BEHAVIORAL MODEL ANOMALY DETECTION IN AUTOMATIC IDENTIFICATION SYSTEMS (AIS)

By

Jacob Coleman

Farah Kandah
Professor of Computer Science
(Chair)

Anthony Skjellum
Professor of Computer Science
(Committee Member)

Craig Tanis
Professor of Computer Science
(Committee Member)

BEHAVIORAL MODEL ANOMALY DETECTION IN AUTOMATIC IDENTIFICATION SYSTEMS (AIS)

By

Jacob Coleman

A Thesis Submitted to the Faculty of the University of Tennessee at Chattanooga in Partial Fulfillment of the Requirements of the Degree of Master of Science: Computer Science

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

May 2020

ABSTRACT

Over 90% of all goods in the world, at some point in their life, are on a vessel at sea. Currently, the maritime industry relies on the Automatic Identification System (AIS) for collision avoidance and vessel tracking. AIS is an unencrypted, unauthenticated protocol that is vulnerable to various types of cyber attacks allowing malicious actors to alter the location of vessels. With the advent of the Ocean of Things (OoT), vessels are sharing more information than vessel location alone at sea. Increasingly, more information is becoming critical for safe and efficient operation at sea. This thesis presents a novel approach of applying machine learning to build vessel behavior models that exploit AIS information. These models will allow vessels to detect anomalous communication from vessels nearby. This will enable vessels to determine the quality of the message shared between each other and, more critically, identify malicious actors.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Farah Kandah, for his guidance through each stage of the process. I want to thank Dr. Anthony Skjellum and Dr. Craig Tanis, for being on my thesis committee and providing helpful feedback.

Second, I would especially like to thank my wife, Leora, for her support and countless sacrifices to help me get to this point. To my parents, Dale and Bonnie Coleman, and my brother and sister in law, Joshua and Amanda, for their continued support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AIS, Automatic Identification System

ATON, Aid to Navigation

COG, Course Over Ground

CPA, Closet Point of Approach

DBSCAN, Density-Based Spatial Clustering of Applications with Noise

DBSCANSD, Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction

DoS, Denial of Service

EE, Elliptic Envelope

GMM, Gaussian Mixture Model

IBC, Identify-Based Cryptography

iForest, Isolation Forest

IID, Identically Distributed

IMO, International Maritime Organization

IOS, Inter-Organizational Information Systems

iTree, Isolation Tree

KDE, Kernel Density Estimator

LOF, Local Outlier Factor

LSTM, Long Short-Term Memory

MANET, Mobile Ad-hoc NETworks

MiTM, Man in The Middle

ML, Machine Learning

MMSI, Maritime Mobile Service Identity

NEMA, National Marine Electronics Association

OoT, Ocean of Things

p2p, Peer to Peer

PKI, Public Key Infrastructure

SO-TDMA, Self Organizing Time Division Multiple Access

SOG, Speed Over Ground

SOLAS, Safety of Life at Sea

SQL, Structured Query Language

SVM, Support Vector Machine

TREAD, Traffic Route Extraction for Anomaly Detection

VHF, Very High Frequency

CHAPTER 1

Introduction

Over 90% of the world's trade is carried by vessels at sea [9]. The maritime industry is looking forward to the future Smart Ocean to provide reduced operating costs, while simultaneously increasing crew safety. The Smart Ocean will consist of a large number of connected devices comprising the Ocean of Things (OoT) [10, 11]. Currently, vessels communicate via the Automatic Identification System (AIS). AIS allows vessels to identify themselves, their direction, and speed to other vessels within a 10 to 20-mile range of the transmitting vessel. AIS is essential to collision avoidance and the safe operation of vessels. AIS uses a plain-text peer-to-peer (p2p) form of communication that can easily be modified or spoofed to transmit false information [12]. AIS currently assists a vessel's crew with navigation and identification of nearby vessels [13]. Because of the lack of authentication, AIS is susceptible to various modifications. To increase confidence in AIS readings, and help vessels have greater assurance while at sea, machine learning can be applied to model normal vessel behavior. This process allows vessels that report abnormal information to be identified.

1.1 Background

This thesis examines the intersection of machine learning and maritime AIS. The background includes information on the Automatic Identification System and how it is used in vessel to vessel communication, along with vessel to shore communication. Along with AIS, an outline of four different machine learning methods useful in anomaly detection will be discussed in this thesis. The four machine learning methods are: isolation forest, local outlier factor, support vector machine, and robust covariance elliptic envelope.

1.1.1 Automatic Identification System (AIS)

Currently, the maritime industry tracks vessels at sea through the Automatic Identification System (AIS). The 2002 International Maritime Organization (IMO) Safety of Life at Sea (SOLAS) requires vessels over 300 gross tonnages to be equipped with AIS [1]. A popular online vessel tracking application, currently is tracking over 166,000 vessels close to shore using AIS, with over 400,000 estimated installments, and up to 1,000,000, once fully deployed globally [13]. The SOLAS requirement accounts for the widespread adoption of AIS in the maritime industry.

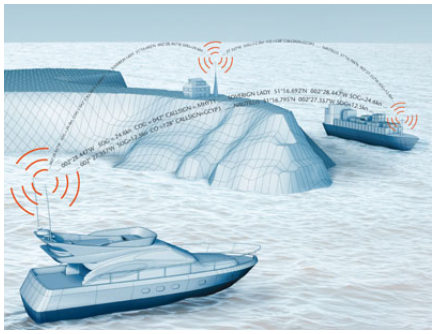


Figure 1.1 AIS communication between vessels and shore side stations [1]

AIS is a Very High Frequency (VHF) Self Organizing Time Division Multiple Access (SO-TDMO) protocol [2] allowing vessels to share National Marine Electronics Association (NMEA) messages.

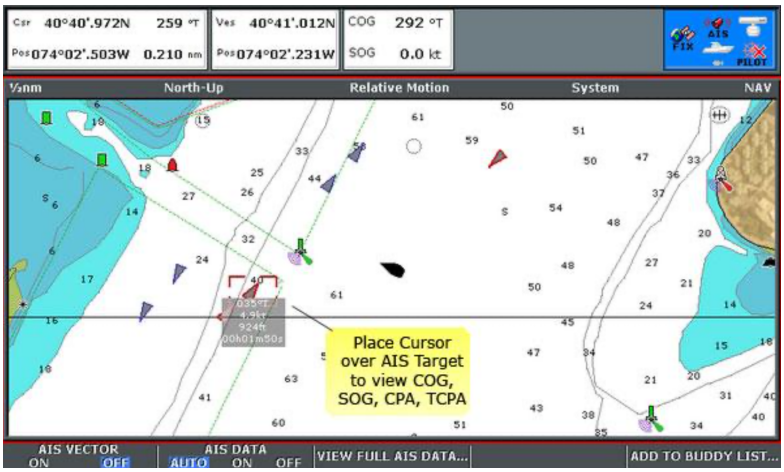


Figure 1.2 AIS display used on vessels showing nearby vessels [2].

AIS messages include, but are not limited to, latitude; longitude; speed over ground; course over ground; position accuracy; timestamp; Maritime Mobile Service Identity (MMSI) number; true heading; type of ship; name; dimensions of ship; draught of ship; and destination.

AIS distinguishes between vessels, aids to navigation (ATON), and vessel traffic service (VTS). These messages occur through 2,250 time slots per second on two VHF frequencies, AIS-1 161.975 MHz and AIS-2 162.025 MHz. Using both channels AIS-1 and AIS-2, 4,500 slots are available per second. Each time slot is 26.67 ms long with a message size of 256 bits per message.

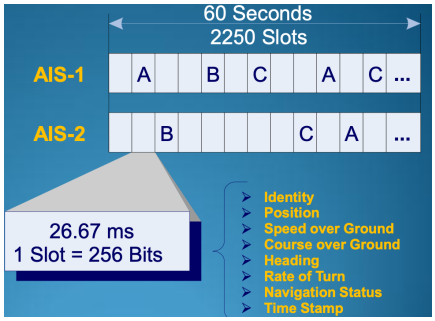


Figure 1.3 AIS protocol on two frequencies with 2,250 time slots on each channel [2].

AIS is an unencrypted, unauthenticated protocol [1]. This protocol allows various types of attacks [12] in AIS to occur. One could assume that applying encryption to AIS could solve or improve AIS. This suggestion has been made with many obstacles to adoption on a global scale across many nationalities and economic zones [7].

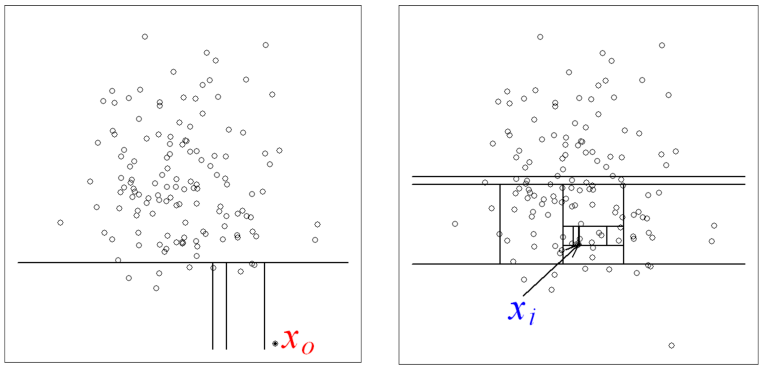
1.1.2 Machine Learning

Machine learning is the process of utilizing computational methods using past information to make accurate predictions. Machine learning can assists with many types of tasks including text classification, natural language processing, computer vision, and problems such as fraud detection. Classification is a problem well-suited for machine learning. Classification consists of assigning a category to an item based on past information [14].

One type of machine learning classification is determining if an item is a member of a group or not a member. This type of classification is anomaly detection. Ansecombe and Guttman [15] define statistical anomalous behavior detection as "An observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed." Historical normal kinematic behavior data is assumed to be Independent and Identically Distributed (IID). Once collected, historical observations are compared to new movement data with a statistical inference test applied to determine if a new observation belongs to the model or not.

A few different types of machine learning anomaly detection methods are isolation forest, local outlier factor, support vector machine, and elliptic envelope.

Isolation Forest [3] (iForest) focuses on isolating anomalies instead of normal profiling points. This is achieved by building isolation trees (iTree) for a given data set. Anomalies are those instances that have short average path lengths in an Isolation Tree. The benefit of this method is with a reduced subsampling size, a high detection performance can be achieved with high efficiency. Isolation Forest's key difference from standard profiling methods is that it is not distance-based, nor density-based, which reduces the computational overhead needed to calculate the cost of distance or density. The time complexity of iForest is linear, with low constant and low memory requirements. iForest can scale to handle considerable data sizes and high dimensional problems with large numbers of irrelevant attributes.



(a) iForest Outlier (b) iForest Inlier

Figure 1.4 Isolation Forest adapted from [3] of the number of partitions needed to detect an inlier.

Anomalies are identified by a short path length (Figure 1.4). Given 135 points, a normal point, Figure 1.4b, requires twelve random partitions for isolation. Outliers, (Figure 1.4a), require only 4 partitions to be isolated.

Local Outlier Factor (LOF) [4] utilizes k-nearest neighbor through k-distance. LOF considers the relative density of observations and can detect both local and global outliers for skewed datasets. LOF identifies an outlier as a Hawkins-Outlier which is, "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism [4]." LOF calculates the k-distance of observations as any positive integer k between any objects within a given data set. From these distances, a k-neighborhood is identified for a given k-distance of observation; the k-distance neighborhood of an observation contains every object whose distance from observation is not greater than the k-distance. When an observation is identified as outside the local neighborhood, it is considered an outlier. Using LOF, multiple neighborhoods can be identified in a data set so that both global and local outliers can be isolated.

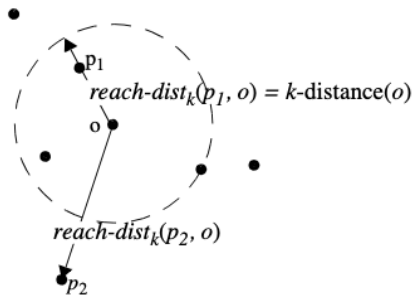


Figure 1.5 K-distance between points used to identify a point [4].

Support Vector Machine (SVM) is an implementation of a Support Vector Network (SVN) by Cortes and Vapnik [5]. The SVM maps the input vectors into a dimensional feature space Z through a non-linear mapping chosen for the data set. Support vectors are smaller training data set points near the edge of the data set that constructs a hyperplane. Using clean error-free training data to create support vectors allows error detection by the ratio between the expected value of the number of support vectors and the number of training vectors. Hyperplanes are constructed well when the classifier has the most significant distance to the nearest training vector.

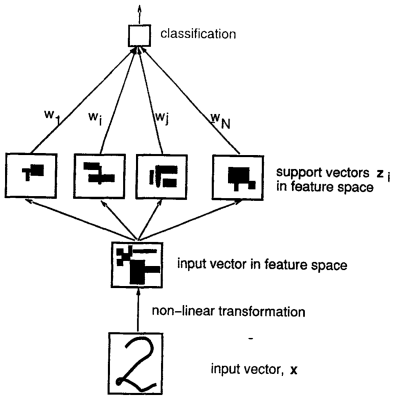


Figure 1.6 Support Vector Machine (SVM) example classifying a number two from a drawing [5].

When a Robust Covariance Elliptic Envelope [6] detects a single outlier, the distance between normal observations and an outlier is quite easy to detect by the Mahalanobis distance. However, this method suffers from multiple outliers by a masking effect. The masking effect occurs when a classifier selects a model that is sub-optimum, masking that a better model could have been selected. A solution to the masking effect is distance based on robust estimators of multivariate location and scatter. The classifier selects the normal observations as an elliptic envelope around the data for all inliers. Outliers are all data points outside the elliptic envelope.

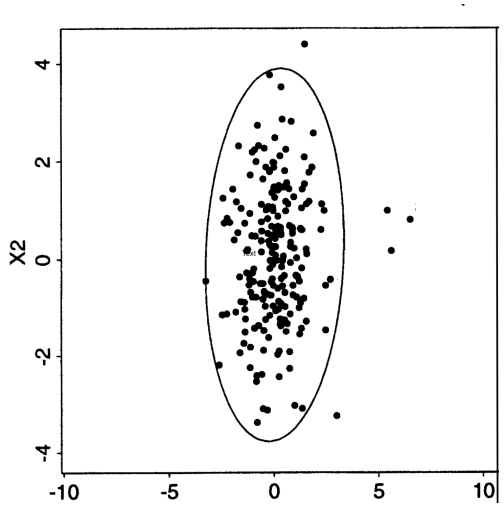


Figure 1.7 Elliptic Envelope adapted from [6] of inlier classification from the elliptic shape.

1.2 Outline

After the above stated background is a literature review of existing work in maritime AIS, using machine learning along with the type of vulnerabilities in AIS. The background is followed by a discussion of the problem statement about AIS, then our approach of using machine learning to model vessel behavior using AIS is next. After demonstrating how behavioral models are built, a numerical analysis presents the results of testing our method against specific use cases followed by a discussion of the results. Finally, a discussion including future work and the conclusion.

CHAPTER 2

Literature Review

The literature review examines existing machine learning approaches using AIS data and the type of possible attacks against AIS. The majority of machine learning approaches focus on self-reported vessel trajectory based on the received AIS information. The second section focuses on reviewing the types of AIS attacks that can occur.

2.1 Machine Learning in AIS

Liang, *et al.* [16] propose a two-step Long Short-Term Memory (LSTM) supervised learning method to reconstruct a vessel's trajectory when AIS location data is lost. AIS allots 4,500-time slots per minute, in a congested region, an AIS transceiver becomes starved for resources due to a lack of available time slots to transmit on. When this occurs, missing AIS data creates a gap in information for the location of a vessel. Missing AIS data can also happen in inclement weather. As the signal drops, the information is lost after transmission. This allows those monitoring a vessel's movement to project more accurately the ship's prior location, to better understand the ship's previous and future movements.

Sidibe, *et al.* [17] survey techniques to identify anomalous behavior in the maritime domain using AIS. They categorize the detection methods based on three categories: statistical, machine learning, and data-mining.

Data-driven approaches are created using a two-phase method. First, a vessel's normal behavior is modeled based on historical data. Second, the learned model is applied to current vessel movement data with any differentiation considered anomalous behavior.

Anneken, *et al.* [18] uses Gaussian Mixture Model (GMM) and Kernel Density Estimator (KDE) to predict anomalies incurred a high rate of false alarms. Gaussian Process and Active Learning were used, but at the cost of high computational complexity in training models. Bayesian Networks have been trained to account for AIS data, combined with real-world contextual data, such as weather and time with vessel interactions.

Pallotta, *et al.* [19] identifies currently, point-based anomalous behavior and trajectory-based anomalous behavior detection approaches. These two methods focus on the location of the vessel's travel, either where the vessel currently is located or the trajectory of the vessel.

Pallotta, *et al.* [20] use a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Wang, *et al.* [21] Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction (DBSCANSD). Both utilize Hadoop MapReduce clusters using parallel meta-learning algorithms. The authors note that the algorithm improves linearly in time and accuracy with more nodes in the Hadoop cluster. These algorithms are of time complexity of $O(n^2)$ for n number of data points in the training set. Traffic Route Extraction for Anomaly Detection (TREAD) [19] is used to learn vessel routes from AIS data to predict the vessel's future position. Location-based approaches account for speed, either increased or decreased, based on historical movement. They also detect anomalous heading for off-route vessels that normally follow an existing route.

Sidibe,*et al.* [17] note that anomalous vessel behavior detection causes a high rate of false-positive anomalies detected.

