AI," or AI that performs a single task which is narrow in scope. Common applications of this include classification and regression tasks. In contrast, artificial general intelligence, "strong AI," and related terms broadly refer to the type of cognitive intelligence possessed by humans. Although research is being performed in this area, this type of AI is unlikely to be applicable in the near term and will not be discussed further.

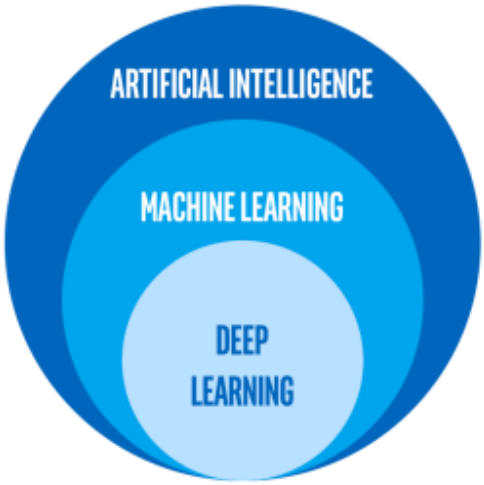

Fig. 1—A common Euler diagram of the AI field [13]

While ML models may vary greatly in their architecture, most employ the same fundamental implementation pattern. In general, a ML model maps some input data to a corresponding output. Training data is used during the model optimization process to determine the optimal model parameters. It is therefore imperative that the data used to train a ML model represents the problem as holistically as possible. Any biases, absent modalities, or other problem representation errors in the data can cause a model to give biased predictions or simply perform poorly. Furthermore, the metrics used to optimize and evaluate a model must be carefully chosen such that they accurately portray the model's performance. This is true regardless of whether a given algorithm works in isolation or is part of a larger integrated system. It is also true regardless of learning paradigm.

This report broadly defines learning paradigm as the manner in which a ML model's parameters are tuned. Figure 2 shows the three main paradigms: supervised learning, unsupervised learning, and reinforcement learning. These basic paradigms may serve as constituents for other derived paradigms such as semi-supervised learning. Supervised learning is the most well-studied and is used most often in practice. In this paradigm, each training data sample is labeled: it includes model input data as well as the corresponding ground truth, or target output. Models are optimized by tuning the model parameters to minimize the value of some arbitrary loss function which compares model output with ground truth. In contrast, unsupervised

learning algorithms search for patterns and structure in the input data without without leveraging explicit feedback from labeled ground truth data. Finally, reinforcement learning uses a feedback system to train an intelligent agent to take actions in its environment which maximize a reward scheme tailored to the agent's desired performance [14].

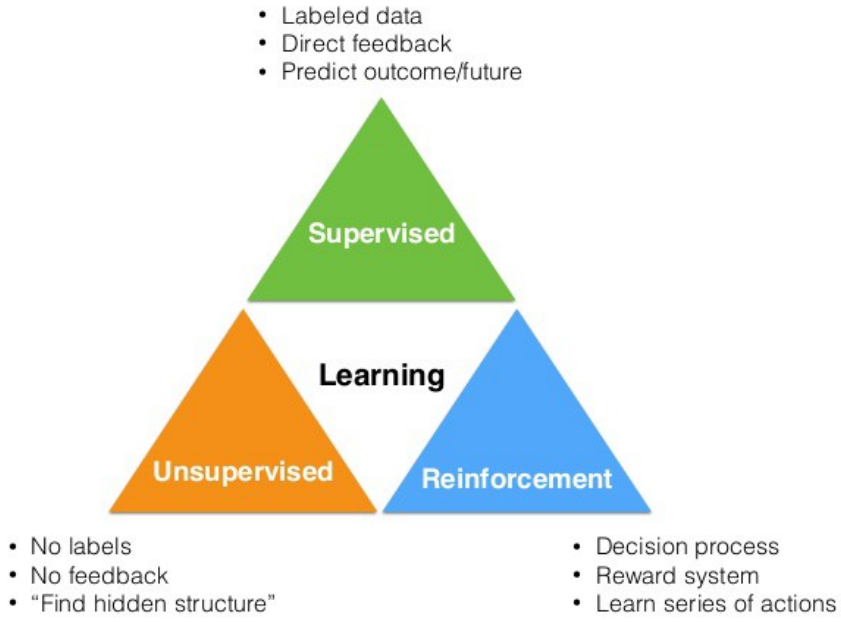

Fig. 2—The three basic machine learning paradigms [15]

2.3 Space System Anatomy and Operation

Post-launch, satellite systems consisting of unmanned spacecraft typically are comprised of three main operational segments: a space segment, a ground segment, and a user segment. The typical space system segments and their role in the system are shown in Figure 3. This section focuses on the space and ground segments because they serve as the foundational infrastructure for the user segment, which in turn provides end users with the products of the satellite system.

2.3.1 Space Segment

The space segment encompasses all spacecraft in the satellite system; this may consist of a single spacecraft or a constellation of many spacecraft. Satellites may vary greatly in their specific construction, but the prototypical satellite can be deconstructed into two main components: the bus and payload. While payloads serve a specific mission-dependent function, the primary objective of any spacecraft bus is to facilitate the in-space portion of the mission by providing the infrastructure necessary to support the payload(s). Though the presence and importance of specific subsystems may vary by mission, this discussion includes the typical subsystems for Earth-orbiting satellites which are pictured in Figure 4.

For brevity, these subsystems will not be discussed in detail. However, it is important to note that each serves a specific role in the spacecraft's infrastructure. As such, telemetry data is typically collected

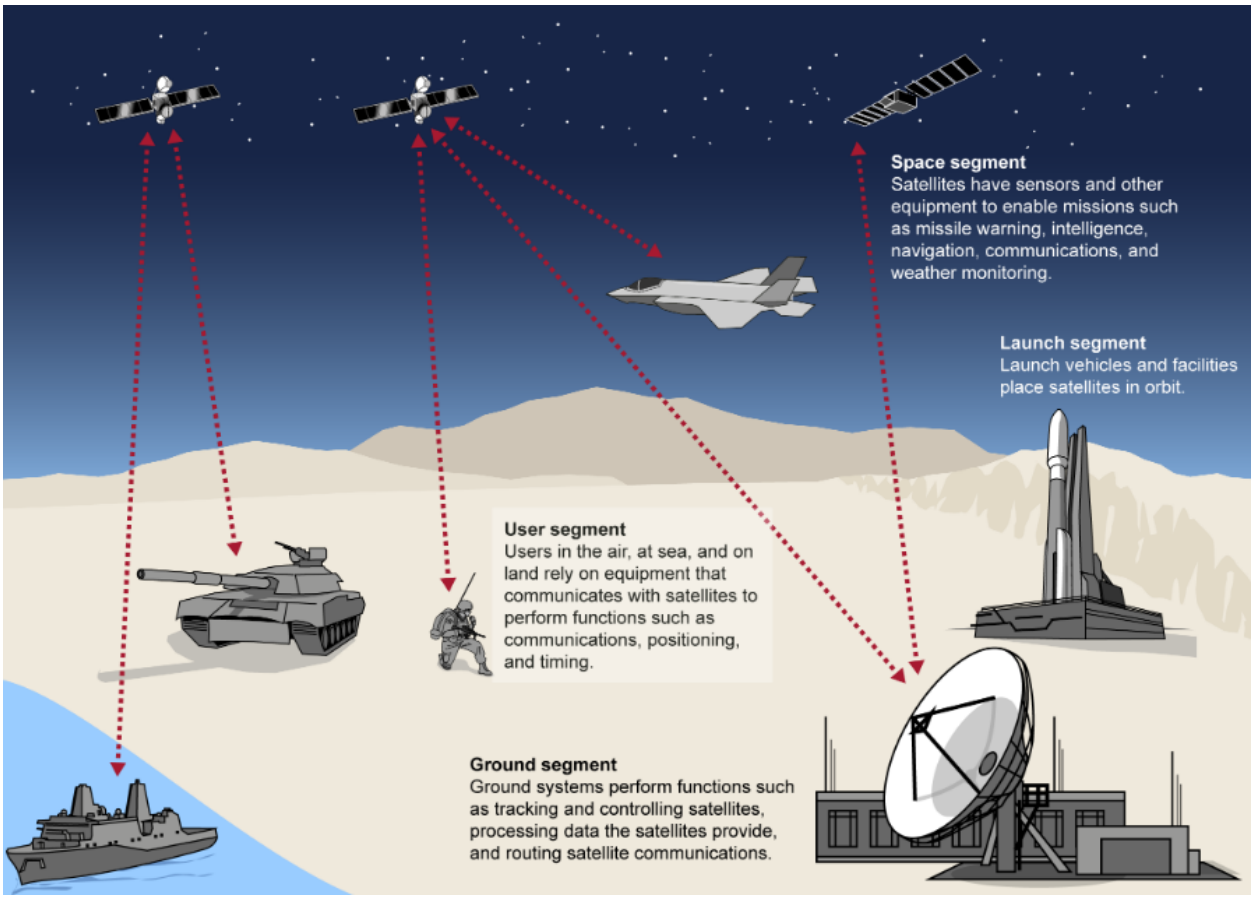

Fig. 3—Typical space system segments [16]. Operationally (i.e., post-launch), the three relevant segments are the ground, space, and user segments.

from each of these subsystems to assess the state of health of the spacecraft, often at multiple levels of hierarchy such as the component, subsystem, and system level. The number of telemetry channels recorded for a spacecraft may range from the dozens to thousands. Typically, the type and quantity of telemetry data collected is determined by subject matter experts who decide what level of information is necessary to support a given mission.

2.3.2 Ground Segment

The main responsibility of the ground segment is to interface with the space segment and distribute various types of data throughout the rest of the system. A typical ground segment consists of several elements, including a Mission Control Center or Mission Operations Center, ground stations, ground networks, and remote infrastructure. An overview of typical ground segment components is given in Figure 5. The ground segment interfaces with the space and launch segments and takes over control of the mission from the Launch Control Center post-launch. The primary functions of each element are given in Table 1 [17]. The primary operational role of the ground segment is to facilitate communication with the space segment, allowing for data to be up-linked to and down-linked from the spacecraft. Any space segment functionality which is not implemented onboard the spacecraft must be monitored and carried out by the ground segment via command and control communications. The types of data transmitted to the ground include mission data, telemetry

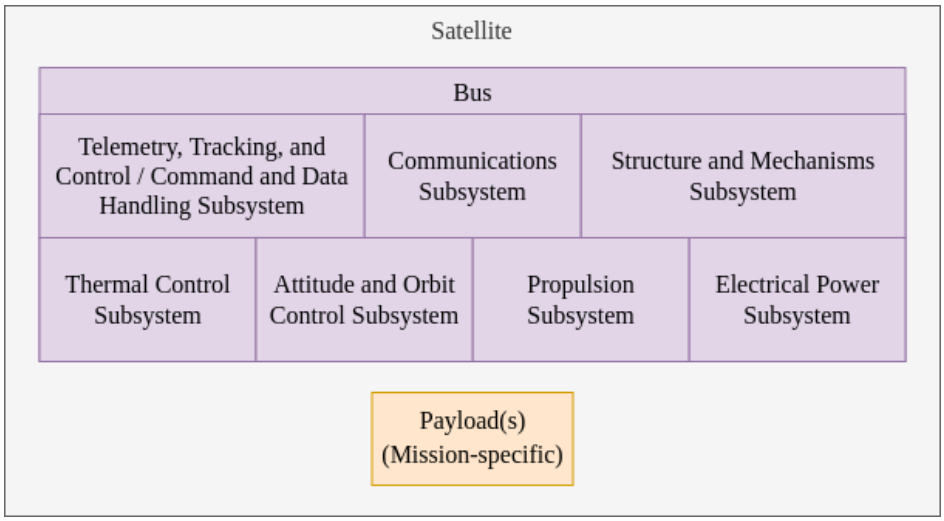

Fig. 4—A satellite is typically comprised of a bus, which provides the infrastructure for the mission, and the payload, which performs the mission task(s). Typical bus subsystems are pictured.

or housekeeping data, and tracking data. Typically, the data transmitted to the spacecraft is command and control data. Telemetry and tracking data is used by mission operators to perform various housekeeping activities such as orbit calculation and maintenance, mission planning, and spacecraft health assessments [17].

