INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA




Deteção de Comportamentos Anómalos no Tráfego Marítimo

VÍTOR DIOGO COELHO NETO

Outubro de 2023

POLITÉCNICO
DO PORTO

Automatic Maritime Traffic Anomalous Behaviors Detection

Vítor Diogo Coelho Neto

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Graphics, Systems and Multimedia

Supervisor: Gonçalo Valadas
Advisor: Paula Escudeiro

Porto, October 14, 2023

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore the work presented in this document is original and authored by me, having not previously been used for any other end.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, October 14, 2023

Vitor Diogo Coelho Neto

Abstract

Maritime traffic plays a very important role in the world economy, with over 90% of global trading done through naval transportation. The high amount of vessel traffic, mainly due to cargo transportation, leads to several new risks, threats, and concerns, such as increased criminal activity in the sea. The OVERSEE project is proprietary software developed by Critical Software and used by Marinha Portuguesa, Irish Coast Guard, and Papua New Guinea's Coast Guard. The OVERSEE project displays vessel information in real-time through AIS messages, which are mandatory for most cargo vessels to report consistently. Anomaly detection and behavior monitoring tools are computer-based systems that analyse real-time data to detect anomalous behaviors. This project aims to develop a solution capable of detecting anomalous behaviors committed by vessels using AIS messages, which will be reported in real-time automatically via e-mail and the extant OVERSEE graphical interface. The solution is developed with the use of Long Short-Term Memory Recurrent Neural Networks, and a deeper analysis is provided to compare the obtained results with the ideal results. The network training and testing are done with real data, with cross-classification techniques to improve the trustworthiness of the algorithm, hence providing more accurate results.

Keywords: Machine Learning, Neural Networks, Maritime Traffic, Anomalous Behaviors, Microservices, Supervised Learning

Resumo

O tráfego marítimo desempenha um papel muito importante na economia mundial, com mais de 90% do comércio global feito por meio do transporte naval. O grande volume de tráfego de embarcações, principalmente devido ao transporte de cargas, leva a vários novos riscos, ameaças e preocupações, como o aumento da criminalidade no mar. O projeto OVERSEE é um software proprietário desenvolvido pela Critical Software e usado pela Marinha Portuguesa, Guarda Costeira Irlandesa e Guarda Costeira da Papua Nova Guiné. O projeto OVERSEE exibe informações da embarcação em tempo real por meio de mensagens AIS, cuja maioria das embarcações de carga são obrigadas a relatar num período de tempo regular. As ferramentas de detecção de anomalias e monitoramento de comportamento são sistemas baseados em computador que analisam dados em tempo real para detetar comportamentos anómalos. Este projeto visa desenvolver uma solução capaz de detetar comportamentos anómalos cometidos por embarcações por meio de mensagens AIS, que serão reportados em tempo real automaticamente via e-mail e interface gráfica existente do OVERSEE. A solução está desenvolvida com o uso de Redes Neurais Recorrentes[1] de Memória-Curta de Longo Prazo[2]. Uma análise mais profunda é fornecida para comparar os resultados obtidos com os resultados ideais. O treinamento e teste da rede são feitos com dados reais, com técnicas de classificação cruzada para melhorar a confiabilidade do algoritmo, fornecendo resultados mais precisos.

[1] Do inglês *Recurrent Neural Network*
[2] Do inglês *Long Short-Term Memory*

Acknowledgement

I would like to thank everyone involved in making this project possible, namely:

- **My family**, especially my parents and grandparents to whom I owe everything, including the realization of my academic achievements.
- **My supervisor and advisor**, Gonçalo Valadas, and Paula Escudeiro, who guided and helped me through this journey.
- **My work colleagues**, special mention to Eduardo Pacheco, Ricardo Clérigo, Rui Nunes, Catarina Pires and Rui Lopes, without disregarding the rest of my colleagues and friends who also helped me through this journey.
- **My close friends** who have put up with me over the course of the development of this project.
- **My professors over the years** without whom I wouldn't have the knowledge I have and would not be able to achieve the development of this project.