Data-Mining methods seek to improve upon the high false-positive rates of trajectory and point-based methods.

Osekowska, *et al.* propose one such approach, [22] by developing and modeling traffic as a potential field for the geographic tracks that a vessel moves through at sea. The field is stronger with greater amounts of vessel traffic and weaker with less traffic. The field has three properties: strength, decay, and distribution. The field strength increases with greater traffic. As fewer vessels traverse a path, the path decays, and the strength value decreases. Distribution is the distance between two points and is described by a two-dimensional Gaussian smoothing, using Euclidean distance between two points. In this system, a vessel whose current position is detected outside the local potential field is marked as anomalous.

Soleimani, *et al.* [23] propose a geometrical method based on the vessel trajectory for the vessel's near-optimal path. A near-optimal path is generated using a graph search algorithm. If a vessel departs from the near-optimal path, then the unanticipated movement of a vessel generates an abnormality score.

Roy, *et al.* [24] generate alerts based on rules in ports of known port parameters. Parameters known are the maximum speed allowed in a port and marked restricted areas within a port. If the parameters are broken, then a vessel is marked as anomalous.

The approaches discussed above consider only the vessel's location (latitude, longitude), Speed Over Ground (SOG), and Course Over Ground (COG). Sidibe *et al.* [17] identify two significant issues with current approaches to AIS anomaly detection. First, is the question of availability, with high computational complexity, and the need to apply this globally. Second, is that vessels can switch the AIS transceiver off so that vessel movement occurs off-line, creating gaps in vessel movement.

Sidibe, *et al.* note the scarcity of literature on anomalous vessel behavior using big data and real-time processing techniques from kinematic attributes and static characteristics of vessel behavior.

Hanyang, *et al.* [25] develop a method to detect anomalously vessel trajectories over space-based AIS (S-AIS). A newer technique in AIS is the use of space-based satellite monitoring of the

Earth. Vessel to vessel AIS range is typically 20 nautical miles. AIS receivers on-shore can be placed higher and have a slightly higher range of 35-40 nautical miles. The newer, space-based AIS (S-AIS) range is much greater and, in clusters, can cover the entire globe. In stage one, data pre-processing occurs by calculating the standard deviation of each vessel's single day Speed Over Ground (SOG) and Course Over Ground (COG) and applying logarithmic normalization. In stage two, the Elbow Rule is applied to find the best number of clusters, then K-means is used to cluster vessels.

2.2 AIS Attacks

Mazzarella, *et al.* [26] investigate the detection of and AIS signal from the vessels crew by examining the signal strength of an AIS transmission. AIS is dependent on the system, receiving a continuous stream of data, without interruption or intervention from the vessel crew. The crew has used AIS, switching on or off, to hide the location of vessels. If a signal is strong or weak, and a signal changes status, it can be an indicator that human intervention has occurred. Mazzarella *et al.* account for electromagnetic propagation conditions of fixed AIS base stations to determine the conditions under which the AIS signal strength should alternate due to physical conditions without human intervention. Mazzarella *et al.* address the issue of detecting anomalies, presuming a user is acting to conceal the location of a vessel using the physical properties of electromagnetic propagation.

Balduzzi, *et al.* [12] detail the various type of AIS attacks and categorize them into two categories: first, implementation-specific in software; second, protocol-specific in the AIS radio transponders. At the software layer, one could spoof another vessel's Maritime Mobile Service Identity (MMSI) and pose as another vessel. Spoofing would make the vessel broadcasting appear to be another vessel, along with spoofing the location of the vessel one is broadcasting as. Spoofing as another vessel could also allow one to program a malicious route so that a vessel appears to

have taken a false route. Software attacks occur when attacking the application layer based on the applications used by various systems that log AIS messages. An example of this is a port authority. If a port authority logs messages from AIS in a SQL database, one could craft a message to enter into the SQL database executing arbitrary code through AIS.

For radio attacks, one can alter the message broadcast by a physical vessel. This allows one to modify the location in real-time of vessels in transit. A type of attack is a man-in-water Spoofing. An S.O.S is sent, then received by nearby vessels, compeling them by regulation to attempt a rescue. Simulating an S.O.S would allow an attacker to lure a victim vessel to a hostile location.

Closest Point of Approach (CPA) triggers a collision warning alert, encouraging a vessel to alter course to avoid a collision. One can spoof a vessel's location so that it appears close to a vessel and the trajectory indicating a collision will occur. This will trigger an alarm on the victim's vessel that a collision is imminent.

Frequency Hopping (DoS++) can occur by an attacker spoofing as a port authority. This forces the vessel's transponder to a non-default frequency and masks the transponder to other vessels operating nearby. This would render a ship invisible to other vessels nearby on AIS.

Slot Starvation (DoS++) occurs when a base station, such as a port authority, exhausts all available slots for message broadcasting. A base station has a high priority compared to vessels. Spoofing as a base station, one can book the next 100 milliseconds and then another 100 milliseconds continuously, so that all slots are continuously taken, barring any legitimate vessel's messages from being broadcast.

Timing Attack (DoS++) instructs an AIS transponder to delay its transmission for a period in time. One can broadcast continuously, causing an AIS transponder to delay transmission, essentially disabling the transponder continuously. One can also change the transponder to transmit more often and flood all messages for a given region.

Hardware Panic (DoS) attacks saturate the channel's electromagnetic spectrum with copious quantities of noise. Based on the hardware, malfunctions can occur at the recipient's memory or processor, which could be overloaded.

As the maritime industry moves toward autonomous shipping, researchers are utilizing AIS, the existing ship to ship communication and tracking technology. AIS is designed for ship avoidance and self-broadcasting of vessel information. The AIS protocol is neither encrypted, nor authenticated. Current public key infrastructure (PKI) models applied to the international maritime industry have been developed and are in public use without widespread adoption. A possible reason for this is the global international scope required to develop and maintain such a system. Goudossis, *et al.* [7] suggest an Identity-Based Cryptography and Symmetric Cryptography (IBC) enhance AIS security. Identity-Based schemes have also been suggested for aviation for its Automatic Dependent Surveillance-Broadcast (ADS-B).

A nefarious type of attack is the deliberate switching-off of AIS by vessel operators to conceal a vessel's location. Reasons for this type of action include: following official guidelines in dangerous waters, violation of regulation to conceal the location of high value passengers.

Based on the vulnerabilities in AIS Goudossis *et al.* propose [7] these needed enhancements to AIS security. Figure 2.1 demonstrates how a target could be tracked with AIS.

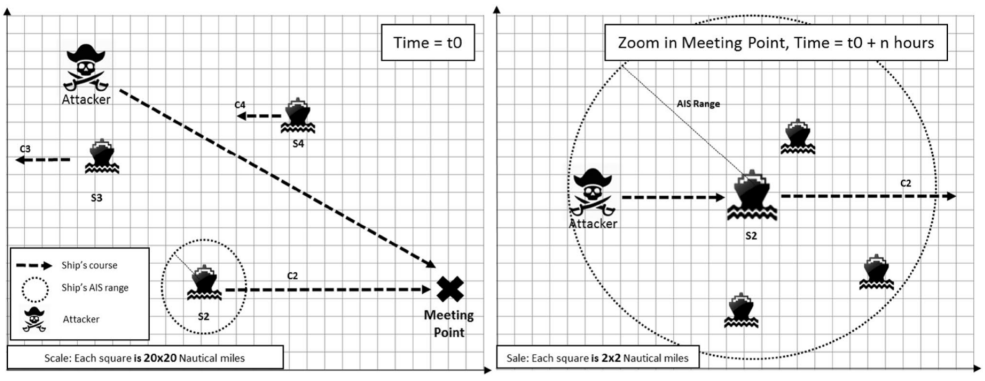


Figure 2.1 AIS Attacks illustrated on a plot of louring a vessel using AIS [7].

1. Confidentiality AIS-broadcast messages should be encrypted.

2. Privacy and anonymity upon request where confidentiality offers some protection from threats against privacy, full privacy and anonymity of a vessel upon request need to be offered, to address authenticated adversaries. Thus, it must be possible to prove that even an anonymous vessel is an authorized and legitimate one; and that some non-repudiation capability must exist even for anonymous vessels.

3. Message source authentication and data integrity AIS-broadcast messages must be authenticated.

4. Non-repudiation of AIS messages.

5. Completeness, simplicity, and feasibility. Finally, yet importantly, the approach for a security-enhanced AIS must be complete and feasible in the complex maritime domain where AIS is a productive system, on-board the majority of vessels around the globe today. A proposed solution for AIS needs to be flexible for use by crews, simple, widely acceptable, easy to integrate, and financially affordable.

Goudossis, *et al.* suggest [7] a tiered scheme of Ad-hoc networks. Tier one is the IMO at the international level; tier two is the individual nations and flag states; tier three is each individual vessel creating its own network. The vessel level network is a type of Mobile Ad-hoc NETworks (MANETs). This type of network is dynamic as the network changes over time as the vessel moves around the globe.

Goudossis, *et al.* also propose [7] Identity-Based Cryptography and Symmetric Cryptography (IBC) to simplify the keying infrastructure, which derives the public key from a vessel's distinctive attributes and a private key created by IMO. The IMO would create keys and delegate private key generation for vessels under their flag's registration. In addition to secured AIS, a cryptographic implementation is needed to assign public and private keys for secure communication.

2.3 Maritime Digitalization

Sanchez-Gonzalez, *et al.* [8] report on the current status of the digitalization of the Maritime Industry. They define the difference between digitization and digitalization. Digitization is the process of changing from an analog to a digital format, digitalization is "the use of digital technologies to change business model and provide new revenue and value-producing opportunities, that is, the process of entering a digital business [8]."

The leading countries in maritime transport digitization studies are China, Korea, and Spain, with Europe producing over 50% of all research studies. Figure 2.2 shows the number of papers by country. Figure 2.3 shows the increase in interest to raise per year.

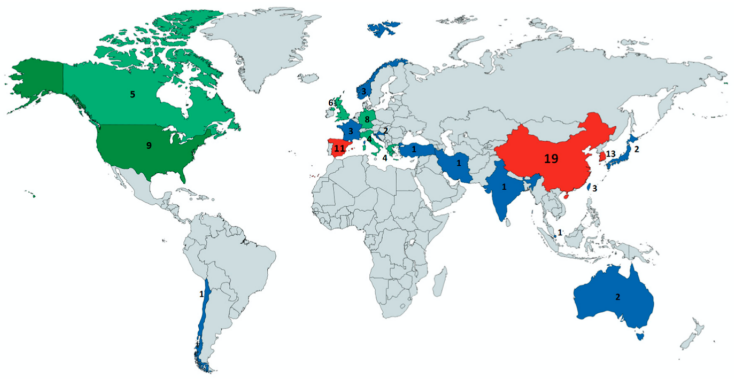


Figure 2.2 Map of Maritime Digitization Studies by Country [8].

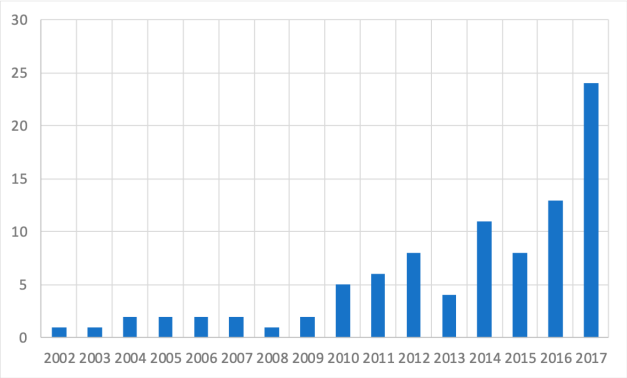


Figure 2.3 Graph of Maritime Digitization Studies by Year [8].

The maritime transport industry is in early-stage digitalization compared to other sectors, such as financial services, retail, and land transport. An example of this study is a study on the fundamental aspects of digitization for inter-organizational information systems (IOS). The results of this study showed that over 70% of land transport companies used an IOS, while less than 25% of maritime transport companies used an IOS.

A significant factor in the need to digitalize the maritime industry is the rapid growth of 40% within a single decade between the years 2005 and 2015. This growth occurred, even during a global economic downturn.

CHAPTER 3

Motivation and Contribution

Vessels at sea are susceptible to various attacks via AIS. Machine learning can help mitigate this exposure to provide a safer operating environment. Currently, ships do not share sensor data. As autonomous shipping and the OoT increases in scope [10, 11], vessels can, and will, be sharing more information. The method proposed is a novel approach of applying machine learning to vessel sensor data to detect anomalous behavior. By sharing sensor data, vessels can increase awareness and monitor each other.

3.1 Motivation

The issues surrounding AIS include the lack of encryption and authentication. AIS broadcasts a message in plain text without verifying the sender or the content. A secure form of AIS exists, but has not been widely adopted. Even with a secure version, the maritime industry has a long cycle for the adoption and integration of new technology [8]. Besides slow adoption, many geo-political issues exit to forming a global encryption key distribution mechanism. With this in mind, a more localized approach is needed that integrates with the existing AIS protocol, allowing vessels at sea to classify each message to determine message authenticity or if a vessel is acting abnormally. By using machine learning to model a vessel's normal behavior, when a vessel is observed to be acting abnormally, it can be identified more easily.

3.2 Problem Statement

The current maritime AIS vessel communication lack of encryption and authentication, leaving the protocol and vessels susceptible to many types of manipulation. The Automatic Identification System is the primary means for vessel self-identification at sea. Since the protocol is plain text and unauthenticated, any transmitter can broadcast any information, including fraudulent information. There is no mechanism in AIS to check the validity of a message or the sender of a message. AIS transmissions assume that any information broadcast is true, with no checks in place.

3.3 Contribution

In this thesis our contribution is as follows:

1. Developing a machine learning anomaly detection method for vessels at sea.
2. Analyzing different machine learning methods to select the best method of securing AIS transmission accuracy.
3. Design multiple use cases that challenge the behavioral model built for vessels operating at sea. We present results to determine which model produces the best results.

This method examines and demonstrates machine learning behavior modeling using temperature sensors. Temperature sensors are the first step in demonstrating effectiveness. Once a model is fit to normal observations, classification occurs. Classification consists of identifying the validity of a message. With sufficient messages, a model is generated, determining whether a vessel is broadcasting normally or abnormally.

As vessels operate within a given region, ships monitor messages to determine if the vessel and its messages are normal or should be abnormal. Machine learning modeling, coupled with AIS, would allow ships to track communications received via direct and indirect methods. The more a

vessel interacts with another vessel, it would have a higher internal scale to rank the communication received. With a shift in the industry toward autonomous shipping, ensuring that vessels are sharing reliable information is crucial. Erroneous (or even nefarious) data could be both financially costly and harmful to those interacting with these vessels.

Machine learning can be applied to model vessel messages to classify a vessel's message as an inlier or outlier. The adoption of machine learning in maritime is increasing and can provide many benefits. Machine Learning models can be trained to classify [24] normal observations to detect new messages or anomalies that deviate from normal operations. Most machine learning research has focused on AIS tracking to determine vessel behavior [23]. This thesis' novel approach to cross-checking AIS data demonstrates that machine learning behavior modeling can be applied to vessels at sea to increase confidence in AIS.

3.4 Threat Models

3.4.1 Impersonation

AIS plain text messages are susceptible to various attacks [12]. Many of the attacks can be limited by identifying spoofing attacks where an attacker fraudulently poses as another vessel. Slot Starvation attacks, can occur, if one impersonates a base station, when the attacker is not a base station. Frequency Hopping attacks, can occur, if one impersonates a port authority, when the attacker is not a port authority. Closest Point of Approach attacks originate from a false collision being triggered by an attacker impersonating as another vessel.

3.4.2 Selective Transmission

Another type of attack is a vessel intentionally turning AIS off [17] to conceal a vessel's location for a period of time and then turning AIS back on again when it is advantageous. This can be dangerous as vessels are then operating without broadcasting their location to other vessels nearby.

3.4.3 Model Manipulation

Once a model is fit, it becomes susceptible to attack as nefarious actors attempt to manipulate the model used to classify observations on-board a vessel. Model manipulation attacks try to play the system and determine locations where classification could be weak, allowing invalid information not to be classified correctly. One such type of attack is "breakout fraud", where an attacker maintains a good rating for a period and then starts injecting invalid information [27].

CHAPTER 4

Behavioral Model Anomaly Detection Methodology

This chapter presents a new approach of applying machine learning to model vessel behavior in real-time. The chapter details the Behavioral Model Anomaly Detection approach, to determining if an AIS message is within a reasonable range of previous communication values, using machine learning. This approach is composed of three primary phases: Model Induction, Progressive Analysis, and Model Re-Induction. Model Induction consists of collecting initial readings for new vessels and aggregating all collected data at a port and building machine learning models of normal behavior. Progressive Analysis is the distribution of trained models from a port to vessels that use these pre-built models offline to analyze messages from know vessels used in model induction. Model Re-Induction occurs as vessels operate together in a common geographic location often. This chapter introduces this novel approach first, followed by a detailed outline of each step of the process.

4.1 Behavioral Model Anomaly Detection Introduction

When vessels are at sea, it is common for a vessel to be offline and without a reliable internet connection. Vessels at sea communicate with each other using AIS. AIS is unsecured, unencrypted, and lacks any integrity check. These weaknesses lend AIS to various vulnerabilities. AIS is deployed on hundreds of thousands of vessels worldwide.