Table 1—High-level Ground Segment Functions

Element	Function
Ground Stations	Telemetry, tracking, and command interface with the spacecraft
Ground Networks	Connection between multiple ground elements
Control Centers	Management of the spacecraft operations
Remote Terminals	User interface to retrieve transmitted information for additional processing

2.4 Spacecraft Fault Management

Spacecraft FM is still a maturing discipline. Although FM practices have existed as long as spaceflight itself, it is still common for FM to be implemented on an ad hoc, mission-by-mission basis. Within the past decade, however, multiple members of the space industry have acknowledged the need for standardization and have begun to organize FM into a formal systems engineering discipline [18]. As a result there has been significant progress in the formalization of FM activities as well as the aggregation of best practices and lessons learned from previous missions. Part of this effort has included definitions for FM terminology. With regard to FM practices this report adopts the definitions from the NASA Fault Management Handbook [19]. Some important definitions are repeated here:

Anomaly - The unexpected performance of intended function.

Failure - The unacceptable performance of an intended function.

Fault - A physical or logical cause, which explains a failure.

Fault Diagnosis - Determining the possible locations and/or causes of a failure.

Fault Management - The engineering discipline that encompasses practices which enable an operational system to contain, prevent, detect, isolate, diagnose, respond to, and recover from conditions that may interfere with nominal mission operations.

Nominal - An intended, acceptable state or behavior.

Off-Nominal - A state or behavior beyond the boundaries of possible expected states or behaviors. There are three off-nominal states: anomalous, degraded, and failed.

Prognosis - Prediction of future states or behaviors.

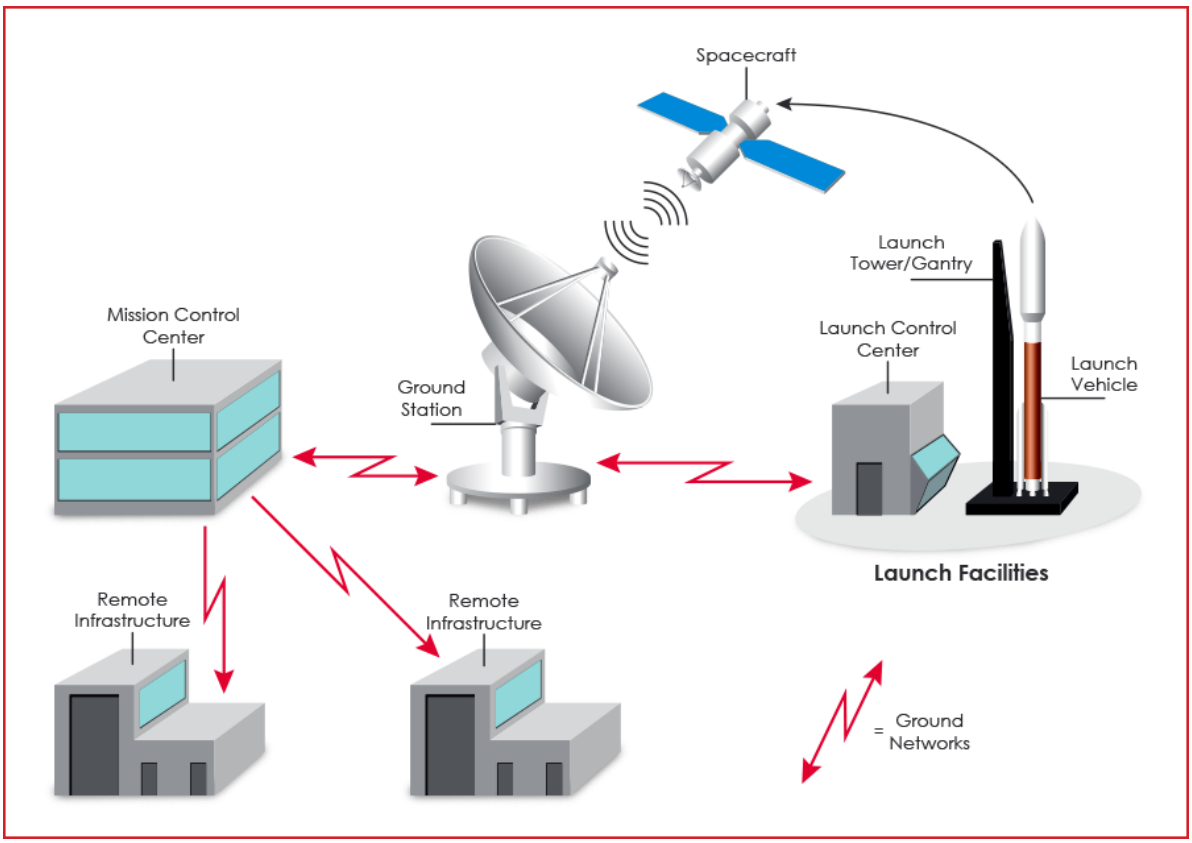

Fig. 5—Simplified overview of ground segment components and their interfaces with the space and launch segments [20].

2.4.1 Fundamental Requirements and Responsibilities

As with other disciplines, the specific requirements for a given FM system are derived from the basic mission objectives. Particular consideration must be given to the mission's goals, importance, and risk tolerance, which aid in determining the mission's risk posture. Requirements flow down from the mission level to the system, subsystem, and component levels in a manner similar to other disciplines. Once the

overall FM requirements are established, it is the job of the engineering team to determine how to satisfy the requirements while also adhering to mission resource limitations. Both bottom-up and top-down analyses are performed to obtain the most complete view of the system possible. Common analyses performed include Failure Mode and Effects Analyses (FMEA), success tree analyses, fault tree analyses, and event sequencing. Similarly to other space system tasks, any FM functionality which does not exist onboard the spacecraft must be implemented on the ground. A specific FM function may be assigned to the space segment, the ground segment, or a mixture of both [21].

NASA's FM handbook subdivides FM strategies into two approaches: prevention and tolerance [19]. Whereas prevention strategies work to avoid failures altogether, tolerance strategies seek to enable the mission to continue in the presence of failures. Prevention can be further divided into design-time prevention and operational prevention. Design-time prevention refers to the engineering practices which minimize the likelihood of a failure of occurring. Operational prevention works by first performing prognosis on an operating system and then taking preventative actions to avoid any anticipated failure. Tolerance strategies can be divided into masking, recovery, and goal change approaches. Masking approaches seek to minimize the effect of a failure by preventing it from propagating further in the system's function. Examples of masking include redundancy and error correction which allow a failure to occur but "hide" it from the rest of the system by correcting it before it propagates. Recovery approaches seek to diagnose the root cause and location of a fault and subsequently take actions to restore nominal system operation. This process is often referred to as Fault Detection, Isolation (location), and Recovery/Response (FDIR). Lastly, the goal change approach responds to a failure by modifying the mission goals to accommodate any change in system capability caused by the failure.

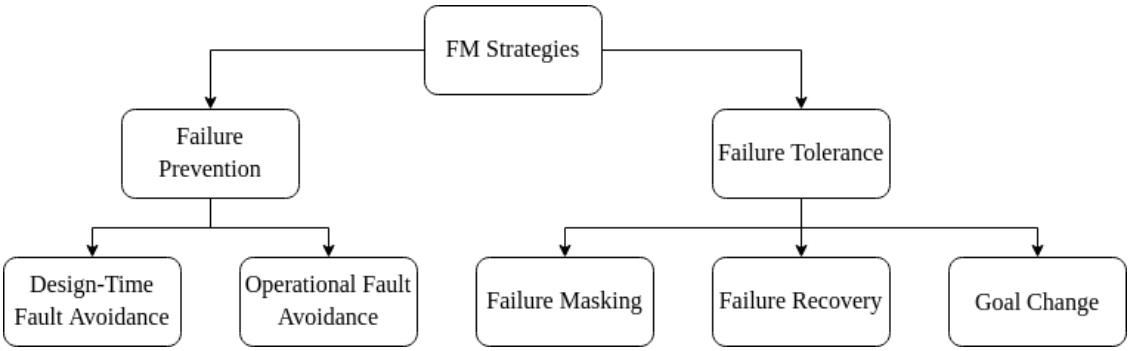

Fig. 6—Organization of the five main FM strategies according to NASA's FM Handbook [19]

In general, the specific capabilities of a mission's FM system depend on requirements which derive from the mission's objectives, complexity, and overall reliability expectations. It is typical in practice for a given mission to employ some combination of the approaches shown in Figure 6 when creating a FM system and strategy. For critical missions, most or all of the listed strategies may be employed to maximize the likelihood of mission success.

2.4.2 Current Operational Approaches

Once the system is operational there are a number of fundamental tasks which a FM system must perform. An overview of the operational FM process is given in Figure 7. Though not pictured, prognosis is also a common step in the process. The distribution of functionality between the space and ground segments

depends on several factors including mission criticality, budget, resources, and operational constraints. However, the same fundamental tasks apply regardless of the system configuration. Traditionally, FM tasks have been biased toward the ground segment due to the increased analysis capability it affords [22]. While modern spacecraft operations typically employ some functionality in both the ground and space segments, it is still common for the only onboard FM functions to be those which cannot practically be performed on the ground due to timing or communication constraints: when time-critical actions must be taken to ensure the safety of the spacecraft, the onboard system must be able to take the appropriate action independently of external aid. Additionally, there has historically existed a cultural bias against the extensive use of spacecraft flight software due to the perceived risk [4, 23].

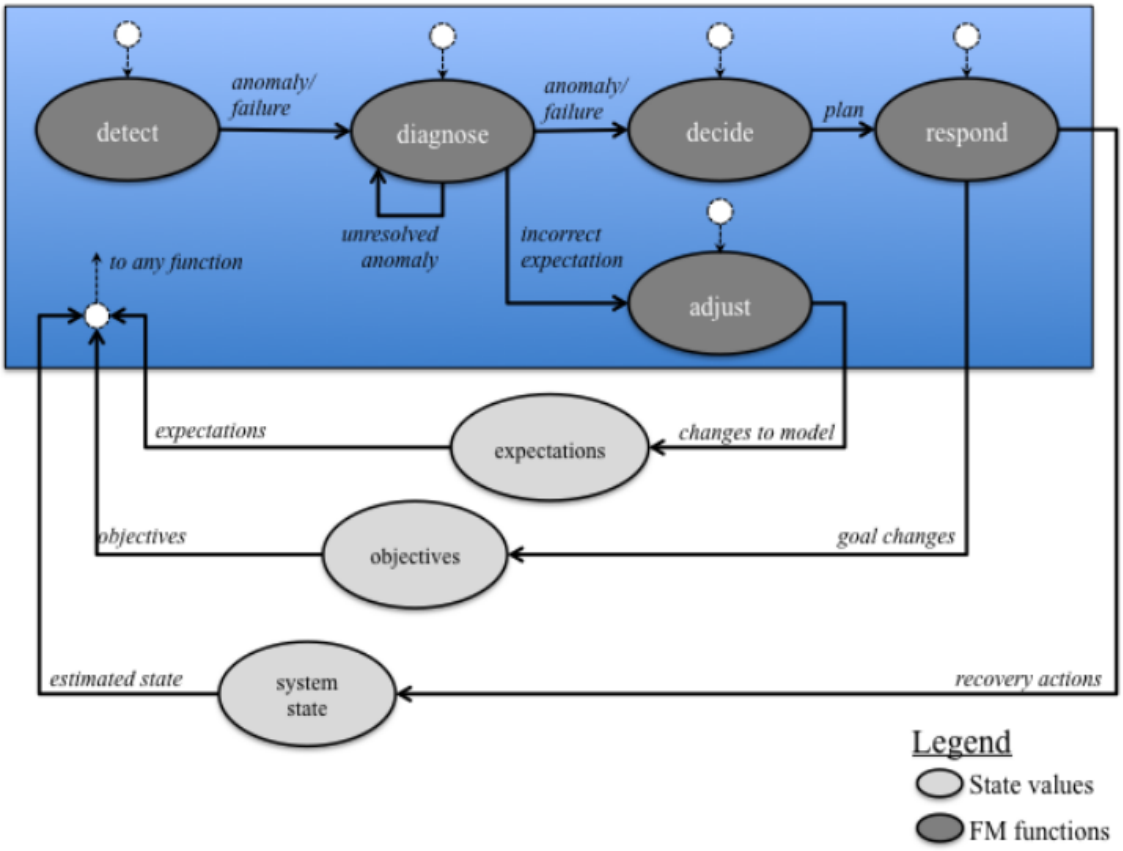

Fig. 7—Operational FM process overview (prognosis not pictured) [19]