To everyone involved, a sincere **thank you**, this phase of my life will always be remembered, not only by the extreme amount of work I undertook, but also because of everyone who supported me and made me see how important our close ones are in allowing me to keep myself together in times of stress and achieving great things.

Contents

List of Figures

List of Tables

List of Source Code

List of Acronyms

AHP	Analytic Hierarchy Process.
ARPA	Automatic Radar Plotting Aid.
AUC	Area Under ROC Curve.
CNN	Convolutional Neural Networks.
CRC	Cyclic Redundancy Check.
CSW	Critical Software.
DL	Deep Learning.
DTMO	Detection and Tracking of Moving Objects.
EKF	Extended Kalman Filter.
ETA	Estimated Time of Arrival.
FCS	Frame Check Sequence.
FEI	Front End of Innovation.
FFE	Fuzzy Front End.
FN	False Negatives.
FP	False Positives.
GCS	Geographic Coordinate System.
GMSK	Gaussian Filtered Minimum Shift Keying.
GNOME	General NOAA Operational Modeling Environment.
GNSS	Global Navigation Satellite System.
HMM	Hidden Markov Model.
IRCG	Irish Coast Guard.
KNN	K-Nearest Neighbors.
LEO	Low-Earth Orbit.
LMS	Laser Measurement Systems.
LSTM	Long Short-Term Memory.
MAE	Mean Absolute Error.
MAPE	Mean Absolute Percentage Error.
MMSI	Maritime Mobile Service Identity.
MSE	Mean Squared Error.
MSI	Maritime Safety Information.

NCD	New Concept Development.
NCDM	New Concept Development Model.
NN	Neural Networks.
NOAA	National Oceanic and Atmospheric Administration.
NPD	New Product Development.
NPPD	New Product and Process Development.
QEF	Quantitative Evaluation Framework.
QFD	Quality Function Development.
RNN	Recurrent Neural Networks.
ROC	Receiver Operating Characteristic.
RTS	Real-Time System.
SAR	Synthetic Aperture Radar.
SAT-AIS	Space-Based AIS.
SLAM	Simultaneous Localization and Mapping.
STDMA	Self-Organizing Time Division Multiple Access.
SVM	Support Vector Machine.
SWOT	Analysis of Strengths, Weaknesses, Opportunities, and Threats.
TN	True Negatives.
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution.
TP	True Positives.
TWS	Track While Scan.
VC	Value for the Customer.
VDT	Vessel Detection and Tracking.
VNS	Vessel Navigational Systems.
VP	Value Proposition.
VSETP	Vessel State Estimation and Trajectory Prediction.
VTMIS	Vessel Traffic Monitoring and Information Systems.
VTS	Vessel Traffic Services.