It is proposed that machine learning can be used to build vessel behavior models to assists AIS. By using machine learning models to monitor various AIS data points, anomalies behaviour

can be detected. To increase confidence in the models built, a shared information approach is used such that vessels report AIS readings collected to a central location. At a collective location, such as a port, by aggregating AIS readings from multiple vessels, a consensus can be formed of normal vessel behavior. Using a consensus data set, a machine learning model is trained using a vessel behaviors. Once a model is trained at the collection point, models are distributed to all vessels that request a model for use. Once the normal ranges are modeled, anomalies or modifications can be detected to determine if a vessel sensor is creating false data.

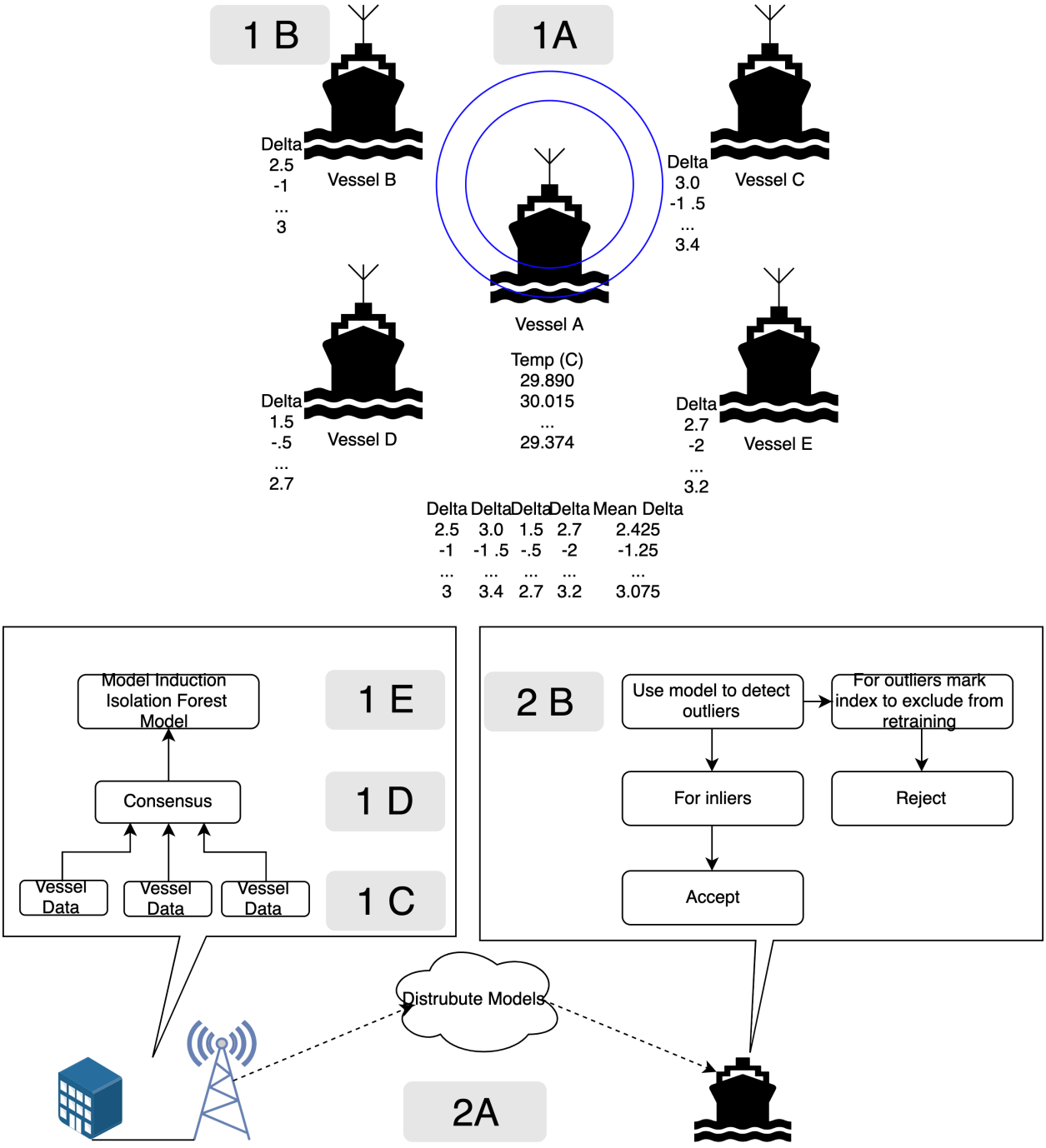


Figure 4.1 Behavior Model Anomaly Detection Process Diagram

4.2 Behavioral Model Anomaly Detection Process

1. Model Induction

    (a) Vessel Broadcast using AIS.

    (b) Vessels within the vicinity receive the AIS and record readings.

    (c) Once in port, the receiving vessels report the recorded readings.

    (d) Using the recorded readings, a consensus is formed of what each vessel is reporting.

    (e) Using the consensus data set, a model is fit to vessel behavior.

2. Progressive Analysis

    (a) The trained models of each vessel are distributed to vessels that request a model.

    (b) While at sea, vessels process all received messages for a vessel using the model fit for anomaly detection.

        - If an observation is classified as inlier, accept.
        - If an observation is classified as outlier, reject.

3. Model Re-Induction

    (a) While at sea, vessels are continuously recording messages and reporting new messages when in port. As more readings are recorded for a vessel, the models are refit as more information becomes available for each vessel.

To facilitate learning vessels, collect AIS data while at sea and record the received values. Once alongshore or at the port, the values are transferred to a shore side server for processing. By sharing sensor data, a model of the operation of a vessel can be created to determine each vessel's sensors' normal operating range. At shore side, collections of all reports on a vessel are weighted together to provide a single truth of how a vessel historically operates. The more reports,

the stronger the data. Using the weighted vessel reports at shore side, a machine learning model is trained for each vessel. While a vessel is at port, the models are transferred to the vessel for offline AIS operation.

With a fit model, new AIS temperature sensor observations are fed into the model for classification. If the classifier identifies reported data as anomalous, said data is recorded and stored to see how often the sensor generates an anomaly.

The results of each test case are compiled into a final summary. The total number of inlier and outliers along with false readings are listed. The average of all tests are listed with the results of analyses.

CHAPTER 5

Numerical Analysis

This chapter contains an analysis of test cases to demonstrate the application of machine learning for anomaly detection using synthetic temperature samples. Designed use cases test the classification of each type of machine learning model after it is fit to a training set with normal observations. The designed use cases check model classification to determine the machine learning model and consensus method to attain the highest true accuracy for classification. We examine Four machine learning methods for use in classification. The four models are selected based on usage for anomaly detection. The four selected are Isolation Forest, Local Outlier Factor, Support Vector Machine, and Elliptic Envelope. Section 5.1 details the data set designed to determine the efficacy of our behavioral modeling methodology using simulated temperature sensor readings. Section 5.2 includes code snippets explaining our machine learning model configuration. Section 5.3 gives an example of behavioral modeling with a reduced number of vessels follow by section 5.4 showing the method of use case result evaluation. The results of each use case in section 5.5 detail each case with an analysis of the results.

5.1 Synthetic Example

A python random weather generator was used to create weather samples. The weather generator randomly generates sample weather data for a given position by latitude and longitude for a date and time. By using historical weather measurements for these locations from Dark Sky API, a set of synthetic samples can be generated for a location, date, and time. Weather samples of

one week are generated, simulating interactions for two vessels over six days that are then used for training a model and one day used for model validation.

A range of dates, along with the number of requested samples, is given for sample generation. Fives sets of samples are created for six days, generating 1000 samples for the two sets, simulating five vessels. The difference between each sample, at each index, is taken to create a single vessel behavior set. The single vessel behavior set simulates the interaction between two vessels with one set for the vessel receiving samples from another vessel within operating range. Using the multiple vessel behavior sets, three methods are used to determine a fit for model behavior. The three methods are mean, median and maximum. Mean is the average of all numbers for a given time period from all sample collected, creating a new synthetic number from a mixture of all the numbers. Median selects the middle reading from all the readings for a given time period. Maximum is taken from the absolute value from the data of either negative or positive for the largest difference recorded.

5.2 Software

Using the difference set for each scenario, different models were trained to detect outliers. The models compared are Isolation Forest, Local Outlier Factor, Support Vector Machine, and the Robust Covariance Ellipitic Envelope.

All simulations are performed in python use scikit-learn models for Isolation Forest, Support Vector Machine, Local Outlier Factor, and Elliptic Envelope. [28]

Model Parameters

1. `IsolationForest(behaviour='new', max_samples=1000, random_state=rng, contamination=0.003, n_jobs=-1, n_estimators=1000)`

    (a) behaviour - Behaviour of the decision function which can be either ’old’ or ’new’. Passing behaviour=’new’ makes the decision function change to match other anomaly

detection algorithm API which will be the default behaviour in the future.

(b) max samples - The actual number of samples

(c) random state - If int, random state is the seed used by the random number generator; If RandomState instance, random state is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

(d) contamination - The amount of contamination of the data set; that is, the proportion of outliers in the data set. Used when fitting to define the threshold on the decision function. If 'auto', the decision function threshold is determined as in the original paper. With our training data set there are no errors present so the value is set to a low value. It still needs some value or it is trained that no outliers ever exists.

(e) n jobs - The number of jobs to run in parallel for both fit and predict.

(f) n estimators - The number of base estimators in the ensemble.

2. `svm.OneClassSVM(nu=0.1, kernel="rbf", gamma=0.1)`

(a) nu - An upper bound on the fraction of training errors and a lower bound of the fraction of support vectors. Should be in the interval (0, 1]. By default 0.5 will be taken.

(b) kernel - Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to precompute the kernel matrix.

(c) gamma - Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

3. `LocalOutlierFactor(novelty=True)`

(a) novelty - Set novelty to True to use LocalOutlierFactor for novelty detection.

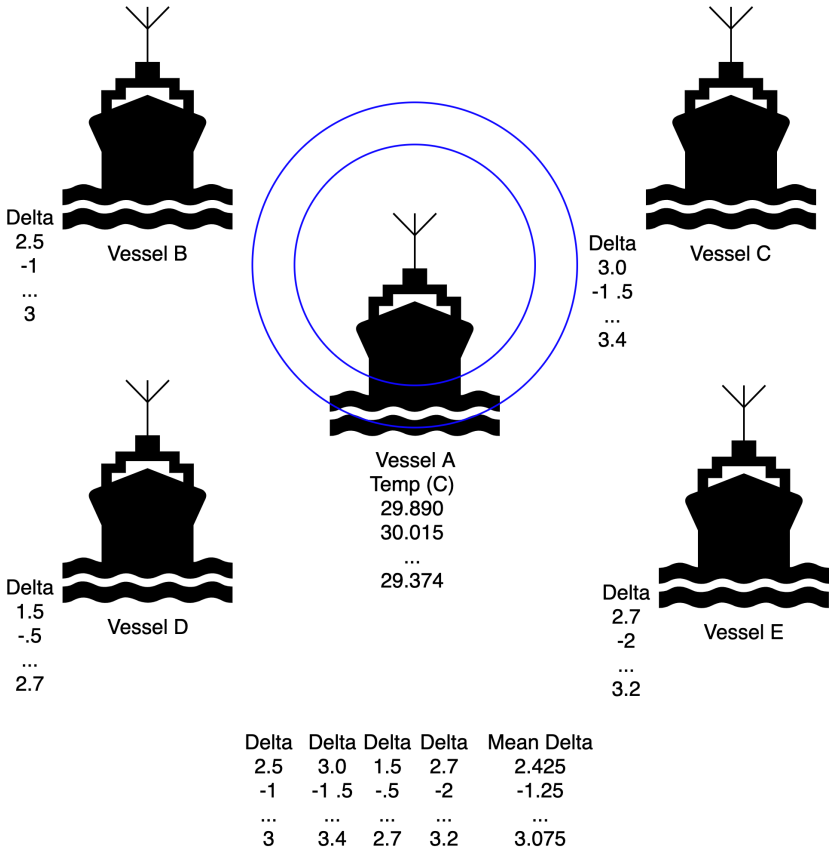
4. `EllipticEnvelope(random_state=0)`

   (a) random state - The seed of the pseudo random number generator to use when shuffling the data.

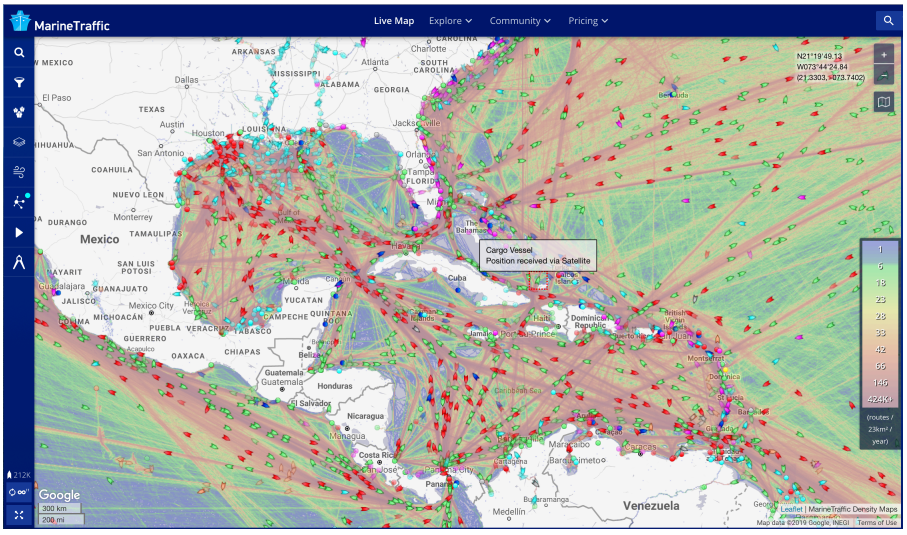
5.3 Model Analysis

To simulate multiple vessels collecting data on a vessel, a python weather simulator generated five synthetic sample sets of 1,000 samples. Each set of samples is the sea surface temperature readings for a vessel. In this case, the vessels will be labeled Vessel A, Vessel B, Vessel C, Vessel D, and Vessel E.

Three cases are given for analysis to select the vessels' weight. First, is the mean of all reports for a given time period. Second, is the median for a given time period. Third, is the max in absolute value for a given time period.



(a) An example vessel communication between vessels A,B,C,D, and E.



(b) Marine Traffic Density Map adopted from [13] of well used vessel routes.

Figure 5.1 Figure demonstrating vessel to vessel communication

Vessels commonly operate within shipping lanes along well-used vessel routes. Marine Traffic, a company that logs vessel movement via AIS, illustrates this fact with a heat map of vessel AIS locations (Figure 5.1b). Areas in red are well-used routes by vessels showing that ships share the same routes in the same region. In some cases, vessels will operate outside of conventional lanes for various reasons. This approach builds off cases in which vessels are within the same area within AIS range, and multiple vessels can communicate with each other. While vessels are within range of each other, AIS observations are recorded by each vessel. As each vessel arrives at a port, they offload the stored AIS observations to port data collection centers for consensus building. Figure 5.1 is a microcosm of the larger case to demonstrate the principles of vessel communication.

Figure 5.1 shows five vessels in a given region. Vessel A is transmitting its AIS readings. Vessels B, C, D, and E are receiving and calculating their difference from Vessel A. The difference is taken into account to factor in change in water temperature over time and regions. What is sought to be known is the difference and the change in reading from one vessel to another. Once in port, the deltas are combined to form a consensus of what difference vessel A is from other vessels. With a consensus, a model is trained and distributed back to vessels for use offline at sea.

Use cases are given below to demonstrate model fit and attempts to falsify information. Attempts to falsify AIS data occurs for many reasons, including spoofing a vessel, generating false readings by error, or to degrade another vessel's rating. Once a valid model is trained, an attacker might attempt to give false readings. The cases below demonstrate the scenario where an attacker attempts to send false readings at various time frames to demonstrate how the machine learning models would classify those readings.

5.3.1 Experimental Use Cases

- Synthetic Reduced Set (Section 5.5.2)
  - This set is for initial testing to demonstrate the model fit on a reduced set with a few inliers and outliers.
- Errors Beginning and End (Section 5.6.1)
  - Anomalies are inserted at the beginning and the end to test if the model detects errors at the start or end of a session.
- Errors in Middle (Section 5.6.2)
  - Errors are inserted in the middle of a session to determine if the model correctly classifies anomalies inserted in the middle of a session.
- Errors at the Edge (Section 5.6.3)
  - This case tests the model to see if it can detect errors outside of the training set to demonstrate what the model fit at the edges.
- Errors Breakout Fraud (Section 5.6.4)
  - Breakout fraud shows the model where one might begin spoofing as a user within the range of the original vessel, but tries to push the readings to a new normal outside of the vessel model.
- Errors Significant (Section 5.6.5)
  - Significant errors demonstrate the model fit for large values outside of the training set to see if the model can accurately classify the errors as outliers. This might be the case in an on-off attack scenario where a vessel shuts off its AIS. If a vessel does not receive a reading from another vessel, then the difference would be significant compared to

previous readings. This would indicate that the vessel is not sending accurate readings or potentially no readings at all.

- Errors Large Uniform (Section 5.6.6)
  - Large uniform errors check that a model correctly classifies errors outside of the set. The model does not determine those readings to be inlier observations, even if those errors appear on a consistent regular basis.
- Errors Random Frequency Selective (Section 5.6.7)
  - This test case combines features from the previous test case to determine model classification if readings are random.

5.4 Analysis Methodology

Model analysis consists of five primary factors: true outlier, false outlier, true inlier, false inlier, true accuracy.