Telemetry monitoring and analysis is the primary method of conducting operational FM efforts such as prognosis and FDIR. Telemetry is used to predict, diagnose, and isolate faults, as well as aid in determining appropriate response actions. In the event of anticipated or actual off-nominal conditions, response actions take the form of command sequences which are intended to preserve or restore a nominal operating state. These command sequences may be automatically generated if a response to a specific fault has been predetermined, or they may be constructed manually by engineers in the event of a novel or unknown fault. Monitoring often takes place on the ground via down-linked telemetry data as well as onboard through some combination of hardware and software. Some examples of onboard FM integrated into hardware include watchdog timers and built-in tests, as well as various subsystem-specific measures such as over-voltage/under-voltage monitors in the electrical system. In software, the most common onboard FM fault

response is a safe mode procedure which suspends all non-essential functionality until mission engineers can command the spacecraft back into a nominal state [21].

Out-of-limits (OOL) approaches are perhaps the simplest and most common method of telemetry monitoring [23]. The approach relies on the assumption and expectation that each telemetry channel has well-defined nominal and off-nominal ranges. These ranges are typically determined by or with assistance from spacecraft engineers who have subject matter expertise. Additionally, OOL approaches often employ tiered systems wherein multiple limits are established and each successive limit signifies a transition into a more severe off-nominal state. Tiered OOL systems have remained extremely popular due to their ease of implementation, use, and interpretation. The limits themselves are determined by experts, the system is easy to implement, and the results are straightforward to interpret. Additionally, the established limits can be used to aid with prognosis through the use of telemetry trend analyses. Limit checking is the most common telemetry monitoring method both on the ground and in space, and often serves as the foundation for more advanced methods [24].

Most current FM systems which incorporate some degree of automation rely on a rule-based approach within the monitor-response paradigm [12]. The basic premise of a rule-based approach is straightforward: for each rule, one or more conditions are continuously monitored, and if all conditions are met, a predefined response is executed. The rules can be simple or complex, both in terms of the number of conditions and their complexity. Similarly, responses may be simple or may be comprised of an extended sequence of commands. A valid response may also be to alert a human operator in the event of an off-nominal condition not solvable by the system. Rule-based systems provide for powerful automation when fault conditions and the appropriate responses are known. However, they are typically not suited to handle unknown faults unless a reasoning component is included, such as in a formal expert system [25].

2.4.3 Challenges and Opportunity

The traditional FM approach has a number of drawbacks which have been highlighted by the FM community. This extends to both the fundamental concept of operations as well as specific methods. Culturally, the lack of standardization and recognition of FM as a formal engineering task has made it nearly impossible for mission-to-mission practices to be put in place. Additionally, there is rarely effective reuse of tools or resources [18]. From a systems engineering perspective, FM is a challenging task because it is impossible to model or predict in advance every possible failure mode of a complex system. No matter how thorough, analyses such as FMEA cannot anticipate all possible faults. This large failure space "makes comprehensive testing infeasible," posing significant challenges to reliable Verification and Validation (V&V) [26]. While analysis and testing remain critical steps to any FM approach, it has been recognized that they cannot be considered fully exhaustive from a V&V perspective.

There are also operational challenges, especially for high-reliability missions. Notably, the current monitor-response paradigm has several shortcomings. The ground segment is inherently limited in its ability to perform FM because it does not have physical access to the spacecraft. Additionally, not all telemetry collected by a spacecraft can be down-linked and monitored, complicating ground-based FDIR efforts. This phenomenon has begun to compound as constellations become more prevalent. Safe mode, the most common automated FM function, can lead to extended periods of mission downtime while operators diagnose and recover the spacecraft. Failed spacecraft recovery attempts can result in a "recovery loop" in which other faults are induced and must be resolved [22]. Most limit checking systems are fairly rigid and limits must be manually changed over time as natural degradation of spacecraft function occurs. Rule-based systems can

provide for sophisticated automation but are typically based on traditional FM analyses and thus suffer from the same limitations regarding failure space. Moreover, rule-based systems become increasingly complicated to verify and validate as the number and complexity of rules increases [12].

The existing challenges to effective spacecraft FM have led to several responses by the FM community of practice. Engineering improvements are concerned with requirements development and assignment, cost drivers, risk assessment, and V&V throughout the product development life-cycle. Operationally, the primary topic of interest is creating FM practices which can scale to meet the increasing demands of space missions [27]. As with other aspects of spacecraft operation, human-in-the-loop practices are becoming less sustainable as the number and complexity of spacecraft increases. At NRL, this has led to the development of automated functionality in the Neptune ground system software [28]. More broadly, it has driven increased interest in areas such as Model-Based Systems Engineering (MBSE) and Integrated System Health Management (ISHM) [29].

3. APPROACH

Spacecraft FM is a broad discipline and some of its constituent tasks are not suitable candidates for autonomy or ML. Fundamentally, ML-enhanced systems never find practical use unless it can be quantitatively shown that the ML component does not pose a risk of reduced system performance, and this will prove especially true for spacecraft operations. In the near-term, certain FM tasks may be too sophisticated for current technology. For some tasks, the addition of ML may never offer a legitimate benefit over simpler methods and in fact should be deliberately avoided. This can be demonstrated in the recovery portion of the FDIR process; if a response strategy is known for a given fault or failure mode then automation is the only tool necessary. Replacing deterministic automation with non-deterministic ML in this scenario likely introduces more risk to the system than it removes. A more suitable task for ML in this situation may be to suggest recovery commands in the event of an unknown fault. Therefore, the most promising ML applications are those which never bring the system into a worse state, to within some quantifiable measure of certainty.

Efforts to apply ML to system health monitoring are ongoing in multiple domains, including significant research in structural and industrial health monitoring as well as network and IoT systems [30, 31]. Applying ML to spacecraft FM and health monitoring is also an active area of research spanning back multiple decades [32]. Some of the original attempts at autonomous FM were rule-based ML expert systems [11, 25]. Historically, anomaly detection in telemetry has been one of the most common applications of ML to spacecraft health monitoring [7, 24, 33, 34]. A variety of techniques for anomaly detection have been used including clustering and distance-based approaches, neural networks, support vector machines, and spectral techniques, among others [23]. More recently, modern deep learning techniques have become a popular research area [30, 35]. In general, the vast majority of research focuses only on the prognosis and detection portions of the operational FM cycle because they are the most straightforward applications. However, there is a substantial opportunity to integrate these approaches into a complete FM system.

Anomaly detection is a foundational building block for autonomous health monitoring because off-nominal state detection is often the first step in the prognosis and FDIR process. An automated telemetry anomaly detection system offers benefits to both present-day space system operations as well as future in-space autonomous health monitoring. In either case, the maximum benefit will be achieved via integration into an operational system. In the near term, automated anomaly detection offers the opportunity to alert operators to anomalous behavior before failures occur. Additionally, an automated system can distill large

amounts of telemetry down to a small number of events which bear human investigation, drastically improving efficiency [7]. Near-term systems can be integrated into ground stations as well as onboard spacecraft for high-priority missions. As part of an integrated autonomous FM system, the detection of anomalous behavior can be passed on to a higher-level reasoner which may execute the next FM task, depending on context.

The general anomaly detection problem can be broadly segmented into purely data-driven and integrated model approaches. Data-driven approaches rely on the model input data to provide all information necessary to solve the problem. Even without explicitly incorporating domain knowledge, successful anomaly detection systems have still been created in this way. Additionally, data-driven methods may be the only viable approach when it is not possible or realistic to create other types of models. There are inherent drawbacks, however, in attempts to characterize the nominal behavior of a complex system via purely data-driven methods. This is especially true for spacecraft whose telemetry values are dependent on a wide variety of internal and external factors such as spacecraft operating mode, command sequences, environmental conditions, and physical phenomena. In practice, even extremely high-capacity models are incapable of capturing all of this context. As a result, many approaches in the literature construct a new model for each individual telemetry channel [23]. This is impractical at a systems level which may require dozens of channels to be monitored.

As with other fields, domain knowledge can be leveraged to create a more targeted task to be solved. In the context of spacecraft anomaly detection, the notion of nominal performance is often captured during the system design through modeling and simulation. Just as these tools are used to help inform limit-checking and rule-based approaches, they can also be used to further inform ML methods. The integrated model approach seeks to enhance the ability of the algorithm by applying domain knowledge. This may be accomplished in a variety of ways, but in the context of space systems the most common method is typically through physical or procedural models which characterize the intended behavior of the system. As a motivating example, consider the signal decomposition in Figure 8; by modeling the trend and cyclical portions of the signal it is possible to extract any un-modeled effects in the form of a residual. In this way, the residual provides a direct measure of deviation from expected behavior. Applying an anomaly detection technique to the residual rather than to the entire signal is therefore a much more targeted problem because it is directly characterizing the deviation from modeled nominal behavior. In general, "black box" (i.e., purely data-driven) ML methods must learn a more sophisticated function mapping because they fail to explicitly incorporate any sort of domain knowledge about the system. By using non-ML models of a given system to account for easily characterized nominal behavior, it is possible to frame a ML problem which may be easier to solve.

3.1 Practical Considerations

Due to the fact that the current research is being performed at a proof-of-concept technology readiness level (TRL), many practical implementation considerations of the conceptualized system have not been explored in depth. Size, Weight, Power, and Cost (SWaP-C) concerns are not addressed in this research, nor are computational limitations such as computational cost, complexity, or memory. While these limitations may be less severe for a ground-based FM system, they are critical considerations for implementing any ML system in space. The required hardware is often large, massive, and power-intensive, and algorithms consume significant computational resources. Additionally, the behavior of ML hardware in the space environment is not well understood and requires further research before widespread use is realistic [37]. Spacecraft are resource-constrained systems which operate in a demanding environment; therefore, once a proof of concept has been established, an assessment must be performed of the implementation practicality of the system and the compromise between resource usage and system performance. The assessment will help identify the engineering challenges for a space-ready product.