Chapter 1

Introduction

1.1 Context

Maritime traffic includes every vessel transporting goods across a water environment, from an origin point to a specific destination. Maritime traffic plays a very important role in the world economy and the overall progress of the modern world, transporting essential goods, raw materials, cars, clothes, luxury products, among others. More than 90% of global trading has been done through naval trading in 2019 and is expected to triple by 2050 [OECD 2023], showing just how important to global trading and even country relationships maritime traffic is. It is estimated that, on the strait of Dover alone, one of the busiest shipping lanes in the world, located between the coast of the United Kingdom and France, between the port of Dover and Calais (also called the Strait of Calais), over 400 vessels carry more than 300 tonnes of goods per day each [Glegg, Jefferson, and Fletcher 2015], bringing the total to more than 120 000 metric tonnes of cargo transported through this strait per day. This humongous amount of traffic represents just a fraction of the entire global shipments, with a total of more than 1,85 billion metric tonnes of cargo being transported worldwide every year [Placek 2022], making the traffic in the Dover Strait just around 2,36% of all the global traffic. With just this small percentage of worldwide traffic, it is already chaotic to manage such a substantial amount of vessels going through the narrow channel each day, making it imperative for the English and French authorities to provide a 24 hour, seven days a week radio and radar safety service for all shipping lanes in the Dover Strait. Such a large amount of data can obscure some important maritime events in the regular traffic patterns, making it difficult to identify and handle some said events such as piracy, at sea encounters, and cargo trafficking, among many others. To monitor this large amount of vessels, the Automatic Identification System (AIS) was introduced to supplement the radar system, which allows for supplementary information to be acquired, both by other vessels, but also by vessel traffic services, which include such examples as the marine coast guard. According to the SOLAS regulation V/19, enforced by the International Maritime Organization, adopted in the year 2000, every vessel is required an AIS to be fitted aboard all ships of 300 gross tonnage and upwards engaged on international voyages, cargo ships of 500 gross tonnage and upwards not engaged on international voyages and all passenger ships irrespective of size. This requirement became effective for all ships by 31 December 2004 [IMO 2019] [IMO 2018]. The OVERSEE project is a proprietary information system software developed by Critical Software (CSW) in cooperation with the Marinha Portuguesa. The software is currently used by big industry names such as the Irish Coast Guard, the Marinha Portuguesa and Papua New Guinea's Coast Guard. The purpose of this software is to support maritime operations of private and government agencies tasked with search and rescue, security, or environmental protection. The solution provides an integrated maritime operational picture, centralising information available under a user-friendly roof, with an interface developed by Critical's UxD team. [CSW 2020] The software uses real-time AIS messages to display the information on the user interface. With this solution comes a problem. The massive amount of information displayed on the user interface, as previously mentioned, often obfuscates illegal activities. The objective of this project is to solve this issue, by designing Machine Learning (ML) software that automatically detects this type of behavior, and warns the user about the problem by making use of CSW's OVERSEE project interface. The approach takes into consideration a microservice architecture approach so that it can be easily integrated with the already extant solution. These concerns, as well as the clear definition of what is an anomalous behavior are addressed in further detail in chapter [2] and throughout the development of this project.

1.2 Motivation

Piracy, criminal activity, and safety risks associated with sea traffic hastily demand a solution, as these pose several national security threats or put lives at risk that require search and rescue missions.

With the advent of tools like ChatGPT [Dowling and Lucey 2023], Dall-E, and other, sometimes controversial tools during late 2022 and continuing up to the point of the publication of this dissertation, Machine Learning, Deep Learning, and other branches of Artificial Intelligence are fast gaining traction, and worldwide spotlights more than ever before.

Following these two very relevant topics, I intend to contribute to the progress of these two fields, using the advancements made in Machine Learning to help automatically detect and assess possible malevolent behavior in maritime traffic. The objective is to contribute not only to the development of these two areas but also to provide a proof of concept to the OVERSEE project so that it can integrate this project down the line.

1.3 Problem

1.3.1 Problem Context

Automatic anomaly and behavior monitoring tools are computer-based systems that analyse vessel position reports in real-time. These tools detect anomalous behaviors in maritime traffic. OVERSEE is an example of a tool used to track vessels in real-time. This tool, however, does not automatically warn the user when criminal activity is being conveyed. The detection of anomalous patters leads to search and rescue, security, and environmental action responsibilities, which are otherwise not covered.

1.3.2 Problem Characterization

The reason the problem depicted in the previous sub chapter [1.3.1] is very important stems from the fact that each day, millions of position reports through AIS messages are portrayed in the OVERSEE interface. The fact that millions of entries are displayed, poses a threat in the sense that anomalous behaviors can be obfuscated, which leads to problems such as increased criminal activity, and the risk of the loss of lives due to accidents.

1.4 Objectives

The main objective of this project is to build an intelligent system capable of detecting anomalous behaviors produced by vessels using real-time data from AIS messages. This system will support the relevant maritime agencies in managing the vast and complex maritime activity occurring daily. The system's architecture will be designed with a micro service pipeline in mind, rather than a monolithic architecture, as it facilitates the encapsulation of the development and testing environments, and make the implementation easier both for the context of this project and the client.