1. True outliers are observations the model detects as being an anomaly and are not contained in the original data set.
2. False outliers are observations that the model identifies as an outlier but are in the original data set.
3. True inliers are observation the model identifies as inliers and that are in the original data set.
4. False inliers are observations the model identifies as inliers and they are not in the original data set.
5. Model accuracy is calculated as the total number of correct identifications over the total number of observations.

5.5 Experimental Analysis

The experimental analysis examines each of the selected machine learning methods against use cases designed from the types of cyberattacks AIS is susceptible to while at sea. The section begins with the induction of each model in section 5.5.1 used in each use case. A reduced example demonstrates how each model trains given a smaller data set similar to the larger data sets used in the use case test. After the presentation of the initial setup of each model, section 5.6 contains results from the designed use cases.

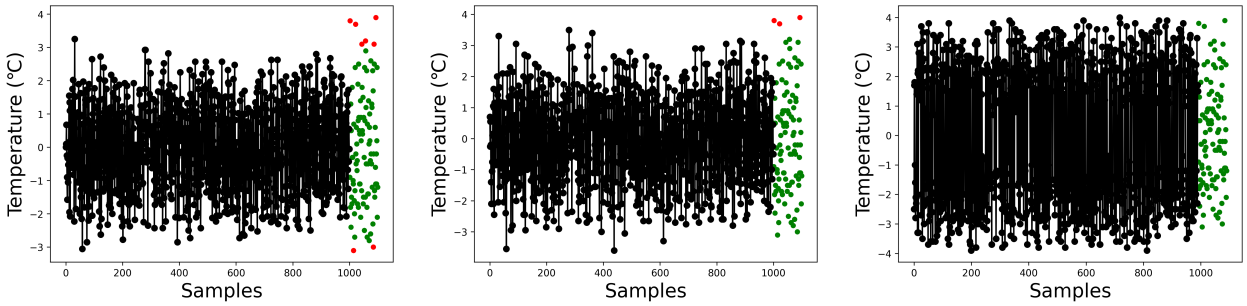
5.5.1 Model Induction

Model Induction is the training and fit of the machine learning models. Using the consensus data sets of mean, median and max, models are fit using each type iForest, SVM, LOF and EE. The models are fit to a sample of 1,000 readings with the last 100 being fed back into the trained model to demonstrate model fit. The models are trained at shore-side, based on vessel consensus. Once a model is fit, the models are distributed to the vessel for offline use while at sea. For every vessel, a model is fit, and used to model one specific vessels' behavior.

Model fit data is plotted using black, dots and lines while test data is illustrated in green and red dots. Green dots indicate an inlier classification, while red dots indicate an outlier classification from the model. A correct model will show only green dots as the test samples are from the original data set without a consensus.

Errors are those readings that are not generated by the original system and are from a different system. In the case of AIS readings, the errors are readings generated by a faulty sensor or another vessel spoofing as another vessel.

5.5.1.1 Model Induction: Isolation Forest

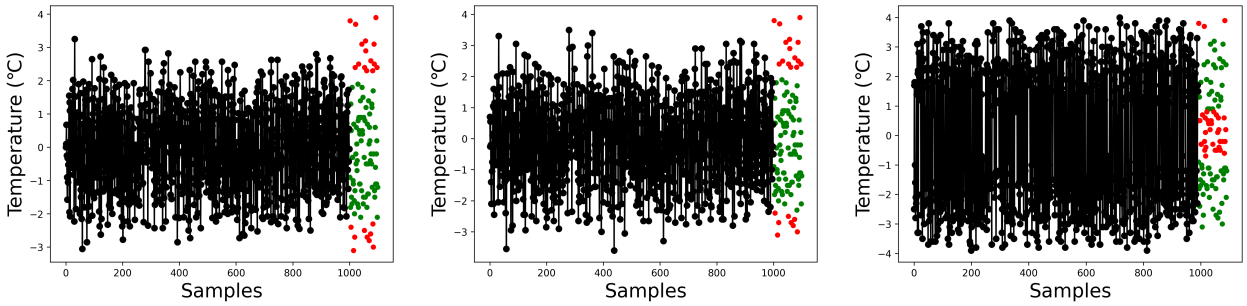


(a) Mean (b) Median (c) Max

Figure 5.2 Isolation Forest Model Induction

Figure 5.2 illustrates model fit using and Isolation Forest using mean, median, and max. In this case, using a python weather simulator, the first 1000 samples are those used for training from the consensus. The last 100 samples are simulated readings without consensus, simulating vessels, communicating offline at sea. Figure 5.2a shows the mean training with some false outliers detected. Figure 5.2b shows the selection of the median sample with fewer false outliers. Figure 5.2c demonstrates the max selected samples of valid values for testing. Max has no false outliers.

5.5.1.2 Model Induction: Support Vector Machine

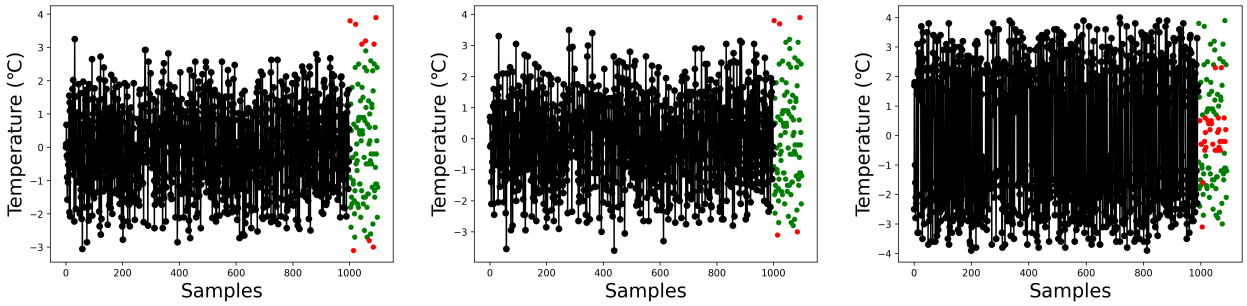


(a) Mean (b) Median (c) Max

Figure 5.3 Support Vector Machine Model Induction

Figure 5.3 illustrates the model fit of a support vector machine using the mean, median, and max consensus. The fit set is shown in black while the test set is in both green and red. Green indicates an inlier is detected. Red indicates an outlier is detected. A perfect fit would show no red outliers.

5.5.1.3 Model Induction: Local Outlier Factor

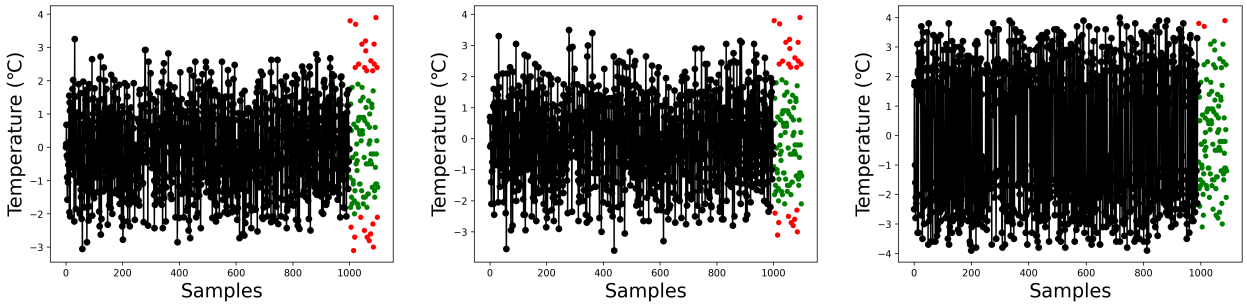


(a) Mean (b) Median (c) Max

Figure 5.4 Local Outlier Factor Model Induction

Figure 5.4 illustrates the model fit by the local outlier factor using the mean, median, and max consensus. The fit set is shown in black, while the test set is in both green and red. Green indicates an inlier is detected. Red indicates an outlier is detected. A perfect fit would show no red outliers.

5.5.1.4 Model Induction: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.5 Robust Covariance Elliptic Envelope Model Induction

Figure 5.5 illustrates the model fit determining the robust covariance elliptic envelope, using the mean, median, and max consensus. The fit set is shown in black while the test set is in both green and red. Green indicates an inlier is detected. Red indicates an outlier is detected. A perfect fit would show no red outliers.

5.5.2 Synthetic Reduced Set

The synthetic reduced set is a subset of the initially generated set for training. Errors are inserted for initial model testing to demonstrate model fit at index 0, 2, 3, and 5.

Table 5.1 Synthetic Reduced Set Summary

(a) Mean

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	5	2	1	3	3	4	1	5	0	0.9
SVM	6	2	1	4	3	4	2	4	0	0.8
LOF	5	2	1	3	3	4	1	5	0	0.9
Elliptic Envelope	6	2	1	4	3	4	2	4	0	0.8

(b) Median

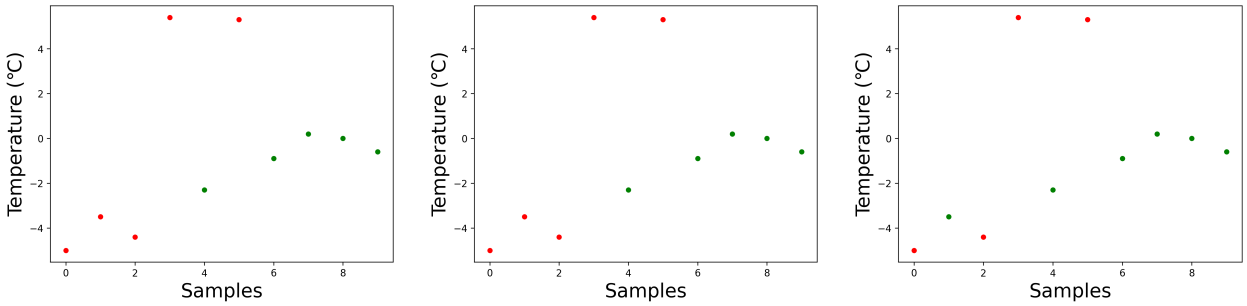
Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	5	2	1	3	3	4	1	5	0	0.9
SVM	5	2	1	3	3	4	1	5	0	0.9
LOF	5	2	1	3	3	4	1	5	0	0.9
Elliptic Envelope	6	2	1	4	3	4	2	4	0	0.8

(c) Max

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	4	2	1	2	1	4	0	6	0	1.0
SVM	7	3	1	3	1	4	3	3	0	0.7
LOF	0	0	0	0	0	0	0	6	4	0.6
Elliptic Envelope	5	2	1	3	3	4	1	5	0	0.9

Table 5.1 compares each model, along with each consensus method of mean, median and max. In this case, Isolation Forest using Max (Figure 5.1c) attains the highest true accuracy rating of 1.0, correctly identifying all true inliers and outliers with no false positives inliers or outliers. The highest attained true accuracy is an isolation forest, using a max consensus method at 100 %.

5.5.2.1 Synthetic Reduced Set: Isolation Forest



(a) Mean (b) Median (c) Max

Figure 5.6 Isolation Forest Small Synthetic Set

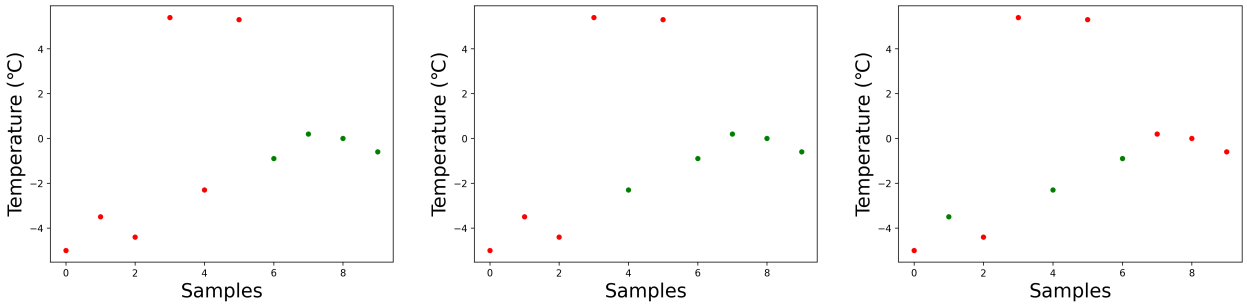
Table 5.2 Isolation Forest Small Synthetic Set

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	5	2	1	3	3	4	1	5	0	0.9
Median	5	2	1	3	3	4	1	5	0	0.9
Max	4	2	1	2	1	4	0	6	0	1.0

Using an isolation forest and max, (Figure 5.6c) the highest true accuracy is attained at 1.0. Max is the only case to correctly identify all the outliers, while not selecting any inliers as false outliers. Table 5.2 numerically compares the isolation forest against the consensus methods of mean, median, and max.

This data indicates that using max as a consensus for the training set correctly fits an isolation forest to identify anomalies outside of the original dataset.

5.5.2.2 Synthetic Reduced Set: Support Vector Machine



(a) Mean (b) Median (c) Max

Figure 5.7 Support Vector Machine Small Synthetic Set

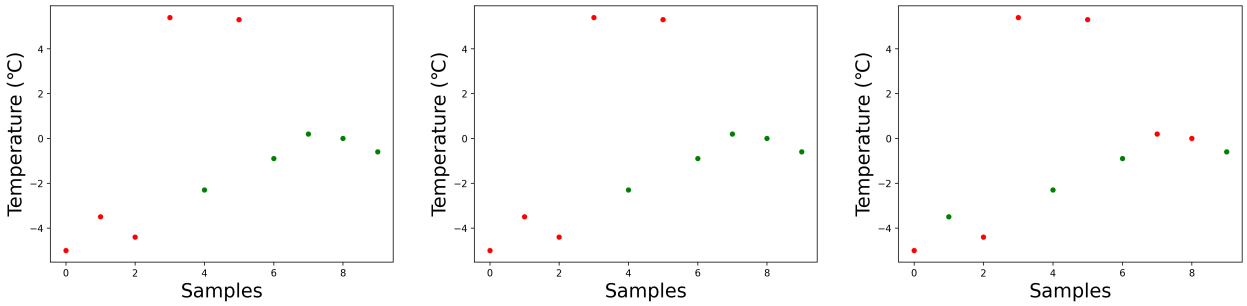
Table 5.3 Support Vector Machine Small Synthetic Set

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	6	2	1	4	3	4	2	4	0	0.8
Median	5	2	1	3	3	4	1	5	0	0.9
Max	7	3	1	3	1	4	3	3	0	0.7

Using a support vector machine and median, (Figure 5.7b), the highest true accuracy is attained at 0.9. Table 5.3 numerically compares the support vector machine against the consensus methods of mean, median, and max.

This test indicates highest confidence that a SVM achieves is a correct classification of 90% of the readings for small synthetic data set using a median consensus.

5.5.2.3 Synthetic Reduced Set: Local Outlier Factor



(a) Mean (b) Median (c) Max

Figure 5.8 Local Outlier Factor Small Synthetic Set

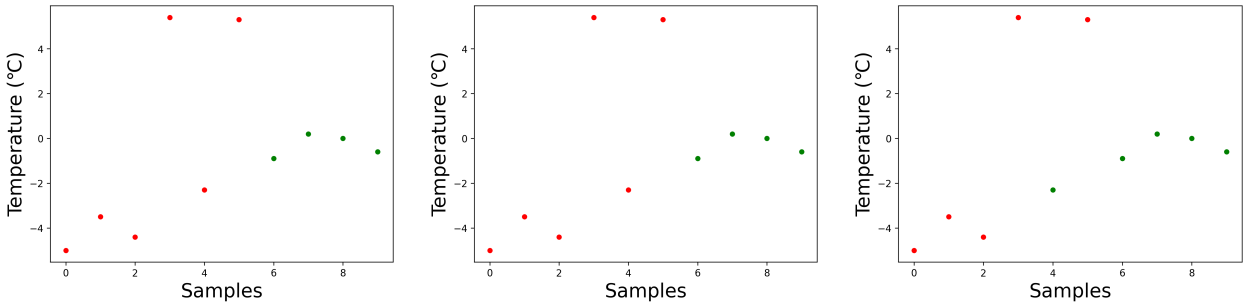
Table 5.4 Local Outlier Factor Small Synthetic Set

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	5	2	1	3	3	4	1	5	0	0.9
Median	5	2	1	3	3	4	1	5	0	0.9
Max	0	0	0	0	0	0	0	6	4	0.6

Figure 5.8 and Table 5.4 shows the training and fit for the small synthetic set, using the local outlier factor with mean, median, and max consensus methods. Both mean, (Figure 5.8a), and median, (Figure 5.8b), attain a 0.9 true accuracy, identifying some of the errors while also identifying one false outlier. The max, (Figure 5.8c), does not identify any errors.

This indicates highest confidence a LOF model, using mean or median consensus, attains a 90% correct classification.

5.5.2.4 Synthetic Reduced Set: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.9 Robust Covariance Elliptic Envelope Small Synthetic Set

Table 5.5 Robust Covariance Elliptic Envelope Small Synthetic Set

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	6	2	1	4	3	4	2	4	0	0.8
Median	6	2	1	4	3	4	2	4	0	0.8
Max	5	2	1	3	3	4	1	5	0	0.9

Figure 5.9 demonstrates a model fit for a robust covariance elliptic envelope using mean, median, and max consensus methods. Table 5.5 presents a numerical comparison of each method with max (Figure 5.9c), attaining the highest at 0.9 with only one false outlier and all true outliers being identified.

This review indicates the highest confidence an elliptic envelope attains is 90% using a max consensus method.

5.6 Experimental Use Cases

The experimental use case section provides tests to determine how each machine learning model performs using each consensus method. Each use case is based upon different types of cyber-attacks that AIS is susceptible to during operation.