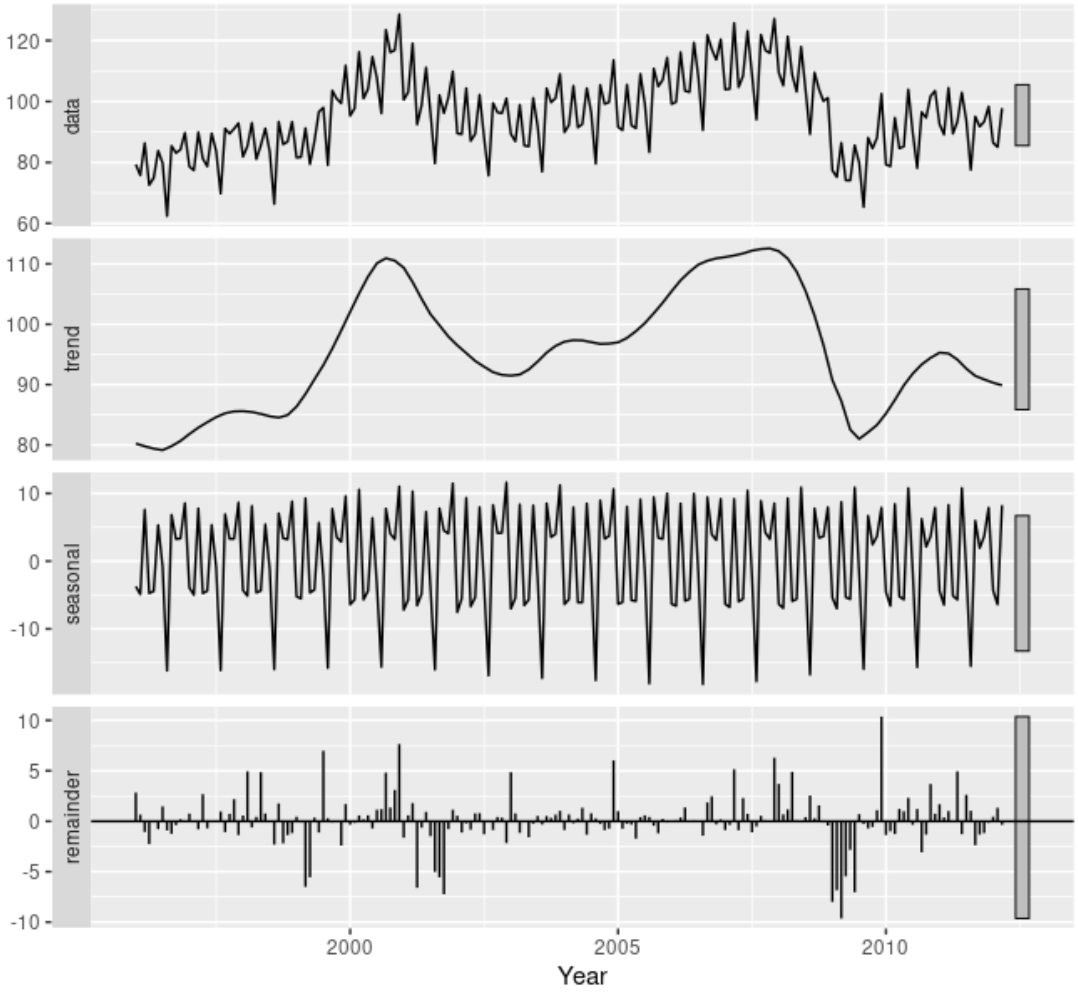

Fig. 8—Time series decomposition of a univariate signal [36]

3.2 Data Collection, Aggregation, and Preparation

While there are a plethora of open-source anomaly detection data sets, there are comparatively fewer telemetry data sets. In an effort to build models on domain-specific data, the data collection effort has included open-source data, NRL-sourced data, and outreach to government and industry. This has yielded multiple telemetry data sets which vary in size, complexity, format, and anomaly type. Government and industry data collection efforts are in progress. In accordance with best practices, all test data is reserved for final evaluation and is not used or viewed in any manner for training purposes. Regardless of the learning paradigm used for training, all test data must contain ground truth information so that a quantitative evaluation of results can be performed.

3.2.1 Open-Source Data

The most promising result of the open-source data search has been an anomaly detection repository from NASA. As part of a project investigating the use of ML in telemetry anomaly detection, a research team from

NASA's Jet Propulsion Laboratory (JPL) has released a data set containing a total of 82 telemetry channels from the Soil Moisture Active Passive (SMAP) and Mars Science Laboratory (MSL) missions [23]. The data contains both point and contextual anomalies. Each telemetry channel contains the telemetry value as well as command information at each time step. An example training telemetry channel from the SMAP spacecraft is shown in Figure 9. No ground truth labels are provided in the training data, meaning that supervised learning methods cannot be used to directly predict anomalies. The test data is labeled with ground truth information providing the indices of any anomalies present in the channel's telemetry values. In addition to the data itself, the developed algorithm was released concurrently and can be used as a baseline performance benchmark. It should be noted that channel A-3 was arbitrarily chosen to serve as a purely developmental data set; that is, both the training and test sets were used to assist development of models, test harnesses, and experiments. A channel from this data set was chosen because it was previously cleaned, normalized, and formatted for model ingestion as a byproduct of NASA's research. Because the test data from this channel has been used multiple times, the channel cannot be used for evaluation. Instead, an evaluation will be performed on the remaining channels.

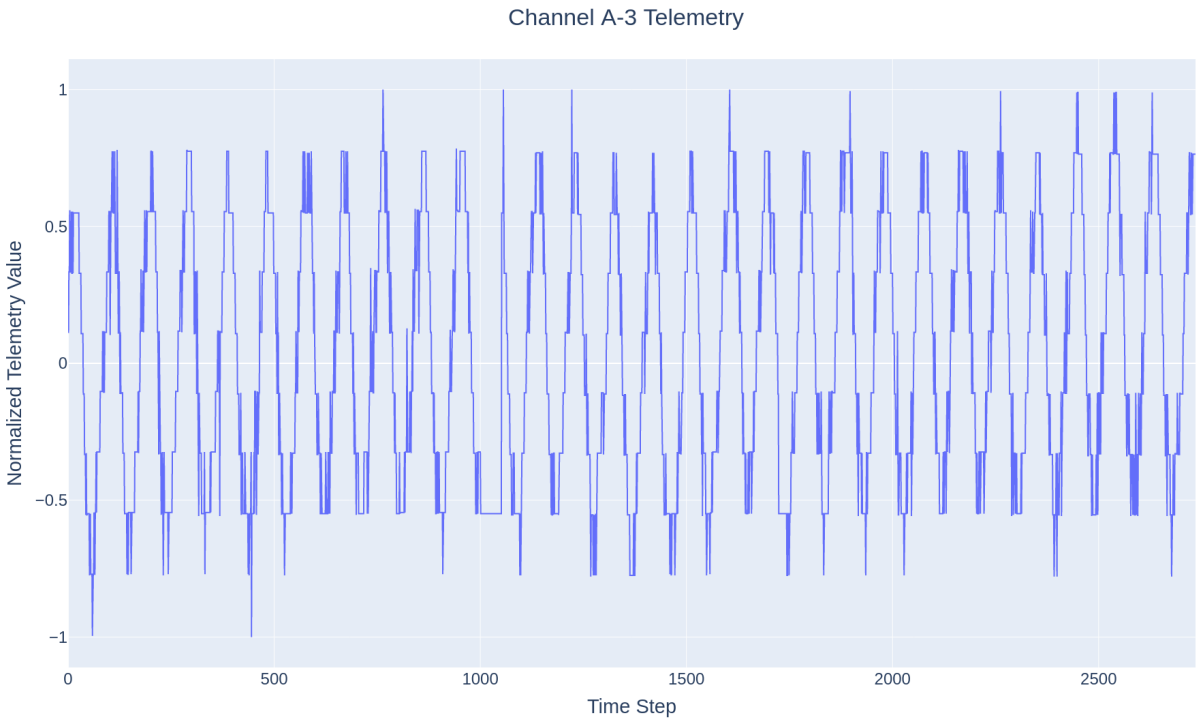

Fig. 9—SMAP A-3 Telemetry Channel Training Data

3.2.2 NRL-Sourced Data

The WindSat payload is the "first fully polarimetric space-borne microwave radiometer" [38]. Launched as part of the Coriolis mission in 2003, the payload provides space-based ocean wind speed and direction measurement. Though the mission is no longer active, the satellite has far exceeded the original mission lifetime of three years and is still operational. The spacecraft is currently operated by NRL's Blossom Point Tracking Facility (BPTF), which records and stores Coriolis telemetry in a records database. The WindSat telemetry database contains a record of multiple years of operational telemetry data. This offers

an excellent opportunity to evaluate algorithms on a real-world data set which, like most anomaly detection problems, contains anomalies as the extreme minority class. A record of known faults can be used during evaluation as ground truth but will not be used for training any algorithms. Final evaluation on this data set can be corroborated by operations engineers at BPTF. In addition to WindSat, several other data collection efforts are ongoing at NRL. As part of various development projects, NCST has developed tools capable of generating telemetry data. These tools can be leveraged to simulate spacecraft operation and inject synthetic fault data; importantly, this offers the opportunity to rapidly create large-scale data sets with variable types and quantities of faults and anomalies. In addition, these tools can be directly used to generate corresponding ground truth data.

3.3 Machine Learning Anomaly Detection Methods

A general definition for anomaly detection offered by [39] is the detection of patterns which "[do] not conform to expected normal behavior." It is an expansive field which is constantly evolving, fueled in large part by IoT big data mining efforts. Machine learning approaches for anomaly detection have become an extremely popular research topic due to their application to tasks such as financial fraud detection, network intrusion detection, medical diagnosis, and industrial health monitoring [40]. Common to all of these domains is the need for automated data processing techniques to assist with the detection of off-nominal patterns in ever-increasing quantities of data. In some contexts, anomaly detection may also be referred to as outlier detection or novelty detection, although these terms have slightly different semantic meaning [41]. Other related problems include concept drift and change detection, which broadly refer to detecting emergent or changing behavior in data. Concept drift and change detection both pose challenges to anomaly detection. The potential for concept drift requires effective algorithms to continuously update their notion of nominal behavior, and a change in data may represent an anomaly or merely the beginning of a new nominal pattern [42].

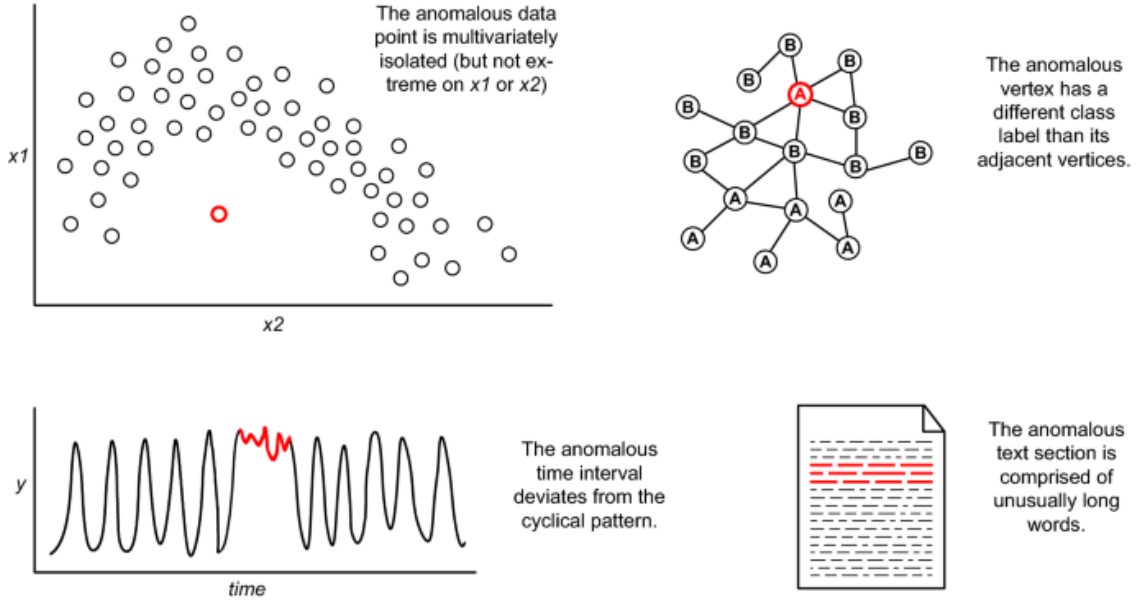

Fig. 10—Anomalies in different types of data [43]

The literature typically specifies three basic anomaly types: point, collective, and contextual. Point anomalies are the simplest and refer to a single data point which is anomalous with respect to the rest of

the data. Collective anomalies refer to groups of data points which together are anomalous, but may not be in isolation. Contextual anomalies are those which cannot be identified without the knowledge of some spatial or temporal context. Both point and collective anomalies can also be framed as contextual anomalies if some form of context was used to identify them [39]. Figure 10 illustrates some examples of anomalies. The top-left and top-right graphs both show point anomalies, and the bottom-left and bottom-right both show collective anomalies. Note that both collective anomalies may also be considered contextual anomalies because they are anomalous in the context of the surrounding data.