The planned activites include:

- **Objective 1:** Value analysis based on customer needs, expectations and requirements;
- **Objective 2:** Understand the OVERSEE project and its purpose;
- **Objective 3:** Requirements gathering and analysis;
- **Objective 4:** Analyse the current world maritime traffic and position reporting mechanisms and the problems deriving from it;
- **Objective 5:** Identify the anomalous behaviors stemming from maritime traffic;
- **Objective 6:** Understand the differences between monolithic and micro service architectures;
- **Objective 7:** Identification of Machine Learning algorithms that support the identification of anomalous behaviors;
- **Objective 8:** Gather the dataset and label anomalous behaviors;
- **Objective 9:** AIS message decoder development, supporting the storage of the AIS messages' contents in CSV files;
- **Objective 10:** Selecting the optimal algorithm to be used in the development of the solution;
- **Objective 11:** Creating the solution with warning messages for anomalous behaviors and training the algorithm;
- **Objective 12:** Results analysis and algorithm optimization;
- **Objective 13:** Integration with OVERSEE project using a message broker;
- **Objective 14:** Solution and results presentation with validation from target audience;
- **Objective 15:** Improvements based on feedback and future work proposals.

1.5 Used Methodologies

This sub chapter aims to lay out the methodologies that shall be used in the achievement of the objectives set in sub chapter [1.4]. First, a value analysis shall be performed using the knowledge acquired from several references, more importantly, [Koen et al. 2001]. This will include the creation of a Value Proposition CANVAS (Figure [3.5]), a Business Model CANVAS (Figure [3.6]), a SWOT analysis (Figure [3.3]), a TOPSIS evaluation (Sub chapter [3.1.4]), and the Quality Function Deployment (Sub chapter [3.3]), which includes the House of Quality (Figure [3.3.1]). To understand the OVERSEE project, as well as the maritime environment, several conversations were had with the client, Critical Software, to understand the requirements of the solution, as well as the possible risks that can be involved in its development. To understand the theoretical concepts behind micro services, Machine Learning, AIS messages, radar based communications, and anomalous maritime behaviors, several papers were read to understand such concepts. The papers read in this scope were investigated from websites such as Google Scholar, ResearchGate, IEEE Xplore, and ScienceDirect, as well as several other websites where useful information was found. The gathered information is presented in detail in the State of the Art chapter [2].

Developing the solution involves knowledge gathered from the Value Analysis chapter [3], in which the TOPSIS analysis [3.1.4] was performed to identify the most ideal Machine Learning algorithm in implementing the project. Neural Networks was the chosen algorithm as the most ideal for the context of this project. The dataset was provided by the client, and labels were attributed to behaviors where anomalies were identified, which is needed to better train the model. This data is formed by AIS messages which need to be decoded. An AIS messages decoder was also developed in the scope of the project, using Python and the [2023, pyais] library. With the initial solution implemented, the network requires training, and as such, the data is distributed in k-folds [6.1.2] to enhance the generalization of the algorithm. The evaluation phase requires the results obtained from the training of the network to incrementally optimize the network accordingly. The evaluation of the solution will be made using several classic methodologies used in Machine Learning, further discussed in sub chapter [6.2].

With the solution complete and trained, it shall be tested with real time, real data, received from the OVERSEE project through the Kafka message broker. [2.5.2] This allows over one million messages to be sent to the network per second. With this, more results will be produced, which will then be reevaluated following the same line of thought used as mentioned previously, for the initial solution development. However, this evaluation will be different, as it will be performed with the client's approval and validation, to provide a real evaluation by a real customer in the area.

Lastly, with the results gathered, the Quantitative Evaluation Framework [6.2.2] is filled to evaluate the quality of the solution and a conclusion is drawn to understand what was accomplished and what future work can be performed to improve the solution.

1.6 Structure

Note: the structure of this document is subject to change, as this is the first evaluation submission. Some chapters mentioned here may not be present in the current document, or may be incomplete.