5.6.1 Errors: Beginning and End

This case demonstrates a model fit for outlier classifiction at the beginning and end of a dataset.

Table 5.6 Errors: Beginning and End Summary

(a) Mean

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	10	7	1	3	1	2	8	90	0	0.92
SVM	26	16	2	10	1	2	24	74	0	0.76
LOF	21	8	2	13	1	2	19	79	0	0.81
Elliptic Envelope	28	16	2	12	2	2	26	72	0	0.74

(b) Median

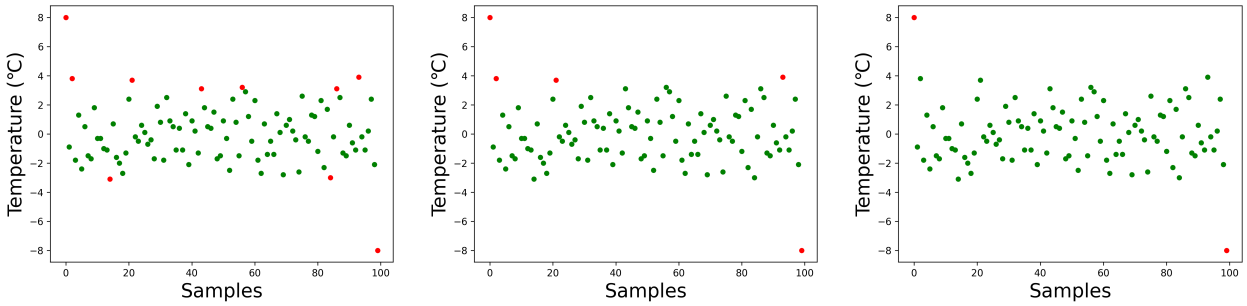
Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	5	4	1	1	1	2	3	95	0	0.97
SVM	25	16	2	9	1	2	23	75	0	0.77
LOF	12	7	1	5	1	2	10	88	0	0.9
Elliptic Envelope	26	16	2	10	1	2	24	74	0	0.76

(c) Max

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	2	1	1	1	1	2	0	98	0	1.0
SVM	37	21	2	16	2	2	35	63	0	0.65
LOF	1	0	0	1	1	0	1	97	2	0.97
Elliptic Envelope	5	4	1	1	1	2	3	95	0	0.97

Comparing mean, median, and max, (Table 5.6) the highest score attained is an isolation forest using a max consensus with 100% correct classification of temperature difference readings.

5.6.1.1 Errors Beginning and End: Isolation Forest



(a) Mean (b) Median (c) Max

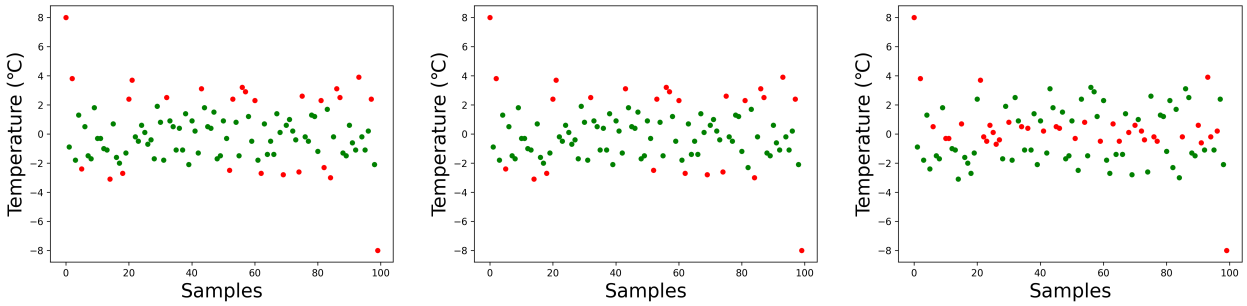
Figure 5.10 Isolation Forest Errors at Beginning and End

Table 5.7 Isolation Forest Errors at Beginning and End

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	10	7	1	3	1	2	8	90	0	0.92
Median	5	4	1	1	1	2	3	95	0	0.97
Max	2	1	1	1	1	2	0	98	0	1.0

Figure 5.10 illustrates model fit and outlier detection of the isolation forest. Table 5.7 indicates that an isolation forest, using a max consensus, attains the highest correct classification at 100%.

5.6.1.2 Errors Beginning and End: Support Vector Machine



(a) Mean (b) Median (c) Max

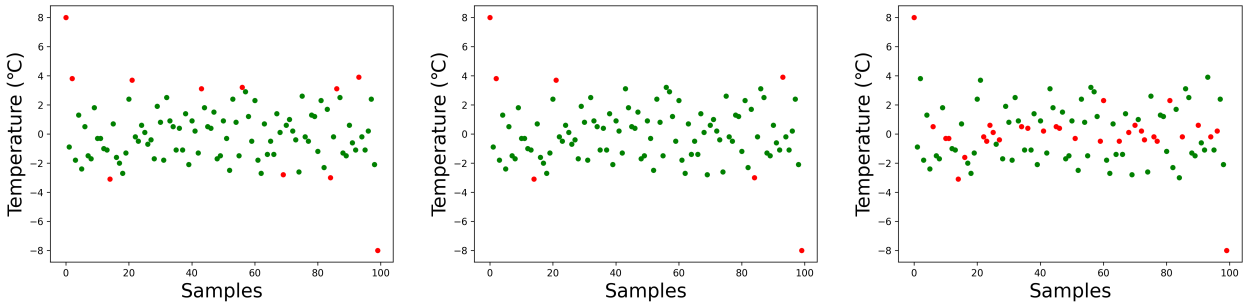
Figure 5.11 Support Vector Machine at Beginning and End

Table 5.8 Support Vector Machine at Beginning and End

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	26	16	2	10	1	2	24	74	0	0.76
Median	25	16	2	9	1	2	23	75	0	0.77
Max	37	21	2	16	2	2	35	63	0	0.65

Figure 5.11 illustrates SVM model classification for errors at the beginning and the end of a session. Table 5.8 shows the highest accuracy attained using the median consensus method at 77%.

5.6.1.3 Errors Beginning and End: Local Outlier Factor



(a) Mean (b) Median (c) Max

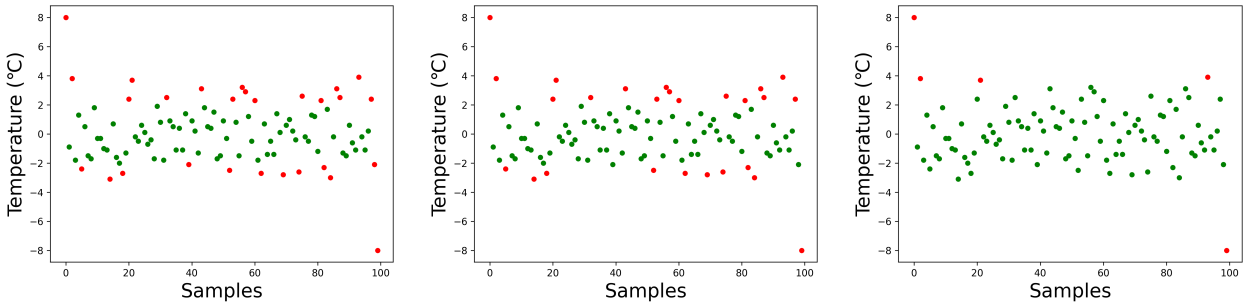
Figure 5.12 Local Outlier Factor at Beginning and End

Table 5.9 Local Outlier Factor at Beginning and End

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	21	8	2	13	1	2	19	79	0	0.81
Median	12	7	1	5	1	2	10	88	0	0.9
Max	1	0	0	1	1	0	1	97	2	0.97

Figure 5.12 plots the classification of the local outlier factor using mean, median, and max. Table 5.9 shows the local outlier factor attains a 97% true accuracy rating. The model does correctly identify most of the inliers, but does not identify any of the outliers. Using LOF with max would not detect any of the outliers properly. The other consensus methods could detect the outliers, but they also classify many inliers as outliers.

5.6.1.4 Errors Beginning and End: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.13 Robust Covariance Elliptic Envelope at Beginning and End

Table 5.10 Robust Covariance Elliptic Envelope at Beginning and End

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	28	16	2	12	2	2	26	72	0	0.74
Median	26	16	2	10	1	2	24	74	0	0.76
Max	5	4	1	1	1	2	3	95	0	0.97

Figure 5.10 illustrates a robust covariance elliptic envelope fit to mean, median, and max consensus sets. Table 5.10 shows the max consensus method attains the highest true accuracy of 97%, while identifying the errors correctly.

5.6.2 Errors: Middle

Anomalies are inserted at the beginning and the end of the data to determine if the model detects errors at the start or end of a session.

Table 5.11 Errors: Middle Summary

(a) Mean

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	10	7	1	3	1	2	8	90	0	0.92
SVM	26	16	2	10	1	2	24	74	0	0.76
LOF	21	8	2	13	1	2	19	79	0	0.81
Elliptic Envelope	28	16	2	12	1	2	26	72	0	0.74

(b) Median

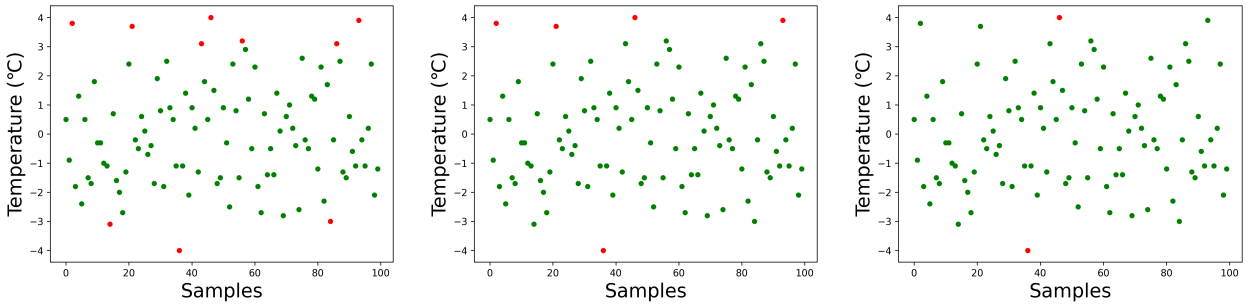
Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	5	4	1	1	1	2	3	95	0	0.97
SVM	25	16	2	9	1	2	23	75	0	0.77
LOF	12	7	1	5	1	2	10	88	0	0.9
Elliptic Envelope	26	16	2	10	1	2	24	74	0	0.76

(c) Max

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	2	1	1	1	1	2	0	98	0	1.0
SVM	36	20	2	16	2	2	34	64	0	0.66
LOF	1	0	0	1	1	0	1	97	2	0.97
Elliptic Envelope	5	4	1	1	1	2	3	95	0	0.97

Table 5.11 presents a summary of mean, median, and max consensus methods for each machine learning model. For errors in the middle, the highest attained true accuracy is using an isolation forest using max consensus for a 100% true accuracy. This indicates that all observations were classified correctly.

5.6.2.1 Errors Middle: Isolation Forest



(a) Mean (b) Median (c) Max

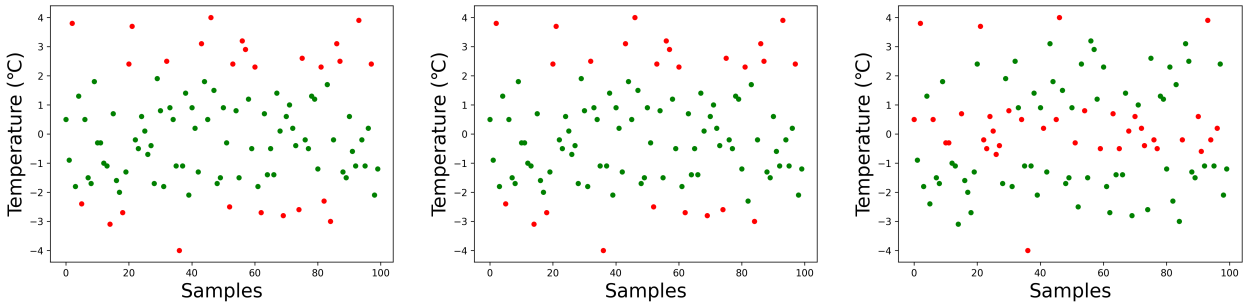
Figure 5.14 Isolation Forest Errors in The Middle

Table 5.12 Isolation Forest Errors in The Middle

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	10	7	1	3	1	2	8	90	0	0.92
Median	5	4	1	1	1	2	3	95	0	0.97
Max	2	1	1	1	1	2	0	98	0	1.0

Figure 5.14 plots the classification of an isolation forest on the data set, inserting anomalies in the middle of the data set. Table 5.12 shows the numerical results of the classifier. The highest attained true accuracy is 100%, using the max consensus method.

5.6.2.2 Errors Middle: Support Vector Machine



(a) Mean (b) Median (c) Max

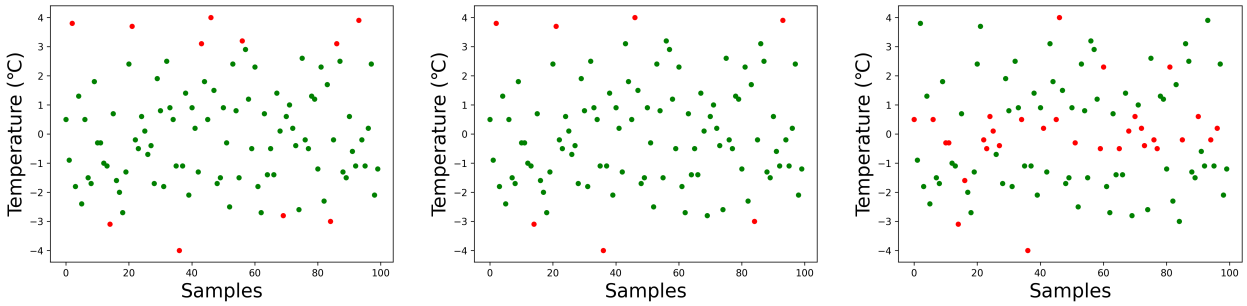
Figure 5.15 Support Vector Machine Errors in The Middle

Table 5.13 Support Vector Machine Errors in The Middle

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	26	16	2	10	1	2	24	74	0	0.76
Median	25	16	2	9	1	2	23	75	0	0.77
Max	36	20	2	16	2	2	34	64	0	0.66

Figure 5.15 shows the results of classification using a support vector machine for the data set containing errors in the middle. Table 5.13 contains the numerical results and that the highest attained support vector machine occurs using the median consensus method in that 77% of the observations were classified correctly.

5.6.2.3 Errors Middle: Local Outlier Factor



(a) Mean (b) Median (c) Max

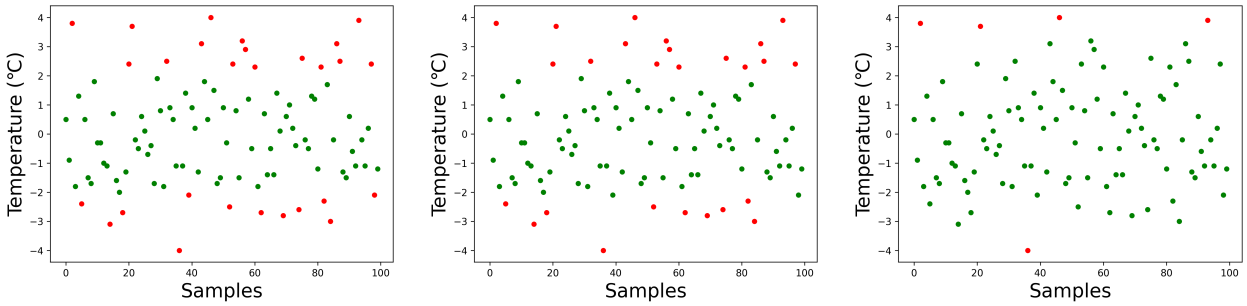
Figure 5.16 Local Outlier Factor Errors in The Middle

Table 5.14 Local Outlier Factor Errors in The Middle

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	21	8	2	13	1	2	19	79	0	0.81
Median	12	7	1	5	1	2	10	88	0	0.9
Max	1	0	0	1	1	0	1	97	2	0.97

Figure 5.14 illustrates classification using the local outlier factor for the data set containing errors in the middle. Table 5.14 shows the numerical results in that the highest attained true accuracy is using the max consensus at 97%. With this, the local outlier factor only classifies the inliers, but does not classify the outlier correctly.

5.6.2.4 Errors Middle: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.17 Robust Covariance Elliptic Envelope Errors in The Middle

Table 5.15 Robust Covariance Elliptic Envelope Errors in The Middle

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	28	16	2	12	1	2	26	72	0	0.74
Median	26	16	2	10	1	2	24	74	0	0.76
Max	5	4	1	1	1	2	3	95	0	0.97

Figure 5.17 illustrates classification using a robust covariance elliptic envelope for the data set containing errors in the middle. Table 5.15 shows the numerical results in that the highest attained true accuracy is using the max consensus at 97%.

5.6.3 Errors: Edge

This case tests the model to see if it can detect errors at the edge of the training set within a one to two temperature degree difference.