There are many challenges facing anomaly detection, especially for applications with large multivariate data spaces. Creating a model which encompasses all possible nominal and off-nominal behavior of a system is difficult, especially when using real-world data which contains noise and often evolves with time. In the context of ML models, procuring training data sets is often difficult or prohibitive due to the limited availability of labeled nominal and off-nominal data [39]. These challenges have led to a variety of approaches across domains. Detailed herein are the types of anomaly detection algorithms which were inspected as part of the literature review. Specific algorithms will not be discussed in detail; rather, classes of algorithms will be discussed for motivational purposes and to highlight potential strengths and weaknesses. This review should not be considered exhaustive due to both the scope restrictions of this research as well as the rapidly evolving nature of the field. For a more comprehensive treatment of anomaly detection techniques and challenges, see [39–41, 44, 45].

3.3.1 Supervised Methods

Supervised approaches for anomaly detection work in much the same way as for other ML tasks. If labeled data is available containing both nominal and anomalous samples, an arbitrary model can be built and trained to make predictions on new input samples. In general, most anomaly detection problems suffer from a severe class imbalance in which the nominal data far exceeds the off-nominal data. For this reason, fully supervised approaches are uncommon because the typical class imbalance prohibits the collection of a comprehensive training data set. In some cases, it is possible to synthetically generate a data set containing nominal and off-nominal samples, though it is generally difficult to synthesize a data set which accurately represents all possible nominal and off-nominal behavior [39]. Partially supervised learning may be useful in support of active or interactive learning schemes which utilize human-in-the-loop feedback to help train the system in a semi-supervised manner [46]. See [47, 48] for in-depth reviews of active and interactive learning.

3.3.2 Semi-Supervised Methods

The classic interpretation of semi-supervised learning is a combination of supervised and unsupervised learning wherein a limited amount of labeled data and a large amount of unlabeled data is used to create a model which has better predictive power than its purely supervised or unsupervised equivalents [49]. In the context of ML anomaly detection, semi-supervised learning often refers to the practice of training a model in a supervised fashion on a single class - nominal or off-nominal - and then using that model to differentiate between the training class and the opposite class [40, 50]. This is most often performed by training a model on nominal data due to the greater availability of nominal data and the difficulty in creating an anomaly data set which contains all possible anomalous behavior.

3.3.3 Self-Supervised Methods

In self-supervised learning, a model is created to solve a "pretext" task which can be formulated as a supervised learning problem using unlabeled data. In this way, the trained model learns a representation of the data itself, which can then be used on a "downstream" task [49]. In the context of anomaly detection, the pretext task is typically to learn a representation of nominal behavior which can then be used to perform the downstream task of differentiating between nominal and off-nominal samples. This is often used in the creation of one-class (i.e., nominal or off-nominal) classifiers.

Prediction

Prediction-based anomaly detection techniques are a form of self-supervised learning which may also be semi-supervised depending on the nature of the input data. The pretext task uses a predictive model to predict a future data point. By characterizing the error between the model and reality - often termed the residual - a downstream technique can be used to identify an anomalous sample by comparing the nominal residual with that of the sample. If the model is not explicitly trained on nominal-only data, it is typically assumed that the off-nominal class is sufficiently rare as to not affect the model's ability to learn nominal behavior [39]. An example of regression-based anomaly detection is shown in Figure 11. The pretext task is the generation of the "expected" curve from the modeled nominal system behavior, and the downstream task is to characterize the nominal residual to determine what level of discrepancy is anomalous [51]. Regression-based anomaly detection is best suited to problems for which the residual exhibits clearly different characteristics for nominal and off-nominal samples. Intuitively, the approach is not well-suited to systems which cannot be reliably modeled, as is the case for highly stochastic or otherwise unpredictable systems.

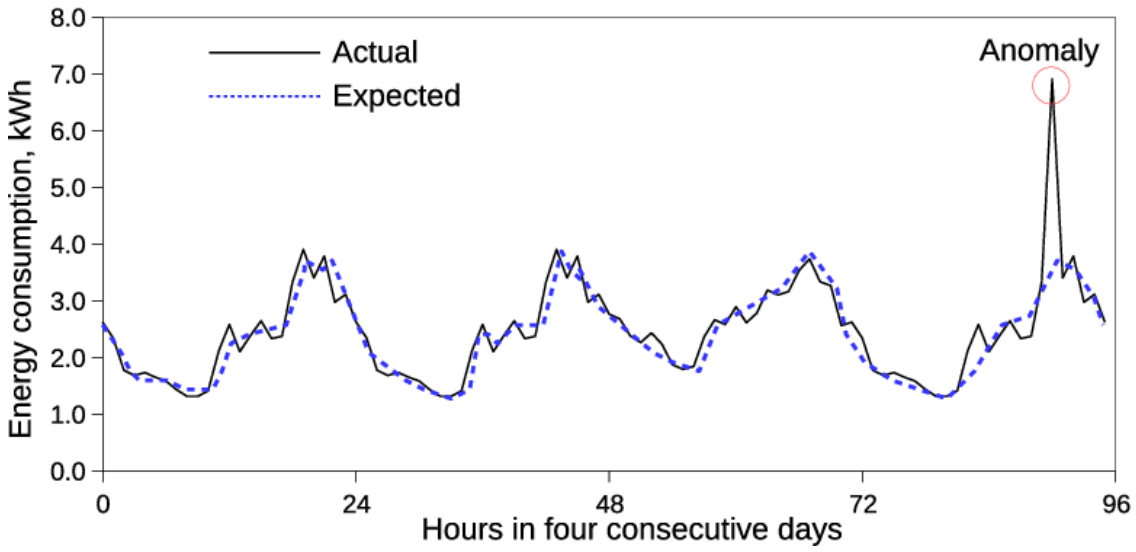

Fig. 11—Prediction-based anomaly detection: large discrepancies between predicted and actual behavior suggest the presence of an anomaly [51]

Reconstruction

Reconstruction models seek to learn a representation of the input data by forcing the model to reconstruct the input from a latent representation [52]. The most common version is under-complete reconstruction

wherein the model must reconstruct the input from a compressed, lower-dimension latent representation. This is in contrast to over-complete models whose latent space is larger than the input. In the context of anomaly detection, reconstruction techniques typically use under-complete models. This is based on the observation that in order to form the best possible reconstruction from a limited latent representation, a model must learn only the most relevant attributes of the data, ignoring irrelevant or erroneous information. In this way, it learns a representation of nominal behavior. When applied to anomaly detection, the method operates on the presumption that if trained to learn a nominal representation of behavior, the model will have low reconstruction errors on nominal data and high errors on off-nominal data. Then, in a manner similar to that of other semi-supervised approaches, some downstream technique(s) can be used to identify anomalies. Modern reconstruction models are most often implemented as neural networks and are referred to in the literature as auto-encoders because they can be viewed as a special case of the general encoder-decoder network architecture. A general representation of an under-complete auto-encoder is shown in Figure 12. Replicator networks have been extensively studied in the literature and are often the basis for a variety of complex anomaly detection techniques such as adversarial auto-encoders and Generative Adversarial Networks (GANs) [53, 54].

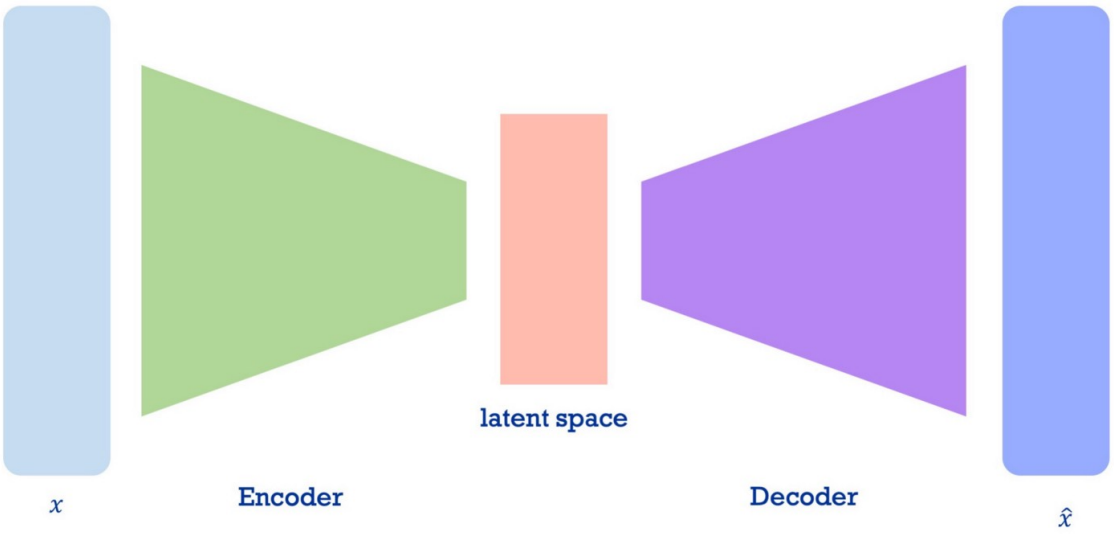

Fig. 12—An under-complete auto-encoder, where $\hat{x}$ is the reconstruction of input x [55]

3.3.4 Unsupervised Methods

Statistics

While many statistical anomaly detection techniques do not strictly belong to the field of ML, their ubiquity in anomaly detection tasks bears mention. In multi-step anomaly detection pipelines they are often used as one of the final processing steps. Statistical tests may be used to determine the anomaly scores themselves, and they may also be used to determine an estimate of confidence in those scores. Statistical anomaly detection relies on the assumption that nominal and off-nominal data exists in the high probability and low probability regions of a stochastic model, respectively. The types of approaches can be segmented into parametric techniques which make an assumption about the distribution of data, and non-parametric techniques which make no assumption about the underlying distribution. A popular parametric approach is to assume a Gaussian distribution of data so that a variety of techniques - such as the Z-score pictured

in Figure 13 - can be used to obtain an anomaly score for the data samples. Due to empirical success, the assumption of normality is often made even when it does not hold in reality. The largest advantage and disadvantage of statistical techniques is the assumption that the data fits an arbitrary distribution. If the assumption holds true, then the results are statistically justifiable and easy to interpret; however, if it does not, the technique may yield invalid or erroneous results. For a review of statistical anomaly detection and formal statistical outlier detection, see [39, 56].

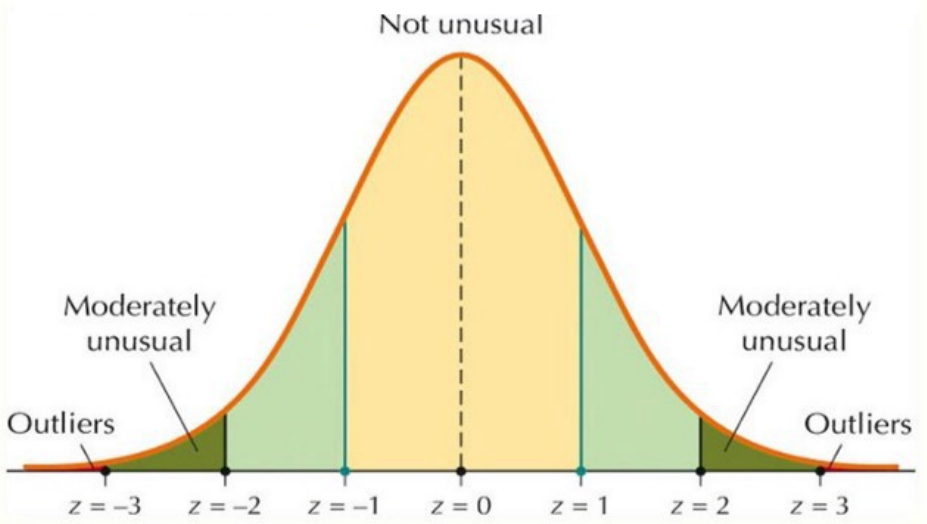

Fig. 13—The Z-score can be used to find outliers in Gaussian-distributed data [57]

Clustering and Neighborhood-Based

Though they are distinct classes of techniques, clustering and neighborhood approaches for anomaly detection both work on the premise that anomalous data can be identified by some distance metric which distinguishes it from nominal data. Clustering approaches make an assumption that the nominal and off-nominal data spaces can be grouped into a number of well-defined clusters. Neighborhood approaches detect anomalies by calculating their relative distance or density with respect to a local neighborhood. While the learning itself is often performed in an unsupervised fashion, the overall approach may often be considered semi-supervised because of the implicit assumption that effectively all of the training data is nominal. Alternatively, some approaches also try to form explicit clusters for anomalies. Due to the fact that the pretext task for these approaches is often a straightforward application of a preexisting unsupervised technique, a large variety of clustering, neighborhood, and density-based anomaly detection algorithms exist. For an overview, see [39].