This document is divided in four distinct chapters:

- **Introduction:** Serves the purpose of introducing the reader to the problem at hand, the objectives, used methodologies, and a brief contextualization of the envolved subjects. It is subdivided in the following sub chapters:
  - **Context:** Contextualization of the subjects relative to the origin of the problem, as well as a quick brief of the affairs leading to the development of the solution presented to solve the issues mentioned beforehand;

- **Motivation:** Presents the personal, and overall motivations behind the stimulus that triggered the interest in the development of the solution;
- **Problem:** Extends the issues discussed on the Context, more thoroughly explaining the reasoning behind the need to solve the problem;
- **Objectives:** Lists the objectives aimed to be completed with the development of this project;
- **Used Methodologies:** Describes the methodologies used to develop the solution;
- **Structure:** Presents the structure of this document and explanation of the purpose of each chapter and sub chapter.

• **State of the Art:** This chapter refers to the current level of development of the particular field researched in the context of this project, which is characterized by the most up to date developments and information regarding the technologies and processes used for the solution development. This project intends to further contribute to the current state of the art of these technologies. This chapter is subdivided in several sub chapters:

- **Research Methodologies:** Explains the used methodologies to obtain information regarding the relevant topics discussed in the scope of this chapter;
- **Theoretical Framework:** Briefs the user on the framing of the topic relevant for this project in the current world;
- **Radar Based Communications:** State of radar communications' technologies and their use cases;
- **Automatic Identification System:** AIS message further explanation and distinction from radar communications, as well as the need for the development of this technology;
- **OVERSEE Project:** Intends to raise awareness to the user around the project developed by Critical Software that provides a graphical interface for maritime vessel tracking and positioning;
- **Anomalous Behaviors:** Lists and comprehensively explains the different types of anomalous behaviors encountered at sea, which will be later used during the solution development as a source of identification for the True Positives;
- **Machine Learning:** Elucidates the reader about the current methodologies and state of development on the latest machine learning techniques and paradigms;
- **Machine Learning Models and Algorithms:** Expounds the most used algorithms in machine learning as well as some of their particular use cases, which will be useful later on to identify the most relevant algorithm for the development of this particular project..

• **Value Analysis:** Value analysis focuses on analysing the business side of the project, immersing the reader in concepts like the New Concept Development Model, solution value, and the Quality Function Deployment. This chapter aims to understand what the solution provides the client and the scientific community. The subdivisions of this chapter are as follows:

  - **New Concept Development Model:** Explains the Front End of Inovation, the New Concept Development Model, opportunity identification and analysis, which includes the SWOT analysis, idea genesis and enrichment, the idea selection using the TOPSIS evaluation, and the concept definition. In sum, this sub chapter focuses on generation the ideas to develop the solution as well as its selection and initial concept definitions;
  - **Value:** This sub chapter focuses on identifying the value the solution has for the client, as well as the scientific community, using the Value Proposition CANVAS and the Business Model CANVAS;
  - **Quality Function Deployment:** The QFD is a technique that helps focus on the main client's requirements and helps relate them to the respective technical requirements. These relationships can be visualised with the House of Quality (Figure 3.7).

- **Experimentation and Evaluation:** This chapter is aimed towards defining the experimentation protocol, i. e. how the obtaining of results for further analysis shall proceed. This chapter also intends to explain how the solution's results, and the solution as a whole, are going to be evaluated in terms of quality and result's quality.
  - **Experimental Design:** This sub chapter defines the hypothesis, experimental and control conditions, the test apparatus and the experimental protocol, which helps in understanding how the results and data are obtained;
  - **Solution Evaluation:** With the solution evaluation, the intention is to improve the results in each iteration of the solution development lifecycle, helping to understand what can be improved. To evaluate this, several techniques are used, such as the F1 score calculation, ROC Curve analysis, and the creation of the Quantitative Evaluation Framework, to assess the quality of the solution as a whole;
  - **Sources of Information:** Explains the sources where information was gathered to address this particular chapter's underlying concerns.