Table 5.16 Errors: Edge Summary

(a) Mean

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	59	29	26	30	29	54	5	41	0	0.95
SVM	69	35	26	34	29	54	15	31	0	0.85
LOF	66	30	26	36	29	54	12	34	0	0.88
Elliptic Envelope	70	35	26	35	29	54	16	30	0	0.84

(b) Median

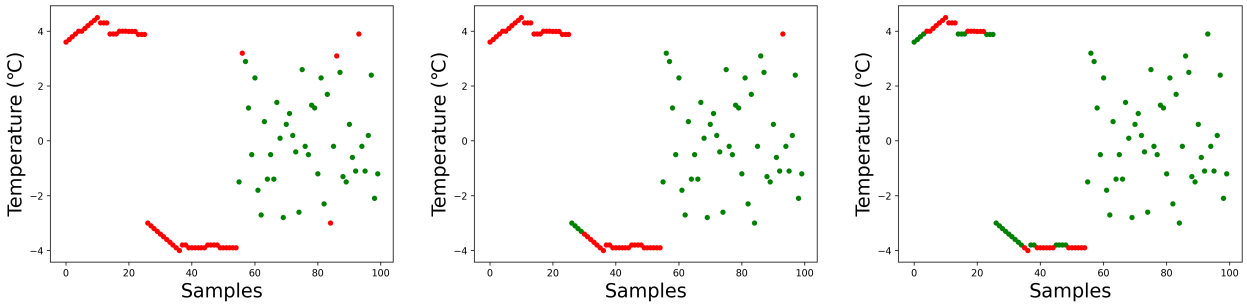
Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	52	27	26	25	25	50	2	44	4	0.94
SVM	68	35	26	33	29	54	14	32	0	0.86
LOF	60	29	26	31	29	54	6	40	0	0.94
Elliptic Envelope	69	35	26	34	29	54	15	31	0	0.85

(c) Max

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	30	16	10	14	6	30	0	46	24	0.76
SVM	64	33	26	31	23	48	16	30	6	0.78
LOF	1	0	0	1	1	1	0	46	53	0.47
Elliptic Envelope	52	27	26	25	25	50	2	44	4	0.94

Table 5.16 presents the results of each model fit using a consensus of mean, median, and max. For the test case of errors at the edge of normal observations, the highest score is attained by the isolation forest using the mean consensus method for a 95% true accuracy rating.

5.6.3.1 Errors Edge: Isolation Forest



(a) Mean (b) Median (c) Max

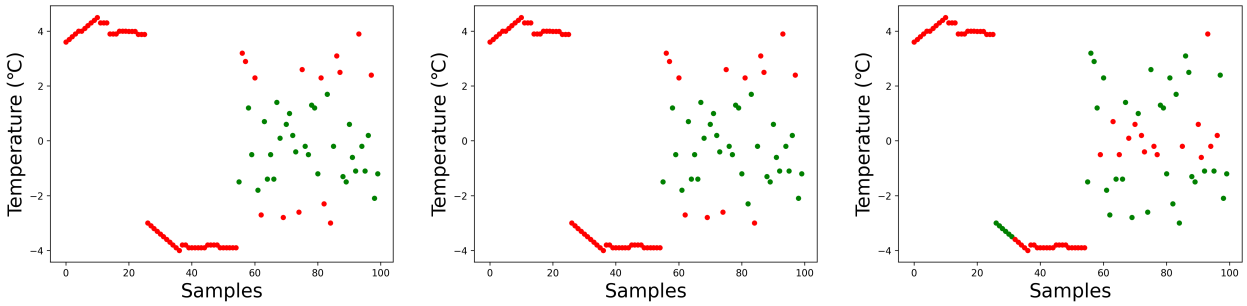
Figure 5.18 Isolation Forest Errors at Edge

Table 5.17 Isolation Forest Errors at Edge

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	59	29	26	30	29	54	5	41	0	0.95
Median	52	27	26	25	25	50	2	44	4	0.94
Max	30	16	10	14	6	30	0	46	24	0.76

Figure 5.18 illustrates classification using the isolation forest for the data set containing anomalies at the edge. Table 5.17 shows the numerical results in that the highest attained true accuracy is using the mean consensus at 95%. In this case, the isolation forest does not detect anomalies for values that are within the range of the observation set the model is fit to. The mean fit set, (Figure 5.18a), appears to only classify the anomalies correctly for the training set mean training set (Figure 5.2a), when the fit is much more narrow than the median or max consensus set.

5.6.3.2 Errors Edge: Support Vector Machine



(a) Mean (b) Median (c) Max

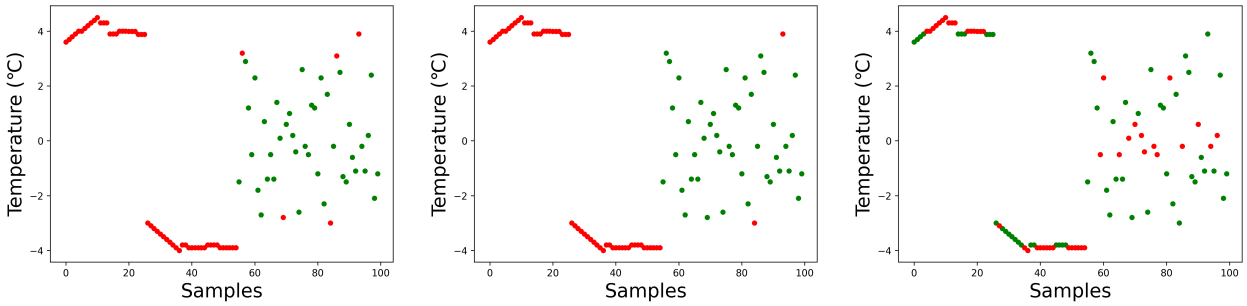
Figure 5.19 Support Vector Machine Errors at Edge

Table 5.18 Support Vector Machine Errors at Edge

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	69	35	26	34	29	54	15	31	0	0.85
Median	68	35	26	33	29	54	14	32	0	0.86
Max	64	33	26	31	23	48	16	30	6	0.78

Figure 5.19 illustrates classification using a support vector machine for the data set containing errors at the edge of the data set within one to two degrees difference. Table 5.18 shows the numerical results in that the highest attained true accuracy is using the mean consensus at 85%.

5.6.3.3 Errors Edge: Local Outlier Factor



(a) Mean (b) Median (c) Max

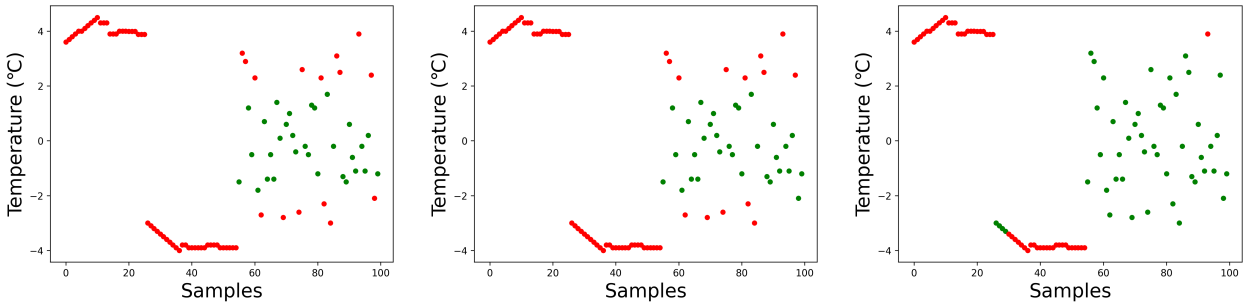
Figure 5.20 Local Outlier Factor Errors at Edge

Table 5.19 Local Outlier Factor Errors at Edge

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	66	30	26	36	29	54	12	34	0	0.88
Median	60	29	26	31	29	54	6	40	0	0.94
Max	1	0	0	1	1	1	0	46	53	0.47

Figure 5.20 illustrates classification using the local outlier factor for the data set containing errors at the edge of the data set. Table 5.19 shows the numerical results in that the highest attained true accuracy is using the median consensus at 94%.

5.6.3.4 Errors Edge: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.21 Robust Covariance Elliptic Envelope Errors at Edge

Table 5.20 Robust Covariance Elliptic Envelope Errors at Edge

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	70	35	26	35	29	54	16	30	0	0.84
Median	69	35	26	34	29	54	15	31	0	0.85
Max	52	27	26	25	25	50	2	44	4	0.94

Figure 5.21 illustrates classification using the local outlier factor for the data set containing errors in at the edge. Table 5.20 shows the numerical results in that the highest attained true accuracy is using the median consensus at 94%.

5.6.4 Errors: Breakout Fraud

Breakout fraud test the model where one might begin spoofing as a user within the range of the original vessel reading, but the attacker tries to push the readings to a new normal outside of the vessel model.

Table 5.21 Errors: Breakout Fraud Summary

(a) Mean

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	17	16	12	1	1	12	5	74	9	0.86
SVM	34	27	17	7	1	17	17	62	4	0.79
LOF	30	19	14	11	1	14	16	63	7	0.77
Elliptic Envelope	35	27	17	8	1	17	18	61	4	0.78

(b) Median

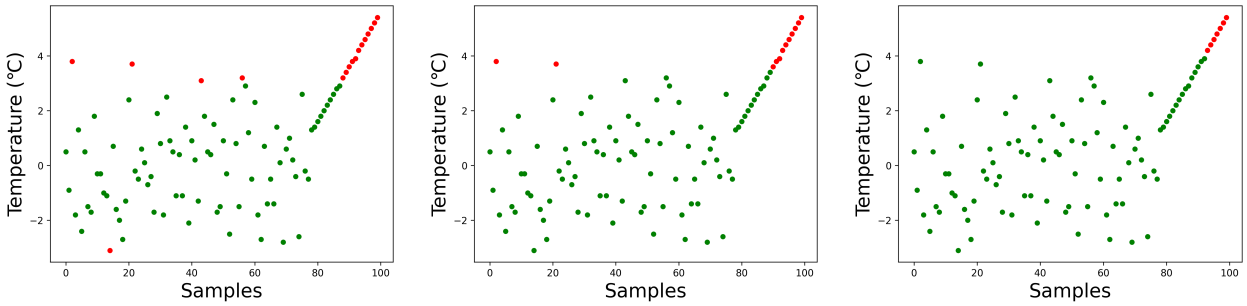
Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	12	12	10	0	0	10	2	77	11	0.87
SVM	33	26	16	7	1	16	17	62	5	0.78
LOF	20	17	12	3	1	13	7	72	8	0.85
Elliptic Envelope	34	27	17	7	1	17	17	62	4	0.79

(c) Max

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	7	7	7	0	0	7	0	79	14	0.86
SVM	40	28	10	12	2	10	30	49	11	0.59
LOF	2	1	1	1	1	1	1	78	20	0.79
Elliptic Envelope	13	13	11	0	0	11	2	77	10	0.88

For breakout fraud table 5.21 shows the highest attained true accuracy is through an elliptic envelope using a max consensus model at 88%.

5.6.4.1 Errors Breakout Fraud: Isolation Forest



(a) Mean (b) Median (c) Max

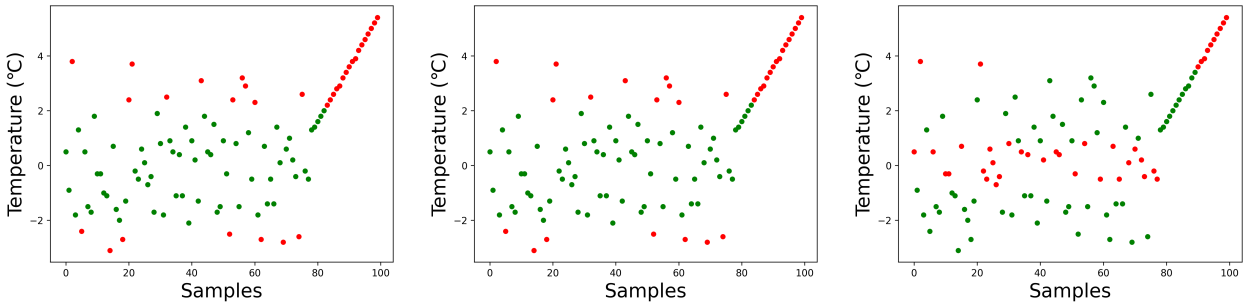
Figure 5.22 Isolation Forest Breakout Fraud

Table 5.22 Isolation Forest Breakout Fraud

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	17	16	12	1	1	12	5	74	9	0.86
Median	12	12	10	0	0	10	2	77	11	0.87
Max	7	7	7	0	0	7	0	79	14	0.86

Figure 5.22 illustrates breakout fraud model testing for the mean, median, and max consensus methods using an isolation forest. Table 5.22 presents the numerical results of model analysis. The highest attained is by using a median consensus at 87%.

5.6.4.2 Errors Breakout Fraud: Support Vector Machine



(a) Mean (b) Median (c) Max

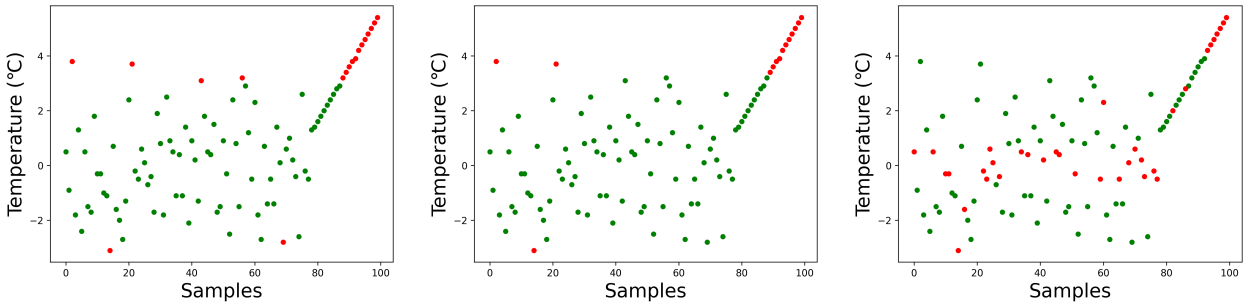
Figure 5.23 Support Vector Machine Breakout Fraud

Table 5.23 Support Vector Machine Breakout Fraud

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	34	27	17	7	1	17	17	62	4	0.79
Median	33	26	16	7	1	16	17	62	5	0.78
Max	40	28	10	12	2	10	30	49	11	0.59

Figure 5.23 illustrates breakout fraud model testing for the mean, median, and max consensus methods using a support vector machine. Table 5.23 presents the numerical results of model analysis. The highest accuracy attained is obtained by using a mean consensus at 79%.

5.6.4.3 Errors Breakout Fraud: Local Outlier Factor



(a) Mean (b) Median (c) Max

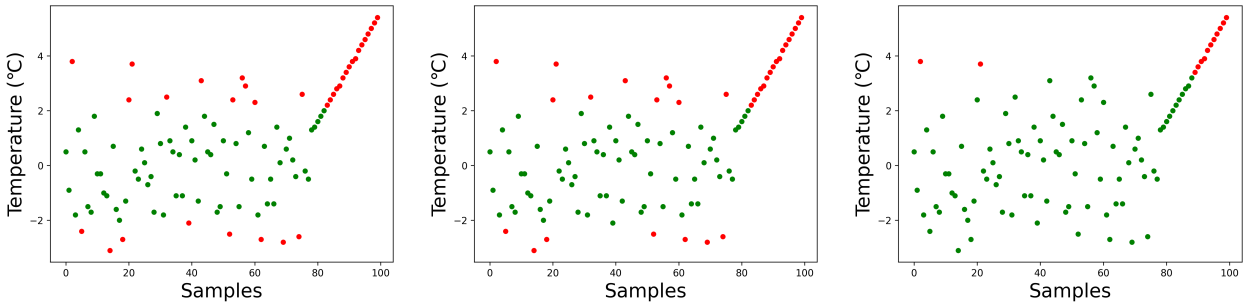
Figure 5.24 Local Outlier Factor Breakout Fraud

Table 5.24 Local Outlier Factor Breakout Fraud

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	30	19	14	11	1	14	16	63	7	0.77
Median	20	17	12	3	1	13	7	72	8	0.85
Max	2	1	1	1	1	1	1	78	20	0.79

Figure 5.24 illustrates breakout fraud model testing for the mean, median, and max consensus methods using the local outlier factor. Table 5.24 presents the numerical results of model analysis. The highest attained accuracy is obtained by using a median consensus at 85%.

5.6.4.4 Errors Breakout Fraud: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.25 Robust Covariance Elliptic Envelope Breakout Fraud

Table 5.25 Robust Covariance Elliptic Envelope Breakout Fraud

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	35	27	17	8	1	17	18	61	4	0.78
Median	34	27	17	7	1	17	17	62	4	0.79
Max	13	13	11	0	0	11	2	77	10	0.88

Figure 5.25 illustrates breakout fraud model testing for the mean, median, and max consensus methods using the elliptic envelope. Table 5.25 presents the numerical results of model analysis. The highest attained is by using a max consensus at 88%.

5.6.5 Significant Errors

Significant errors demonstrate the model fit for large values outside of the training set to see if the model can accurately classify them as outliers. This might be the case in an on-off attack where a vessel shuts off its AIS. If a vessel does not receive a reading from another vessel, then the data difference would be significant compared to previous readings. This would indicate that the

vessel is not sending accurate readings or potentially no readings at all.