Isolation-Based

Whereas many anomaly detection methods work by first profiling nominal behavior and subsequently using that profile to discriminate between nominal and off-nominal samples, isolation-based anomaly detection methods take a fundamentally different approach by explicitly attempting to isolate anomalies from the rest of the data. Intuitively, this approach can be explained by the observation that an anomaly should be significantly easier to separate from the rest of the data than a nominal data point. The original ML isolation-based anomaly detector is the Isolation Forest (IF or iForest) algorithm, which has served as the motivating basis

for related approaches such as Extended Isolation Forest (EIF) and Half-Space-Trees (HS-Trees) [58–60]. The IF algorithm works by building an ensemble of random decision trees. Each tree makes random splits in the data until every data point is isolated (i.e., separated from the rest of the data). The average path length - the number of splits required to isolate the data point, averaged across all trees - is used as the basis for determining an anomaly score. Empirically, anomalous points have a markedly shorter path length. The EIF algorithm removes an inherent bias present in the original IF algorithm. Half-Space-Trees (HS-Trees) take a partitioning approach similar to IF and can be applied to streaming data. A representation of an Isolation Forest is shown in Figure 14.

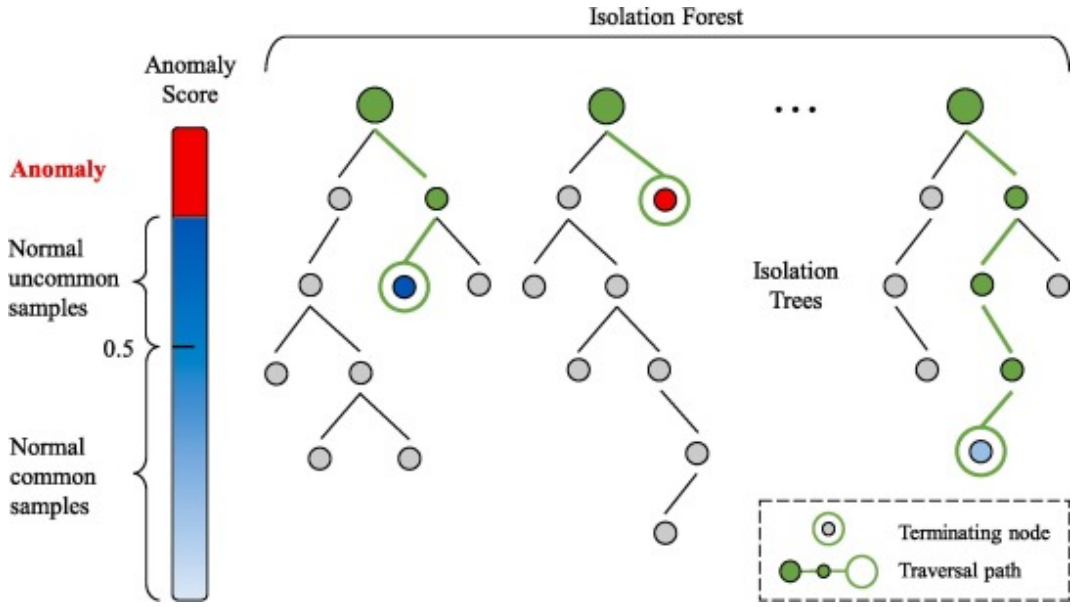

Fig. 14—The Isolation Forest uses isolation path length as the basis for anomaly detection [61]

3.3.5 Other Approaches

Various other techniques for anomaly detection have been proposed. Information-theoretic techniques operate on the assumption that anomalies can be identified by measuring their contribution to the information complexity of a data set. Spectral techniques try to find anomalies by projecting the data onto a subspace of a different dimension where anomalies are more easily found. Some versions of replicator and adversarial models implicitly perform this type of data projection during the data compression phase. More recent contributions to the field include the application of deep reinforcement learning to the active semi-supervised anomaly detection approach [62].

3.3.6 Combination Methods

It should be noted that various algorithms may be used as constituents for an ensemble method or a multi-step algorithm. For instance, the classical ensemble approach can be applied to anomaly detection by combining the predictions of a number of different models in an effort to create a composite prediction which is better than any of its constituents. Additionally, some techniques lend themselves to the creation of a pipeline of algorithms, wherein the output of a given step is used as the input to another. For example, a ML model may be used to perform representation learning such as feature extraction or dimensionality reduction,

and the output of the model may be used in a subsequent step for further processing or determination of anomaly scores. This is a common processing technique for methods which operate on complex high-dimensional data and for approaches which utilize ML as a pre-processing step in the overall detection pipeline.

3.3.7 Considerations for Spacecraft

The nature of the spacecraft telemetry anomaly detection problem imposes certain restrictions on the algorithms which may be used. Telemetry is typically time series data, meaning that temporal context is important when determining anomalies. Intuitively, telemetry values which are normal at one point in time may be abnormal in another. Therefore, algorithms which do not incorporate temporal information may be disadvantaged. Some methods incorporate temporal information as a byproduct of the model used, such as those based on Long Short-Term Memory (LSTM). Other methods attempt to extend non-temporal algorithms by applying a sliding window across the sequence. Additionally, telemetry is typically high-dimensional and multivariate. For example, a thermal control system may be comprised of a number of telemetry channels for multiple other subsystems. In order to form a comprehensive view of the system, some level of system hierarchy and abstraction is needed. Therefore, algorithms which are limited to univariate data are more limited in their applicability then their multivariate counterparts.

Learning paradigm is also a primary consideration, especially for spacecraft telemetry. For most anomaly detection problems, examples of nominal behavior far exceed off-nominal examples. This is especially true for spacecraft. Because spacecraft faults and anomalies are typically quite rare, it is often implausible to use supervised learning to directly train a model. Thus far, approaches which operate in a semi-supervised or unsupervised fashion have empirically shown better performance for tasks which do not have large amounts of labeled data. While semi-supervised and self-supervised approaches are far more common, they are typically not capable of improving detection capability over time through the use of explicit feedback on whether a given detection was correct. In the longer term, active learning and other associated paradigms may allow these methods to incrementally improve via a small number of expert-labeled examples.

3.4 Evaluation Principles

Evaluation of an integrated system requires the ability to quantify both the system's overall performance as well as the individual contribution of each constituent element. This is a difficult task for FM systems in large part due to the V&V challenges which already exist in the field. In general it is not feasible to predict all possible failure modes for a complex system, and therefore all analyses and tests are intrinsically non-exhaustive [26]. Within the more narrow scope of telemetry monitoring and anomaly detection, it is possible to simplify the evaluation problem through a comparison with other baseline telemetry monitoring methods which are already used in practice. In the simplest case, a baseline method can be directly compared with one or more ML methods. However, in the event that the ML component is integrated into a larger system, it becomes necessary to demonstrate its specific contribution to the overall system.

3.4.1 Evaluation of Anomaly Detection Algorithms

For each input sample, the final output of an anomaly detection algorithm typically takes one of two forms: an anomaly score measuring how anomalous the sample is, or a binary label which classifies the sample into either the nominal or off-nominal class. In general, scores are regarded as a more flexible and informative approach because they allow for a continuous spectrum of anomaly levels rather than two

discrete states. Scores can also be converted to binary labels at any time by applying a threshold. However, it becomes difficult in practice to directly compare algorithms which use differing scoring methods because the scores themselves are often derived from fundamentally different concepts and assumptions. Additionally, choosing a threshold for converting scores to labels can be challenging and often relies on a domain-specific assumption about the data [39]. These challenges make it critical to establish well-defined evaluation criteria and explicitly state any assumptions which may affect the results.

Careful selection of metrics is needed to ensure a complete and objective assessment. The typical class imbalance inherent to the anomaly detection problem means that many traditional measures such as accuracy, F-Measure, Average Precision (AVPR), and Area Under the Receiver Operating Characteristic curve (AUROC) may artificially inflate performance [63]. One such example of this is binary accuracy; a binary classifier which naively assigns each and every sample in a given data set to the nominal class will likely score relatively well due to the high class imbalance [64]. It is commonplace to use the binary True-False-Positive-Negative (TFPN) metrics - True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) - as a basis for evaluation because they can be used to compute a multitude of other derived metrics. A confusion matrix is a common way of using the TFPN metrics to characterize overall performance both visually and numerically. An example layout of an anomaly detection confusion matrix is shown in Figure 15. Each sample in a given evaluation data set is assigned to one of the four quadrants, and the totals from each quadrant are used to determine performance. The general consensus in the literature is that while some values such as the Matthews correlation coefficient are more objective than others, no single quantity derived from the TFPN metrics is capable of a complete and objective characterization. Additional consideration must be taken when comparing results across data sets which may have different statistical qualities. In general, metrics with known biases may be used only if their bias can be explicitly quantified and noted in the analysis. For an in-depth discussion of two-class classification metrics and their associated strengths and weaknesses, see [63–66].

Anomaly detection in time series data is further complicated by the temporal dependence between data samples. Notably, the classic confusion matrix does not account for a time dimension, so while it may offer a global view of performance, it provides no insight into how a given algorithm performs locally in time [67]. This is an inherent drawback because time series data is often highly non-stationary and as a result a model's performance may differ drastically depending on temporal context. Additionally, the presence of contextual and collective anomalies spanning multiple points in time leads to ambiguity regarding what constitutes a "hit" for each of the TFPN metrics. Numerous approaches have been offered in the literature; for example, a predicted detection which partially overlaps with a collective anomaly window may be scored as a true positive, a false negative, or some combination of the two [23]. In general, application-specific definitions of the TFPN metrics should be created which are tailored to the priorities of the detection problem [65, 67].

3.5 Software Products

A number of software packages are being developed which facilitate ML and anomaly detection research. Python was chosen as the main development language due to its open-source licensing, ease of development, flexibility, and preexisting support and infrastructure for ML and data science. Though it has been developed to support this research, every reasonable effort has been made to create software which is highly modular and supports a well-documented Application Programming Interface (API) such that it may be easily extended to other similar or related problems.