Chapter 2

State of the Art

This chapter studies the highest level of development, achievement and progress in the particular fields and areas of study relevant to this project, which include all of those mentioned in the subsequent sub chapters under this particular chapter. The State of the Art implies that the most advanced and effective techniques, methods, or technologies are being used to solve the particular problem or to advance in these particular fields. In the context of research, "state of the art" refers to the current level of knowledge and research on a particular topic, including the most recent and influential studies, theories, and approaches. It is often used to describe the background and motivation for a new research project, as well as to position the project in relation to the underlying research in the field.

Sub chapter [2.2] gives an overall idea of where the information that is being researched for the purpose of this project stands in the general computer engineering scene. Sub chapter [2.1] informs about what research methodologies were used to gather information in the development of this chapter, such as the used keywords and websites used to search for information. Sub chapter [2.3] elucidates the reader regarding the current state of radar based communications and its uses. Sub chapter [2.4] explains what AIS messages are in the context of maritime traffic as well as the current laws being enforced regarding the obligatoriness of complying with them. In [2.5] the OVERSEE project is explained, as well as its purpose, current state, clients, and future market expansion possibilities. The [2.6] explains in further detail what is distinguished as an anomalous behavior in maritime traffic. The last two sub chapters, [2.8] and [2.9] explain in further detail the current state of Machine Learning and the more specific algorithms that are most commonly used when solving problems that require them.

In short, state of the art refers to the current level of knowledge, development, and innovation in a particular field or technology, which serves as a reference point for future research and development, and this chapter intends to dive in the particular topics mentioned above to further explain their current states.

2.1 Research Methodologies

The first step in obtaining trustworthy data for the development of this particular chapter, is to define what keywords need to be searched for each particular topic, and where those keywords should be input to obtain the answers sought after for this purpose. Each relevant sub chapter has their own keywords as follows:

- **Radar Based Communications:** radars, radar communications, radar based communications;

- **Automatic Identification System:** Automatic Identification System, Maritime Traffic, SOLAS regulations, AIS decoders;
- **OVERSEE Project:** Critical Software, OVERSEE, Marinha Portuguesa;
- **Anomalous Behaviors:** Anomaly detection, anomalous behaviors, Maritime Traffic;
- **Machine Learning:** Machine Learning, Artificial Intelligence, Deep Learning, Neural Networks, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Machine Learning Models, Machine Learning Evaluation, Machine Learning Algorithms, Machine Learning Cross Validation, Machine Learning Recall, Machine Learning Precision, Machine Learning F1 Score, ROC, AUC;
- **Machine Learning Models and Algorithms:** Machine Learning, Deep Learning, K-Nearest Neighbors, Neural Networks, Support Vector Machine, Hidden Markov Model, Long Short-Term Memory, Gradient Boosting, Autoencoder, Clustering, Naive Bayes Classifier, Linear Regression.

To search these keywords, both Google and Google Scholar were used. When using Google Scholar, the websites with the most results, and ultimately, the most used websites to research for information regarding this chapter were ResearchGate, ScienceDirect, and IEEE Xplore.

2.2 Theoretical Framework

This chapter places this project inside a theoretical standpoint. In the **Introduction**, the overall concepts were laid out, and some initial context was shined to elucidate the reader in the sense of understanding what this dissertation's purposes are and which topics are discussed.

In this sense, this chapter more specifically places this project inside its relative and relevant theoretical standpoints and demonstrates the current states of each of the subject matters. It is also significant to explain each topic's advantages, disadvantages, safety concerns, and limitations, to better guide the selection of the relevant tools and frameworks in the final solution's design.