Table 5.26 Errors: Significant Errors Summary

(a) Mean

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	16	11	6	5	4	11	5	82	2	0.93
SVM	30	19	7	11	4	11	19	68	2	0.79
LOF	27	12	6	15	4	11	16	71	2	0.82
Elliptic Envelope	32	19	7	13	4	11	21	66	2	0.77

(b) Median

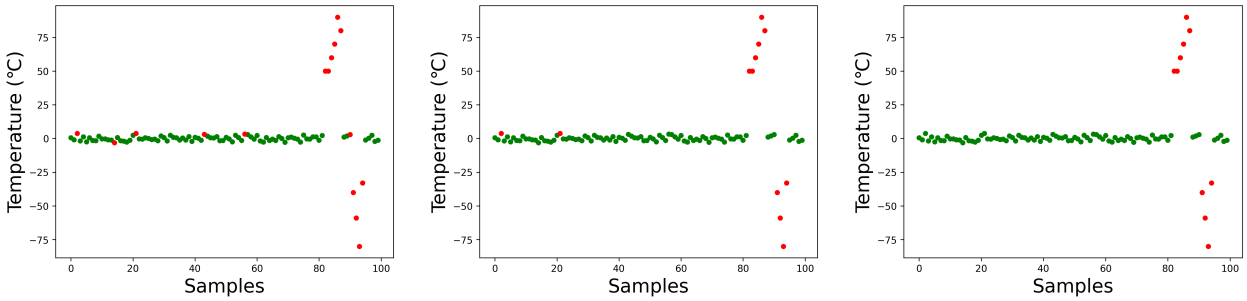
Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	12	8	6	4	4	10	2	85	3	0.95
SVM	30	19	7	11	4	11	19	68	2	0.79
LOF	18	11	6	7	4	11	7	80	2	0.91
Elliptic Envelope	30	19	7	11	4	11	19	68	2	0.79

(c) Max

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	10	6	6	4	4	10	0	87	3	0.97
SVM	41	25	6	16	4	10	31	56	3	0.66
LOF	3	2	2	1	1	2	1	86	11	0.88
Elliptic Envelope	12	8	6	4	4	10	2	85	3	0.95

Table 5.26 presents the numerical results of each machine learning model fit to a mean, median, and max consensus set for a data set with significant errors, both positive and negative. In this case, the highest performing model, with the greatest true accuracy, is an isolation forest using the max consensus set at 97%.

5.6.5.1 Significant Errors: Isolation Forest



(a) Mean (b) Median (c) Max

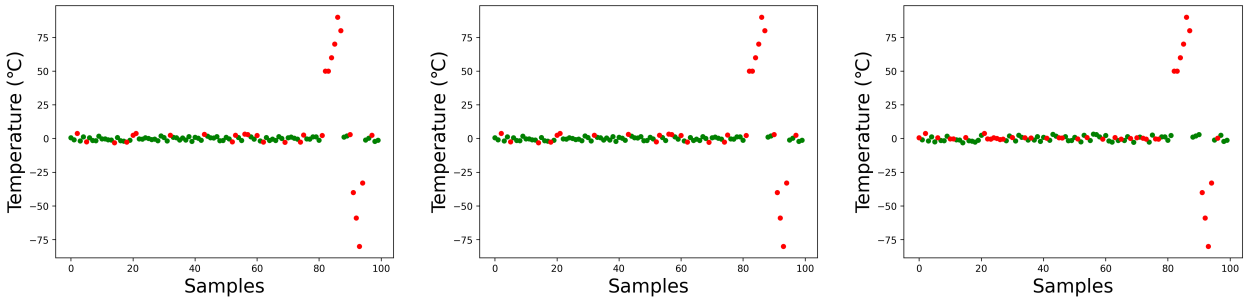
Figure 5.26 Isolation Forest Significant Errors

Table 5.27 Isolation Forest Significant Errors

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	16	11	6	5	4	11	5	82	2	0.93
Median	12	8	6	4	4	10	2	85	3	0.95
Max	10	6	6	4	4	10	0	87	3	0.97

Figure 5.26 shows model classification for significant positive and negative anomalies using an isolation forest. Table 5.27 presents the numerical results of each test. The highest attained true accuracy is through the max consensus fit at 97%.

5.6.5.2 Significant Errors: Support Vector Machine



(a) Mean (b) Median (c) Max

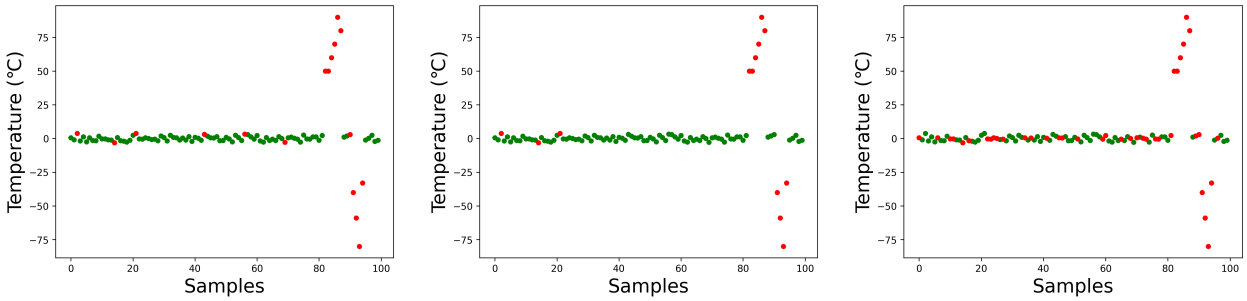
Figure 5.27 Support Vector Machine Significant Errors

Table 5.28 Support Vector Machine Significant Errors

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	30	19	7	11	4	11	19	68	2	0.79
Median	30	19	7	11	4	11	19	68	2	0.79
Max	41	25	6	16	4	10	31	56	3	0.66

Figure 5.27 shows model classification for significant positive and negative anomalies using a support vector machine. Table 5.28 presents the numerical results of each test. The highest attained true accuracy is through the mean and median consensus fit at 79%.

5.6.5.3 Significant Errors: Local Outlier Factor



(a) Mean (b) Median (c) Max

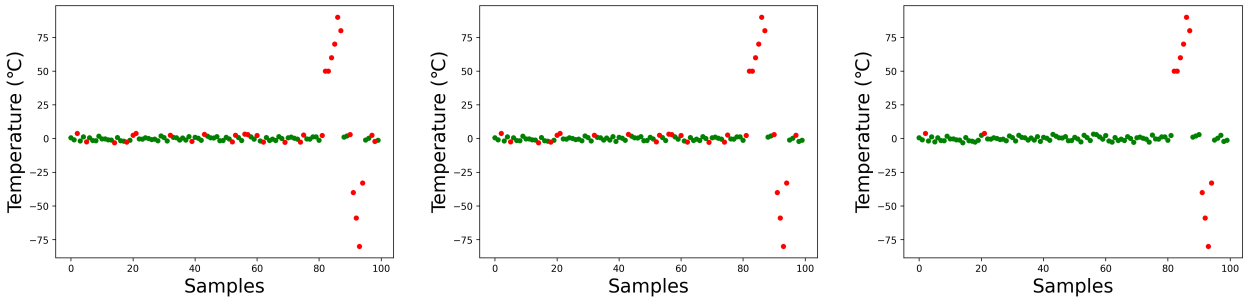
Figure 5.28 Local Outlier Factor Significant Errors

Table 5.29 Local Outlier Factor Significant Errors

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	27	12	6	15	4	11	16	71	2	0.82
Median	18	11	6	7	4	11	7	80	2	0.91
Max	3	2	2	1	1	2	1	86	11	0.88

Figure 5.28 shows model classification for significant positive and negative anomalies using the local outlier factor. Table 5.29 presents the numerical results of each test. The highest attained true accuracy is through the median consensus fit at 91%.

5.6.5.4 Significant Errors: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.29 Robust Covariance Elliptic Envelope Significant Errors

Table 5.30 Robust Covariance Elliptic Envelope Significant Errors

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	32	19	7	13	4	11	21	66	2	0.77
Median	30	19	7	11	4	11	19	68	2	0.79
Max	12	8	6	4	4	10	2	85	3	0.95

Figure 5.26 shows model classification for significant positive and negative anomalies using a robust covariance elliptic envelope. Table 5.27 presents the numerical results of each test. The highest attained true accuracy is through the max consensus fit at 95%.

5.6.6 Errors: Large Uniform

Large uniform errors checks that a model correctly classifies errors outside of the set and that the model does not determine those readings to be inlier observations even if those errors appear on a consistent regular basis.

Table 5.31 Errors: Large Uniform Summary

(a) Mean

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	18	13	1	5	1	10	8	82	0	0.92
SVM	33	22	3	11	1	10	23	67	0	0.77
LOF	28	14	2	14	2	10	18	72	0	0.82
Elliptic Envelope	34	22	3	12	1	10	24	66	0	0.76

(b) Median

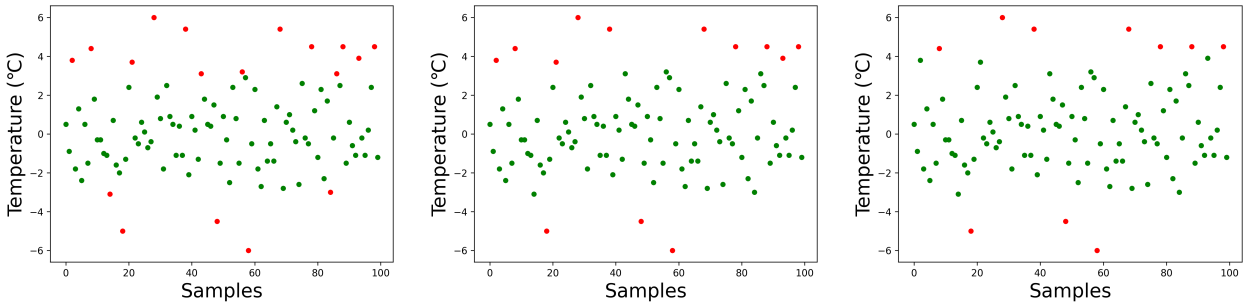
Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	13	10	1	3	1	10	3	87	0	0.97
SVM	32	22	3	10	1	10	22	68	0	0.78
LOF	20	13	1	7	1	10	10	80	0	0.9
Elliptic Envelope	33	22	3	11	1	10	23	67	0	0.77

(c) Max

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	10	7	1	3	1	10	0	90	0	1.0
SVM	45	27	2	18	2	10	35	55	0	0.65
LOF	1	0	0	1	1	0	1	89	10	0.89
Elliptic Envelope	13	10	1	3	1	10	3	87	0	0.97

For large uniform anomalies, Table 5.31 shows the mean, median, and max consensus fit test results. The highest attained true accuracy is from an isolation forest using a max consensus at 100%.

5.6.6.1 Errors Large Uniform: Isolation Forest



(a) Mean (b) Median (c) Max

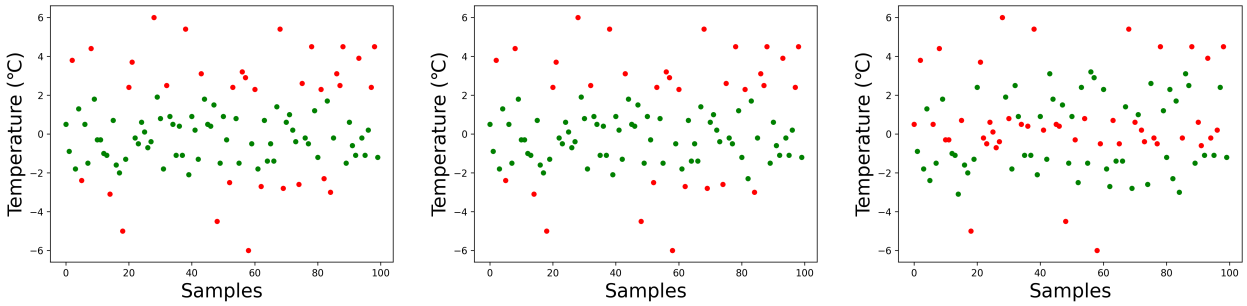
Figure 5.30 Isolation Forest Large Uniform

Table 5.32 Isolation Forest Large Uniform

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	18	13	1	5	1	10	8	82	0	0.92
Median	13	10	1	3	1	10	3	87	0	0.97
Max	10	7	1	3	1	10	0	90	0	1.0

Figure 5.30 plots the large uniform case using an isolation forest for mean, median, and max. Table 5.32 contains the numerical analysis of each case with the highest true accuracy attained using the median consensus at 97%.

5.6.6.2 Errors Large Uniform: Support Vector Machine



(a) Mean (b) Median (c) Max

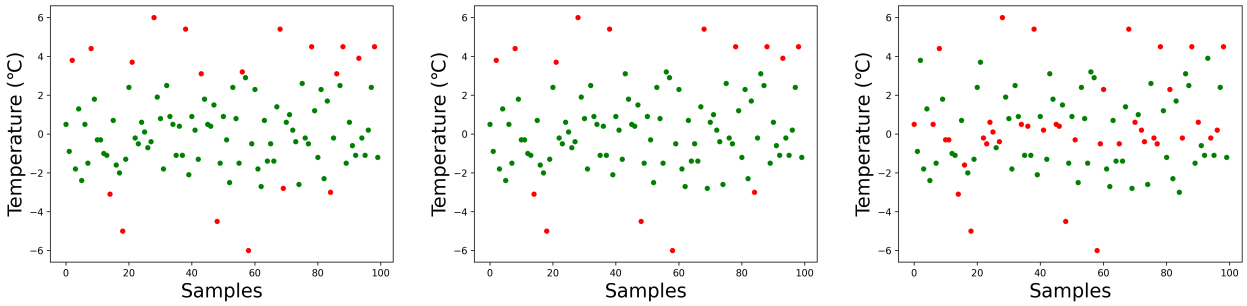
Figure 5.31 Support Vector Machine Large Uniform

Table 5.33 Support Vector Machine Large Uniform

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	33	22	3	11	1	10	23	67	0	0.77
Median	32	22	3	10	1	10	22	68	0	0.78
Max	45	27	2	18	2	10	35	55	0	0.65

Figure 5.31 plots the large uniform case for a support vector machine using mean, median, and max. Table 5.33 contains the numerical analysis of each case with the highest true accuracy attained using the median consensus at 78%.

5.6.6.3 Errors Large Uniform: Local Outlier Factor



(a) Mean (b) Median (c) Max

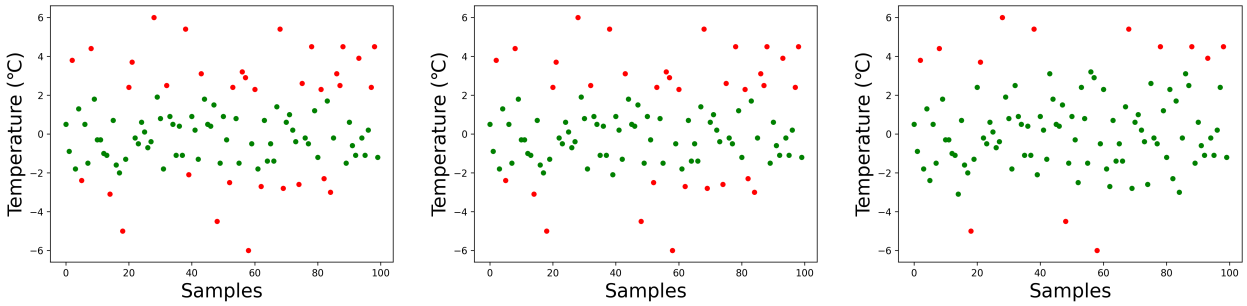
Figure 5.32 Local Outlier Factor Large Uniform

Table 5.34 Local Outlier Factor Large Uniform

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	28	14	2	14	2	10	18	72	0	0.82
Median	20	13	1	7	1	10	10	80	0	0.9
Max	1	0	0	1	1	0	1	89	10	0.89

Figure 5.32 plots the large uniform case using the local outlier factor using the mean, median, and max. Table 5.34 contains the numerical analysis of each case with the highest true accuracy attained using the median consensus at 90%.

5.6.6.4 Errors Large Uniform: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.33 Robust Covariance Elliptic Envelope Large Uniform

Table 5.35 Robust Covariance Elliptic Envelope Large Uniform

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	34	22	3	12	1	10	24	66	0	0.76
Median	33	22	3	11	1	10	23	67	0	0.77
Max	13	10	1	3	1	10	3	87	0	0.97

Figure 5.33 plots the large uniform case using an robust covariance elliptic envelope for mean, median, and max. Table 5.35 contains the numerical analysis of each case with the highest true accuracy attained using the median consensus at 97%.

5.6.7 Errors: Random Frequency Selective

This test case combines different types of features to determine the model classification if the input values are with random selective frequency. This could be the case if an adversary were attempting to send different types of values to see what would and would not be accepted.

Table 5.36 Errors: Random Frequency Selective Summary

(a) Mean

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	20	14	5	6	4	13	7	57	23	0.7
SVM	36	25	5	11	4	16	20	44	20	0.6
LOF	29	15	5	14	4	13	16	48	23	0.61
Elliptic Envelope	37	25	5	12	4	16	21	43	20	0.59

(b) Median

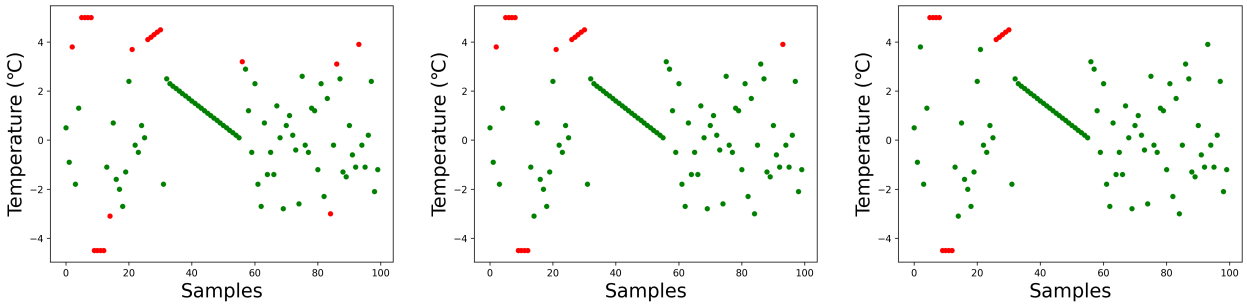
Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	16	12	5	4	4	13	3	61	23	0.74
SVM	33	23	5	10	4	14	19	45	22	0.59
LOF	23	15	5	8	4	14	9	55	22	0.69
Elliptic Envelope	35	24	5	11	4	15	20	44	21	0.59

(c) Max

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Isolation Forest	13	9	5	4	4	13	0	64	23	0.77
SVM	44	30	8	14	4	21	23	41	15	0.62
LOF	2	1	1	1	1	1	1	63	35	0.64
Elliptic Envelope	16	12	5	4	4	13	3	61	23	0.74

In this case, table 5.36 shows the numerical results of testing for each model with mean, median, and max consensus methods. The highest attained true accuracy is through an isolation forest using the max consensus set at 77%.