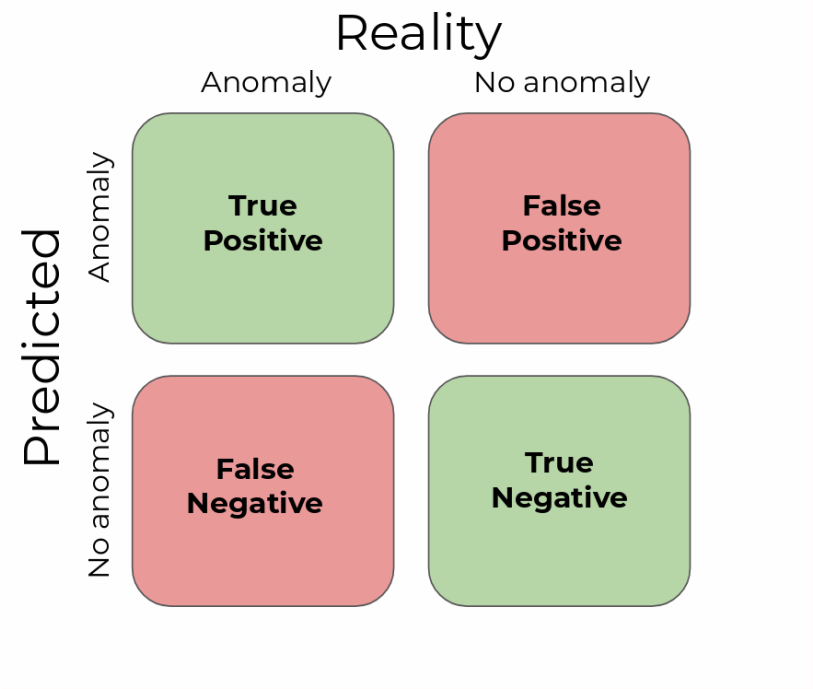

Fig. 15—Anomaly detection confusion matrix

3.5.1 General Software Tools

Several tools have been created in support of this research which may be broadly applicable outside the domains of anomaly detection or ML. Tools identified as such during development have been moved to separate repositories where they can be developed, tested, and released as standalone products either within the NRL community or as open-source software. A Python utilities repository has been created which hosts various common convenience and utility functions. For larger-scale tools, dedicated repositories have been created. Currently, the two main tools are an ensemble builder and a statistical analysis repository.

Ensemble Builder

Ensemble ML methods form a composite model by combining the predictions of multiple individual models. This is done in an effort to produce composite predictions better than that of any constituent model. The ensemble builder developed as part of this code base is API-agnostic and can combine predictions from arbitrary numbers and types of models, allowing for cross-library combinations. Batch processing is supported for large models or data sets. It also supports model deactivation functionality so that the contribution of any given model to the overall ensemble can be easily determined. While most common prediction combination schemes are pre-implemented and readily available, it is possible to implement a custom combination strategy if necessary.

Statistical Tools

Exploratory Data Analysis is a common initial step in many ML projects. Several tools were created to help automate the EDA process, including computation and plotting of data set statistics, as well as statistical

report generation. The reporting tool can also compare the statistics of multiple sets of data; this can be used to detect concept drift within a single data stream or to compare the characteristics of different data sets. In terms of anomaly detection, many algorithms assume a Gaussian distribution of data to justify statistical conclusions about the nature and quantity of anomalies in a given data set. To test the validity of this assumption, a tool was developed which performs a Kolmogorov-Smirnov test on a sample of data to determine whether the assumption of normality is valid. Because the test can support any continuous distribution, the tool was extended to all continuous distributions in the Scipy package [68]. The tool can be used to quickly fit over 100 candidate statistical distributions to a sample of data to determine which distributions, if any, are plausible fits of the data.

3.5.2 Anomaly Detection Code Base

The main software development effort has consisted of the creation of an anomaly detection code base which serves as host to a large number of anomaly detection algorithms as well as experimentation and evaluation tools. Where possible, open-source implementations of algorithms were leveraged to reduce development time. Otherwise, algorithms were manually implemented as needed. Certain types of models such as neural networks allow for extensive customization and architecture tuning; for these, model-building tools were created to allow for rapid model creation and testing. Currently, the code base supports over 50 anomaly detection models, including open-source models from the PyOD [69] and PySAD [70] libraries, individually-released open-source algorithms, and custom models. Custom-implemented models include traditional and variational auto-encoders based on the LSTM architecture. Models and algorithms will continue to be added to the repository as needed.

Packages such as TensorFlow, Keras, and Scikit-Learn each host an expansive set of metrics which can be used interchangeably within the anomaly detection code base through the use of API-agnostic interfaces [71–73]. For simple evaluations, stateless metrics can be used. For larger data sets, the code base supports stateful metrics which can update in batches. Custom metrics can be derived from these or implemented as needed for evaluation. In addition to metrics, an evaluation and ranking tool was created which is able to automatically compare the performance of an arbitrary number of algorithms on a given problem. The evaluation process is similar to and inspired by AutoML, which automatically tunes a number of models to their best possible performance on the training data and subsequently evaluates them on the test data [74].

A suite of plotting tools has been developed in tandem with the rest of the code base on top of the Plotly library [75]. The plotting module serves a number of functions for development, demonstration, and deployment. Firstly, it aids Exploratory Data Analysis (EDA) as well as algorithm development and debugging by providing data visualization. Secondly, it allows for presentation and analysis of detection results. Thirdly, it is intended to allow for real-time data and algorithm monitoring in an application setting. An ongoing effort is the creation of a near real-time dashboard which displays streaming data as well as the detection of any anomalies. This dashboard may be used for human-in-the-loop feedback-based learning, operational monitoring, and technology demonstrations.

4. CONCLUSION

Autonomous capability development continues to be an investment priority for the DoD and the U.S. Navy. Health monitoring and FM are some of the most significant challenges standing in the way of more resilient, reliable, and autonomous Navy space systems, and automated anomaly detection represents a single

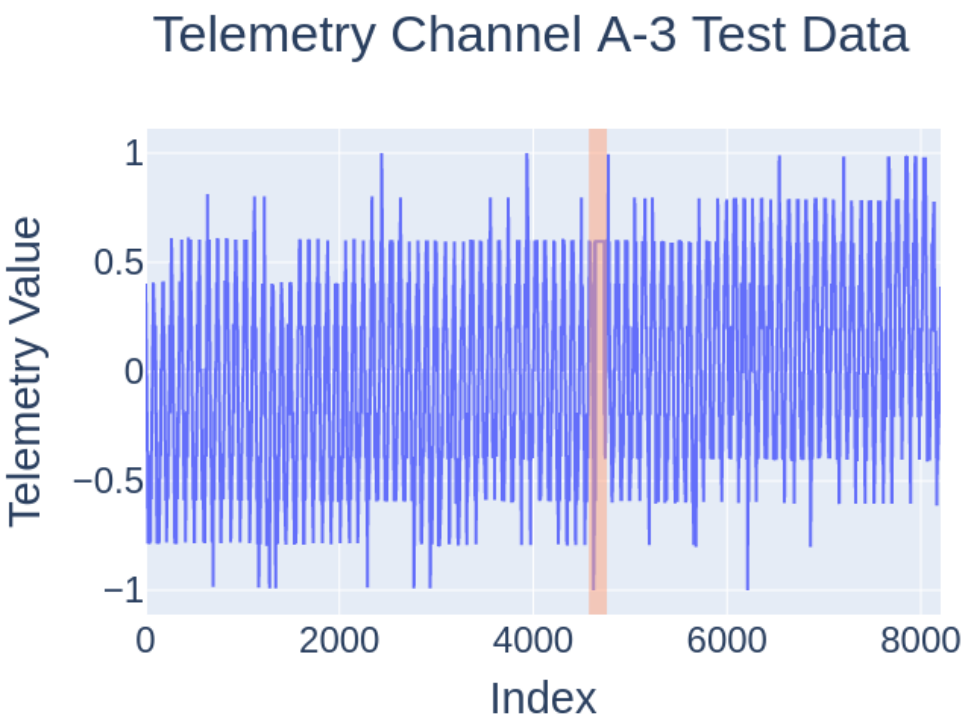

Fig. 16—Anomaly plotter highlighting the SMAP A-3 channel test anomaly

step toward that end. The first year of research established that ML methods offer an opportunity to rapidly improve existing spacecraft anomaly detection efforts, and may potentially be applied to other areas of system health monitoring. The second year of research will focus on the the proof of concept through the continued development of the anomaly detection code base and the completion of experiments evaluating various algorithms on telemetry data. In addition, integrated model approaches will be further investigated and compared with data-driven methods. Health monitoring represents one of the most promising applications of ML to U.S. Navy space system operations. This type of technology is mature enough to be extended to existing systems, and stands to benefit the near-term Navy's ground infrastructure and the future Navy's onboard satellite systems.

REFERENCES

1. V. Kothari, E. Liberis, and N. D. Lane, "The Final Frontier: Deep Learning in Space," 2020.
2. "Naval Research and Development Framework," 2017.
3. J. Mattis, "Summary of the 2018 national defense strategy of the United States of America," 2018.
4. D. S. Board, "Task Force Report: The Role of Autonomy in DoD Systems," 7 2012.
5. "Unmanned Systems Integrated Roadmap: 2017-2042," 2017.
6. T. Fong, "Autonomous systems: Nasa capability overview (2018).

7. R. Mukai, Z. Towfic, M. Danos, M. Shihabi, and D. Bell, "MSL Telecom Automated Anomaly Detection," 2020.

8. "Science Technology Strategy for Intelligent Autonomous Systems," 7 2021.

9. J. A. Starek, B. A(C) cıkme(C) se, I. A. Nesnas, and M. Pavone, "Spacecraft Autonomy Challenges for Next-Generation Space Missions," 2016.

10. A. Jónsson, R. A. Morris, and L. Pedersen, "Autonomy in space: Current capabilities and future challenge," *AI magazine* **28**, 27 (2007).

11. W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff, "Autonomous and autonomic systems: a paradigm for future space exploration missions," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* **36** (5 2006), ISSN 1094-6977, doi:10.1109/TSMCC.2006.871600.

12. G. Cancro, "APL Spacecraft Autonomy: Then, Now, and Tomorrow," *Johns Hopkins Apl Technical Digest* **29**, 226–233 (10 2010).

13. "The difference between Artificial Intelligence, Machine Learning and deep learning." URL `https://www.intel.com/content/www/us/en/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html`.

14. M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning* (MIT press, 2018).

15. D. Parikh, "Learning paradigms in machine learning," 7 2018. URL `https://medium.datadriveninvestor.com/learning-paradigms-in-machine-learning-146ebf8b5943`.

16. C. T. Chaplain, "Space Acquisitions: DOD Faces Significant Challenges as it Seeks to Address Threats and Accelerate Space Programs, Statement of Cristina T. Chaplain, Director, Contracting and National Security Acquisitions, Testimony Before the Subcommittee on Strategic Forces, Committee on Armed Services, House of Representatives, 2019.

17. S. Caldwell, "State-of-the-art of Small Spacecraft Technology," Oct 2021. URL `https://www.nasa.gov/smallsat-institute/sst-soa`.

18. C. J. Dennehy and L. M. Fesq, "The Development of NASA's Fault Management Handbook (2011).

19. L. Fesq, "Fault Management Handbook," 2012.

20. S. Terry, "What are Space Ground Systems?," 3 2021. URL `https://ai-solutions.com/newsroom/about-us/news-multimedia/what-are-space-ground-systems/`.

21. M. Tipaldi and B. Bruenjes, "Spacecraft health monitoring and management systems (IEEE Computer Society), 2014, pp. 68–72. ISBN 9781479920693, doi:10.1109/MetroAeroSpace.2014.6865896.

22. M. L. Hanson, L. M. Fesq, and M. H. Nguyen, "In situ method and system for autonomous fault detection, isolation and recovery," 10 2000.

23. K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding (Association for Computing Machinery), 7 2018, pp. 387–395. ISBN 9781450355520, doi:10.1145/3219819.3219845.

24. S. Fuertes, G. Picart, J. Y. Tourneret, L. Chaari, A. Ferrari, and C. Richard, "Improving spacecraft health monitoring with automatic anomaly detection techniques (American Institute of Aeronautics and Astronautics Inc, AIAA), 2016. ISBN 9781624104268, doi:10.2514/6.2016-2430.

25. P. Jackson, "Introduction to expert systems (1986).

26. J. Day and M. Ingham, "Fault management at JPL: past, present and future (2011).

27. M. E. Newhousel, J. McDougal, K. S. Bryan Barley, and L. M. Fesg, "Results from the NASA Spacecraft Fault Management Workshop: Cost Drivers for Deep Space Missions.

28. J. Cleveland and B. Cassidy, "Neptune Software Introduction (The Aerospace Corporation), 3 2015.

29. K. Kolcio and L. Fesq, "Model-based off-nominal state isolation and detection system for autonomous fault management (IEEE), 3 2016. ISBN 978-1-4673-7676-1, doi:10.1109/AERO.2016.7500793.