2.3 Radar Based Communications

With the increased demand for maritime transportation, safety and security are ever more important. Radio-based communications is a system for voice and data communications between ships, ports, and other maritime stakeholders. Both systems are critical for safe and efficient maritime navigation and communication, but they serve different purposes and have different characteristics. Vessel Detection and Tracking (VDT), together with trajectory prediction and Estimated Time of Arrival (ETA), are important tasks for Vessel Navigational Systems (VNS). [Perera, Oliveira, and Soares 2012] Vessel Traffic Monitoring and Information Systems (VTMIS) are of utter importance in improving, and maintaining maritime safety and security. The first radar system was proposed in 1920, for ship collision prevention. The track-while-scan (TWS) radar systems allow for the tracking of multiple targets under dynamic conditions, such as weather, new ships comming in bound, or differences in radar coverage. [Perera, Oliveira, and Soares 2012] Stemming from these conditions, frequent calibrations are required to the radar systems to avoid their fast degradation. Laser-based systems predict the overcoming of some of the problems that are faced by the radar system. Laser Measurement Systems (LMS) targets are approximated by clusters of data points, and detection and tracking tasks are permitted. Results of these identification systems can present not only other vessels, but can also be used to identify stationary objects, such as rocks. [Perera, Oliveira, and Soares 2012] Automatic Radar Plotting Aid (ARPA) systems provide "accurate information on range and bearing of nearby vessels, and the Automatic Identification System (AIS) (discussed further in [2.4]) is capable of giving information on the vessel structural data, position, course, speed, etc." [Perera, Oliveira, and Soares 2012] Even though these aids are extremely important in VTMIS and VNSs, some more advanced tasks, such as navigational decision, and, as is being discussed in this dissertation, the detection of anomalous behavior patterns, are still very underdeveloped as of the development of this paper. [Perera, Oliveira, and Soares 2012] [Corucci et al. 2012] [Scotti et al. 2015]

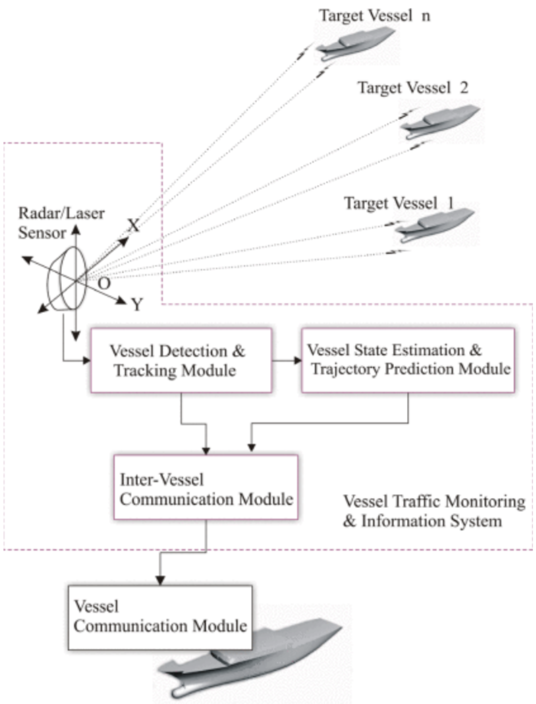


Figure 2.1: VTMIS according to [Perera, Oliveira, and Soares 2012]

The VTMIS presented in Figure [2.1] consists of three main modules:

- The **VDT** module;
- The **Vessel State Estimation and Trajectory Prediction** (VSETP) module;
- The **Intervessel Communication** module.

The VDT module is formed by a Neural Network (NN), designed to handle multiple VDT processes. The VSETP module consists of an Extended Kalman Filter (EKF) which formulates the vessel's state, consisting of the position, velocity and acceleration, as well as the navigational trajectory prediction. [Perera, Oliveira, and Soares 2012] The EKF is used due to its capabilities of "fusing nonlinear system kinematics with a given set of noisy measurements" [Perera, Oliveira, and Soares 2012]. Finally, there is the Intervessel Communication module, which permits the communication among the VTMIS. It is worthy to keep in mind that VTMIS include not only vessels equipped with radar based communication capabilities, but also terrestrial VTMISs, such as radar towers, and other infrastructures such as coast guards and navies. Other equipment capable of receiving radio transmissions from other VTMISs include also aerial means, such as helicopters and airplanes, which are often used for search and rescue operations. The position reports consist of the standard Geographic Coordinate System (GCS), which uses latitude and longitudinal values to reference the position of the VTMIS.