5.6.7.1 Errors Random Frequency Selective: Isolation Forest



(a) Mean (b) Median (c) Max

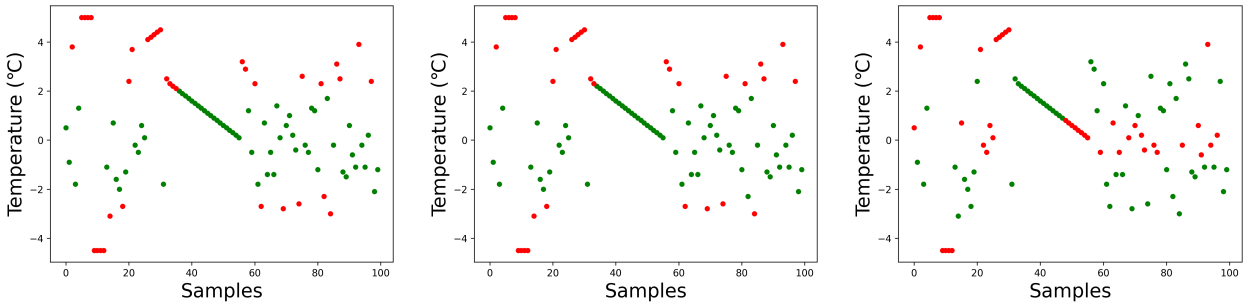
Figure 5.34 Isolation Forest Random Frequency Selective

Table 5.37 Isolation Forest Random Frequency Selective

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	20	14	5	6	4	13	7	57	23	0.7
Median	16	12	5	4	4	13	3	61	23	0.74
Max	13	9	5	4	4	13	0	64	23	0.77

Figure 5.34 illustrates an isolation forest using each of the mean, median, and max consensus sets. In this case, anomalies that fall within the same values as the training set are falsely classified as inliers. Anomalies presenting the same values would not be detected as anomalies. Anomalies that are values outside of the training set, however, are classified as outliers correctly. Table 5.37 presents the numerical results of each case with the highest attained true accuracy using the max consensus method at 77%.

5.6.7.2 Errors Random Frequency Selective: Support Vector Machine



(a) Mean (b) Median (c) Max

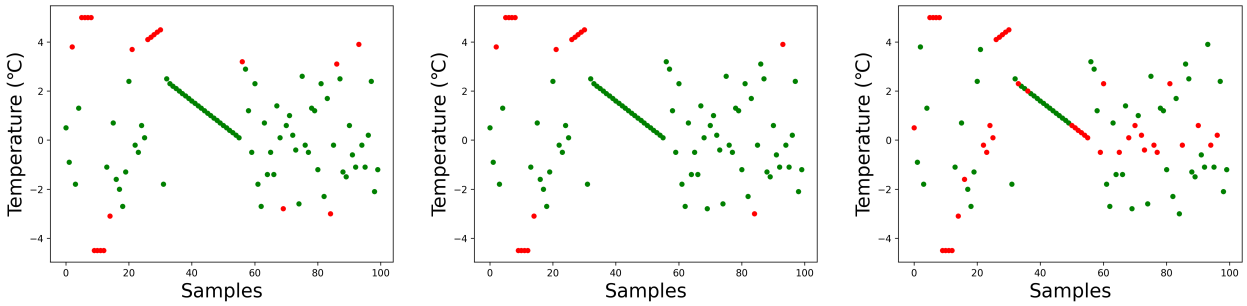
Figure 5.35 Support Vector Machine Random Frequency Selective

Table 5.38 Support Vector Machine Random Frequency Selective

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	36	25	5	11	4	16	20	44	20	0.6
Median	33	23	5	10	4	14	19	45	22	0.59
Max	44	30	8	14	4	21	23	41	15	0.62

Figure 5.35 illustrates a support vector machine using each of the mean, median, and max consensus sets. In this case, max, (Figure 5.35c), detects the most errors for values that fall within the training data set in contrast to the isolation forest (Figure 5.37). Table 5.38 presents the numerical results of each case with the highest attained true accuracy using the max consensus method at 62%.

5.6.7.3 Errors Random Frequency Selective: Local Outlier Factor



(a) Mean (b) Median (c) Max

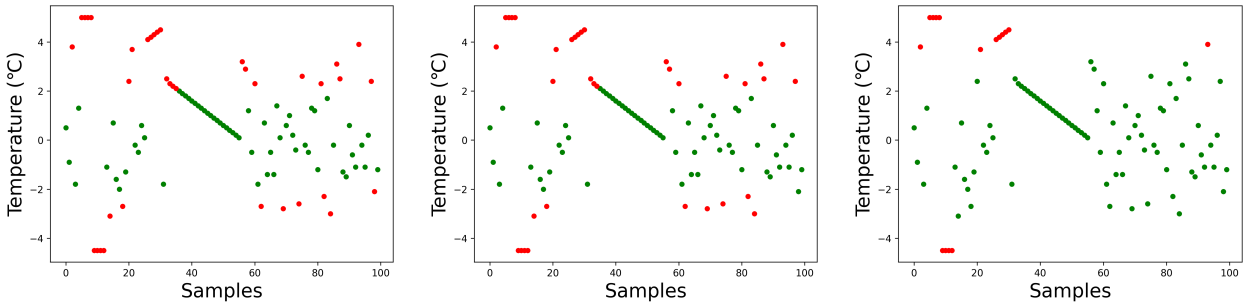
Figure 5.36 Local Outlier Factor Random Frequency Selective

Table 5.39 Local Outlier Factor Random Frequency Selective

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Mean	29	15	5	14	4	13	16	48	23	0.61
Median	23	15	5	8	4	14	9	55	22	0.69
Max	2	1	1	1	1	1	1	63	35	0.64

Figure 5.36 illustrates the local outlier using each of the mean, median, and max consensus sets. Table 5.39 presents the numerical results of each case with the highest attained true accuracy using the median consensus method at 69%.

5.6.7.4 Errors Random Frequency Selective: Elliptic Envelope



(a) Mean (b) Median (c) Max

Figure 5.37 Robust Covariance Elliptic Envelope Random Frequency Selective

Table 5.40 Robust Covariance Elliptic Envelope Random Frequency Selective

Model	**Total**	**Positive**	**Positive Frequency**	**Negative**	**Negative Frequency**	**True Outlier**	**False Outlier**	**True Inlier**	**False Inlier**	**True Accuracy**
Mean	37	25	5	12	4	16	21	43	20	0.59
Median	35	24	5	11	4	15	20	44	21	0.59
Max	16	12	5	4	4	13	3	61	23	0.74

Figure 5.37 illustrates using a robust covariance elliptic envelope using each of the mean, median, and max consensus sets. Table 5.40 presents the numerical results of each case with the highest attained true accuracy, using the max consensus method at 74%.

5.7 Summary

Table 5.41 Machine Learning Model Comparison Summary

Model	Total	Positive	Positive Frequency	Negative	Negative Frequency	True Outlier	False Outlier	True Inlier	False Inlier	True Accuracy
Elliptic Envelope	5439.0	2908.0	194.0	2531.0	145.0	333.0	5106.0	26598.0	93.0	0.8167888888888889
Isolation Forest	1324.0	663.0	144.0	661.0	110.0	287.0	1037.0	30667.0	139.0	0.9194037037037037
LOF	3925.0	1629.0	130.0	2296.0	111.0	225.0	3700.0	28004.0	201.0	0.8306629629629629
SVM	8917.0	4574.0	205.0	4083.0	151.0	336.0	8581.0	23123.0	90.0	0.7349999999999999

Figure 5.41 presents a summary of the numerical analysis by averaging the accuracy of all the tests, using synthetic and real data sets, showing that an isolation forest produces the most accurate classification model at 91.9% overall. In some cases, other models performed similarly or occasionally better. Notably, the Elliptic Envelope, (Figure 5.25), detected the breakout case first and even with false positives. True Outlier Detection: In detecting true outliers, SVM detected the most at 336. False Outlier Detection: The most false outliers were generated by the SVM at 8,581, compared to the Isolation Forest,that only mislabeled 1,037 false messages as outliers. True Inlier Detection: The Isolation Forest detected the most true inliers at 30,667, and the SVM detected the least at 23,123. False Inlier Detection: The SVM generated the least amount of false inliers at 90.

CHAPTER 6

Discussion

In an unauthenticated environment, behavior can be a method to determine the actions of others. This is why modeling the behavior of a vessel provides a layer of monitoring to detect abnormal behavior. Anomalous behavior alerting, gives the crew of a vessel an edge to help maintain a safe operating environment.

The use of temperature sensors is used to illustrate the principle of applying machine learning to model vessel behavior from AIS in real-time on vessels at sea. Future work would consist of also adding additional sensors as features to add degrees of information for machine learning modeling. An example of this type of data would be AIS reported position, along with a vessel's local radar readings, to determine the accuracy of each vessel's location information of what is observed and what is reported. Additional sensors could also increase the accuracy and confidence of a vessel's model.

Currently, AIS has no consensus method, and all AIS receivers display all information as valid. The approach given here seeks to provide a higher degree of confidence than the existing AIS implementation. For a consensus to be formed, and a model to be fit, a few key factors properly need to be in place. First is the need for the original training data to be valid data. If a model is trained on bad data, then anomalies would be built into the vessel behavior. In this case, no proper determination could be made as the model would not correctly identify true inliers or outliers. With any consensus, when fifty-one percent of those participating in data generation agree on a value, a value is selected as the consensus. Future work could study the application of blockchain as a consensus method to determine what vessels are reporting.

Future work could also investigate using machine learning models to influence trust networks built to operate in the maritime domain for vessels to vessel communication. For an example of trust-building, for vessels broadcasting normally, a trust rating is maintained or increased. For vessels broadcasting abnormally, a trust rating is decreased.

Behavior modeling provides an additional layer to a layered security approach. Adding additional sensors increases the number of layers of anomaly detection providing higher levels of confidence in the process. Adding trust, along with behavior models, also adds additional layers of security towards higher levels of confidence and assurance.

CHAPTER 7

Conclusion

Vessels at sea are susceptible to various attacks via AIS that machine learning can help mitigate to provide a safer operating environment. The current maritime vessel communication method, AIS, lacks encryption and authentication leaving the protocol and vessels susceptible to many types of manipulation. By using machine learning to model a vessel's normal behavior, when a vessel is observed to be acting abnormally, it can be identified more easily.

This thesis' provides a novel approach to cross-checking AIS data demonstrates that machine learning behavior modeling can be applied to vessels at sea to increase confidence in AIS. The contributions include; developing a machine learning anomaly detection method for vessels at sea, then analyzing different machine learning methods to select the best method of securing AIS transmission accuracy, finally, designing multiple use cases that challenge the behavioral model built for vessels operating at sea.

Applying machine learning anomaly detection to maritime communication procedures provides a method to allow vessels to identify when a vessel is not operating in a manner consistent with past observations. This allows the crew of a vessel to identify in real-time new situations that should be monitored. Because of the widespread adoption of AIS, many vessels could benefit from implementing behavior modeling into the reception of messages, without the need for the vessels being monitored to be using the same equipment.

REFERENCES

[1] International Maritime Organization. (2019) Automatic identification system. [Online]. Available: http://www.imo.org/en/OurWork/Safety/Navigation/Pages/AIS.aspx

[2] National Marine Electronics Association. Automatic identification systems nema. [Online]. Available: https://www.nmea.org/Assets/nmea%20collision%20avoidance%20through%20ais.pdf

[3] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, Dec 2008, pp. 413–422.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 93–104. [Online]. Available: http://doi.acm.org/10.1145/342009.335388

[5] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995. [Online]. Available: https://doi.org/10.1023/A:1022627411411

[6] P. J. Rousseeuw and K. van Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999. [Online]. Available: http://www.jstor.org/stable/1270566

[7] A. Goudosis and S. Katsikas, "Towards a secure automatic identification system (ais)," *Journal of Marine Science and Technology*, 05 2018.

[8] P. L. Sanchez Gonzalez, D. Díaz-Gutiérrez, T. Leo, and L. Núñez, "Toward digitalization of maritime transport?" *Sensors*, vol. 19(4), p. 926, 02 2019.

[9] I. M. Organization. (2019) Imo profile. [Online]. Available: https://business.un.org/en/entities/13

[10] D. A. R. P. Agency, "Ocean of things aims to expand maritime awareness across open seas," https://www.darpa.mil/news-events/2017-12-06, December 2017, (Accessed on 11/08/2019).

[11] Y. Li, S. Takahashi, and S. Serikawa, "Cognitive ocean of things: a comprehensive review and future trends," *Wireless Networks*, Jan 2019. [Online]. Available: https://doi.org/10.1007/s11276-019-01953-4

[12] D. M. Balduzzi. Ais exposed understanding vulnerabilities and attacks. [Online]. Available: https://www.blackhat.com/docs/asia-14/materials/Balduzzi/Asia-14-Balduzzi-AIS-Exposed-Understanding-Vulnerabilities-And-Attacks.pdf

[13] "Marine traffic," 2019. [Online]. Available: https://www.marinetraffic.com/

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

[15] F. J. Anscombe and I. Guttman, "Rejection of outliers," *Technometrics*, vol. 2, no. 2, pp. 123–147, 1960. [Online]. Available: http://www.jstor.org/stable/1266540

[16] M. Liang, R. W. Liu, Q. Zhong, J. Liu, and J. Zhang, "Neural network-based automatic reconstruction of missing vessel trajectory data," in *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*, March 2019, pp. 426–430.

[17] A. Sidibé and G. Shu, "Study of automatic anomalous behaviour detection techniques for maritime vessels," in *THE JOURNAL OF NAVIGATION*, 2017. [Online]. Available: https://search-proquest-com.proxy.lib.utc.edu/docview/1904076173?OpenUrlRefId=info:xri/sid:primo&accountid=14767

[18] M. Anneken, Y. Fischer, and J. Beyerer, "Evaluation and comparison of anomaly detection algorithms in annotated datasets from the maritime domain," in *2015 SAI Intelligent Systems Conference (IntelliSys)*, Nov 2015, pp. 169–178.

[19] G. Pallotta and A. Jousselme, "Data-driven detection and context-based classification of maritime anomalies," in *2015 18th International Conference on Information Fusion (Fusion)*, July 2015, pp. 1152–1159.

[20] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, no. 12, pp. 2218–2245, Jun 2013. [Online]. Available: http://dx.doi.org/10.3390/e15062218

[21] X. Wang, X. Liu, B. Liu, E. N. de Souza, and S. Matwin, "Vessel route anomaly detection with hadoop mapreduce," in *2014 IEEE International Conference on Big Data (Big Data)*, Oct 2014, pp. 25–30.

[22] E. Osekowska, H. Johnson, and B. Carlsson, "Grid size optimization for potential field based maritime anomaly detection," *Transportation Research Procedia*, vol. 3, pp. 720 – 729, 2014, 17th Meeting of the EURO Working Group on Transportation, EWGT2014, 2-4 July 2014, Sevilla, Spain. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352146514002142

[23] B. H. Soleimani, E. N. De Souza, C. Hilliard, and S. Matwin, "Anomaly detection in maritime data based on geometrical analysis of trajectories," in *2015 18th International Conference on Information Fusion (Fusion)*, July 2015, pp. 1100–1105.

[24] J. Roy, "Anomaly detection in the maritime domain," in *Optics and Photonics in Global Homeland Security IV*, C. S. Halvorson, D. Lehrfeld, and T. T. Saito, Eds., vol. 6945, International Society for Optics and Photonics. SPIE, 2008, pp. 180 – 193. [Online]. Available: https://doi.org/10.1117/12.776230

[25] Z. Hanyang, S. Xin, and Y. Zhenguo, "Vessel sailing patterns analysis from s-ais data dased on k-means clustering algorithm," in *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*, March 2019, pp. 10–13.

[26] F. Mazzarella, M. Vespe, A. Alessandrini, D. Tarchi, G. Aulicino, and A. Vollero, "A novel anomaly detection approach to identify intentional ais on-off switching," *Expert Systems with Applications*, vol. 78, 02 2017.

[27] F. Kandah, J. Cancelleri, D. Reising, A. Altarawneh, and A. Skjellum, "A hardware-software codesign approach to identity, trust, and resilience for iot/cps at scale," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, July 2019, pp. 1125–1134.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: git machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

VITA

Jacob Coleman was born in Nashville, TN, to the parents of Dale and Bonnie Coleman. He is the youngest of two children, with an older brother. He completed a Bachelor of Science in Computer Science at the University of Tennessee at Chattanooga. After taking one year away from his studies, Jacob returned to the University of Tennessee at Chattanooga to complete a Master of Science in Cyber Security.