30. S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mechanical Systems and Signal Processing* **107** (7 2018), ISSN 08883270, doi:10.1016/j.ymssp.2017.11.024.

31. M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," 1 2016, ISSN 10958592.

32. D. Bernard, R. Doyle, E. Riedel, N. Rouquette, J. Wyatt, M. Lowry, and P. Nayak, "Autonomy and software technology on NASA's Deep Space One," *IEEE Intelligent Systems* **14** (5 1999), ISSN 1094-7167, doi:10.1109/5254.769876.

33. T. Yairi, Y. Kawahara, R. Fujimaki, Y. Sato, and K. Machida, "Telemetry-mining: a machine learning approach to anomaly detection and fault diagnosis for space systems, 2006, pp. 8 pp.–476. doi:10.1109/SMC-IT.2006.79.

34. M. M. Fernandez, Y. Yue, and R. Weber, "Telemetry anomaly detection system using machine learning to streamline mission operations, volume 2017-December (Institute of Electrical and Electronics Engineers Inc.), 12 2017, pp. 70–75. ISBN 9781538634622, doi:10.1109/SMC-IT.2017.19.

35. H. Ahn, D. Jung, and H. L. Choi, "Deep generative models-based anomaly detection for spacecraft control systems," *Sensors (Switzerland)* **20** (4 2020), ISSN 14248220, doi:10.3390/s20071991.

36. R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice* (OTexts, 2018).

37. S. Roffe and A. D. George, "Evaluation of Algorithm-Based Fault Tolerance for Machine Learning and Computer Vision under Neutron Radiation, 2020, pp. 1–9. doi:10.1109/AERO47225.2020.9172799.

38. D. McKinney, "Dr. Peter Gaiser honored with Navy Meritorious Civilian Service Award for windsat," Jul 2013. URL `https://www.nrl.navy.mil/Media/News/Article/2562653/dr-peter-gaiser-honored-with-navy-meritorious-civilian-service-award-for-windsat/`.

39. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)* **41**, 1–58 (2009).

40. G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys* **54**, 1–38 (3 2021), ISSN 0360-0300, doi:10.1145/3439950. URL `https://dl.acm.org/doi/10.1145/3439950`.

41. R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey (1 2019). URL http://arxiv.org/abs/1901.03407.

42. S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing* **262** (11 2017), ISSN 09252312, doi:10.1016/j.neucom.2017.04.070.

43. R. Foorthuis, "On the nature and types of anomalies: a review of deviations in data," *International Journal of Data Science and Analytics* **12** (10 2021), ISSN 2364-415X, doi:10.1007/s41060-021-00265-1.

44. H. Wang, M. J. Bah, and M. Hammad, "Progress in Outlier Detection Techniques: A Survey," *IEEE Access* **7** (2019), ISSN 2169-3536, doi:10.1109/ACCESS.2019.2932769.

45. S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song, "Anomalous Example Detection in Deep Learning: A Survey," 2021.

46. N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward Supervised Anomaly Detection," *Journal of Artificial Intelligence Research (JAIR)* **45** (11 2012), doi:10.1613/jair.3623.

47. B. Settles, "Active Learning Literature Survey," 2009.

48. J. A. Fails and D. R. Olsen, "Interactive machine learning (ACM Press), 2003. ISBN 1581135866, doi:10.1145/604045.604056.

49. X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4L: Self-Supervised Semi-Supervised Learning," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* pp. 1476–1485 (2019).

50. L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K. R. Müller, and M. Kloft, "Deep Semi-Supervised Anomaly Detection," 2020.

51. X. Liu and P. S. Nielsen, "Regression-based Online Anomaly Detection for Smart Grid Data," *ArXiv* **abs/1606.05781** (2016).

52. Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," 2014.

53. S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised Anomaly Detection via Adversarial Training," 2019.

54. A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks," 2020.

55. P. Janetzky, "Generative networks: From ae to vae to gan to cyclegan," 7 2021. URL https://towardsdatascience.com/generative-networks-from-ae-to-vae-to-gan-to-cyclegan-b21ba99ab8d6.

56. N. A. Heckert, J. J. Filliben, C. M. Croarkin, B. Hembree, W. F. Guthrie, P. Tobias, J. Prinz, et al., "Handbook 151: NIST/SEMATECH e-Handbook of Statistical Methods (2002).

57. H. Bonthu, "Detecting and treating outliers: How to handle outliers," 5 2021. URL https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/.

58. F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest (IEEE), 12 2008. ISBN 978-0-7695-3502-9, doi:10.1109/ICDM.2008.17.

59. S. Hariri, M. C. Kind, and R. J. Brunner, "Extended Isolation Forest," *IEEE Transactions on Knowledge and Data Engineering* **33**, 1479–1489 (4 2021), ISSN 2326-3865, doi:10.1109/tkde.2019.2947676. URL `http://dx.doi.org/10.1109/TKDE.2019.2947676`.

60. S. C. Tan, K. M. Ting, and T. F. Liu, "Fast anomaly detection for streaming data, 2011.

61. H. Chen, H. Ma, X. Chu, and D. Xue, "Anomaly detection and critical attributes identification for products with multiple operating conditions based on isolation forest," *Advanced Engineering Informatics* **46** (10 2020), ISSN 14740346, doi:10.1016/j.aei.2020.101139.

62. D. Zha, K. H. Lai, M. Wan, and X. B. Hu, "Meta-AAD: Active Anomaly Detection with Deep Reinforcement Learning," *2020 IEEE International Conference on Data Mining (ICDM)* pp. 771–780 (2020).

63. D. Fourure, M. U. Javaid, N. Posocco, and S. Tihon, "Anomaly Detection: How to Artificially Increase your F1-Score with a Biased Evaluation Protocol, 2021, pp. 3–18.

64. P. Branco, L. Torgo, and R. Ribeiro, "A survey of predictive modelling under imbalanced distributions," *arXiv preprint arXiv:1505.01658* (2015).

65. D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061* (2020).

66. J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves, 2006, pp. 233–240.

67. G. Kovács, G. Sebestyen, and A. Hangan, "Evaluation metrics for anomaly detection algorithms in time-series," *Acta Universitatis Sapientiae, Informatica* **11** (12 2019), ISSN 2066-7760, doi:10.2478/ausi-2019-0008.

68. P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods* **17**, 261–272 (2020), doi:10.1038/s41592-019-0686-2.

69. Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *arXiv preprint arXiv:1901.01588* (2019).

70. S. F. Yilmaz and S. S. Kozat, "PySAD: A Streaming Anomaly Detection Framework in Python," *arXiv preprint arXiv:2009.02572* (2020).

71. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. URL `https://www.tensorflow.org/`.

72. F. Chollet et al., "Keras," 2015.

73. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research* **12**, 2825–2830 (2011).

74. E. LeDell and S. Poirier, "H2O AutoML: Scalable Automatic Machine Learning," *7th ICML Workshop on Automated Machine Learning (AutoML)* (July 2020). URL `https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf`.

75. P. T. Inc., "Collaborative data science," 2015. URL `https://plot.ly`.

This page intentionally left blank

Appendix A

ADDITIONAL APPLICATIONS OF ANOMALY DETECTION

The first year of research exposed several applications of anomaly detection to spacecraft operations which extend beyond spacecraft FM. Though they have not been explored in-depth thus far, the Navy can benefit from automated anomaly detection in multiple domains; one such application is the detection of both nominal and off-nominal behavior as part of the Navy's C4ISR efforts. As data processing needs continue to expand, it becomes increasingly important for automated methods to be applied. In the near term, automated anomaly detection can serve as a preliminary step in the C4ISR data processing pipeline, assisting human operators by distilling massive amounts of data down to only the data which bears further investigation. This may be applied to signal of interest detection in communications, situational awareness efforts, and environmental monitoring. All of these applications will help shorten the latency between the collection of raw data and the availability of actionable information.

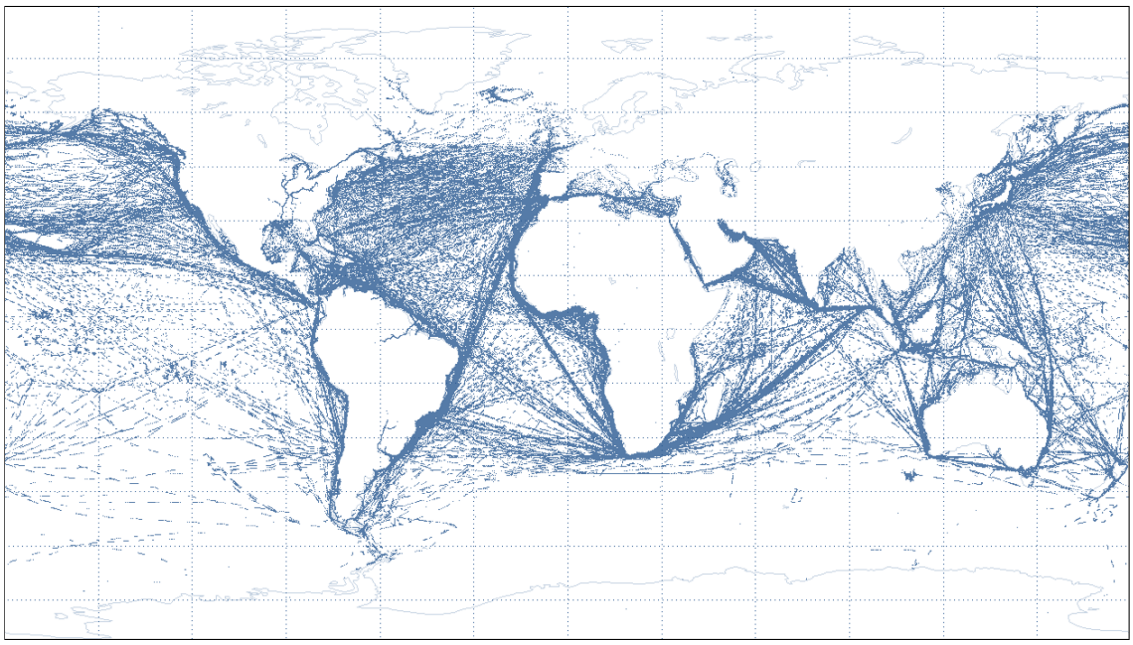

Fig. A1—Global AIS data

As a motivating example, consider that anomaly detection algorithms may assist with ML-based MDA efforts by modeling normal vessel behavior and flagging anomalous behavior for further analysis. Given that millions of vessels must be accounted for by the U.S. Navy, there exists a substantial opportunity for anomaly detection to drastically reduce the amount of data which must be inspected by humans. Figure A1 shows an example of global Automatic Identification System (AIS) data which may be used as the basis for training models. In the near term, vessel track data such as that from NRL's Sea-Link Advanced Analysis (S2A) system can be used to model normal vessel behavior and help reduce the processing load for human operators. In the longer term, these models may work alongside humans and provide advanced decision-making insight via learned pattern recognition.

Acronyms

AI Artificial Intelligence.

AIS Automatic Identification System.

API Application Programming Interface.

BPTF Blossom Point Tracking Facility.

C4ISR Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance.

DoD Department of Defense.

DSB Defense Science Board.

EDA Exploratory Data Analysis.

FDIR Fault Detection, Isolation, and Recovery/Response.

FM Fault Management.

FMEA Failure Mode and Effects Analysis.

IoT Internet of Things.

ISHM Integrated System Health Management.

LSTM Long Short-Term Memory.

MBSE Model-Based Systems Engineering.

MDA Maritime Domain Awareness.

ML Machine Learning.

NASA National Aeronautics and Space Administration.

NCST Naval Center for Space Technology.

NDS National Defense Strategy.

NRL Naval Research Laboratory.

OOL Out-Of-Limits.

PNT Positioning, Navigation, and Timing.

TFPN True-False-Positive-Negative.

V&V Verification and Validation.