Among radar based communications, there have been recent developments in the target detection and tracking, which consist of two important tasks, the Detection and Tracking of Moving Objects (DTMO), and Simultaneous Localization and Mapping (SLAM). DTMO helps avoid collision situations, and the SLAM helps locate the present position and orientation of the navigation system. [Perera, Oliveira, and Soares 2012]. These, however, will not be deeply researched in this dissertation, as they are not in the scope of the project and would extend this section too much.

Regarding radar based communications, there is also ship detection and tracking systems using passive radars. Passive radars generally use two channels, namely the reference channel and the target channel. The [Corucci et al. 2012] The reference channel captures the direct signal of the transmitter which is then compared to the target return, carried out by cross-correlating the reference and the target signals. [Corucci et al. 2012]

In radar based communication networks, photonics based radar networks are composed of a central unit, and two radar peripherals. The main difference between classic radar communication and photonic radar networks is the underlying technology used to transmit and receive signals. Classic radar communication uses electromagnetic waves, which are transmitted and received through antennas, while photonic radar networks use light, which is transmitted and received through optical fibers or free-space optical links. [Serafino et al. 2022]

In comparison to conventional radar communication, photonic radar networks have the potential to provide higher bandwidth, greater precision, and superior tolerance to interference. They may also be lighter and more compact, making them suited for usage in situations where size and weight are crucial considerations. However, photonic radar networks are still in the research and development stage and require more sophisticated and expensive equipment than conventional radar communication. [Serafino et al. 2022]

In summary, radar-based communications are a wireless technology that transmit data using radar emissions. It has the potential to deliver high data rates across considerable distances, especially in difficult situations such through walls and inclement weather. The system transmits and receives data using millimeter wave frequency-operated radar transceivers. The high bandwidth of radar-based communication, which enables the transmission of significant volumes of data quickly, is one of its benefits. It is also a low-power technology, making it suitable for use in battery-operated devices. However, the technology still needs further research to optimize its performance and reliability.

2.4 Automatic Identification System

The Automatic Identification System, from now on referred as AIS, is the standard communication means used nowadays by vessels to broadcast information about their state, which includes their position, velocity, identification, ETA, origin and destination positions. [Sciancalepore et al. 2022] According to the SOLAS regulation V/19, enforced by the International

Maritime Organization, adopted in the year 2000, every vessel is required an AIS to be fitted aboard all ships of 300 gross tonnage and upwards engaged on international voyages, cargo ships of 500 gross tonnage and upwards not engaged on international voyages and all passenger ships irrespective of size. This requirement became effective for all ships by 31 December 2000. [IMO 2019]

Some advantages of the AIS systems include the fact that they are not severely impacted by weather conditions, they are cheap to implement real-time data, and they are freely accessible for everyone. [Farahnakian, Heikkonen, and Nevalainen 2022]

This system also allows for the self-separation and the exchange of short audio messages. As of 2022, over 1 644 000 use AIS regularly. [Sciancalepore et al. 2022] The fact this technology was mainly developed during the early 1990s, and the fact they are sent over in clear-text and not authenticated, leaves a major vulnerability where any malicious intention to forge AIS messages can be carried out, leading to potential threatening scenarios such as leading ships to wrong routes, or preventing shups to rescue distressed vessels. [Sciancalepore et al. 2022] AIS messages are acquired using regular Global Navigation Satellite Systems (GNSS). This feature enables the identification, route adjustment, accident prevention, remote tracking, and search and rescue operations to be carried out, allowing two vessels to establish a one-to-one communication channel, which allows them to exchange dedicated binary messages, which translate to audio messages. AIS transceivers are also mounted within ports and other naval infrastructures, such as coast guard, and navy infrastructures, to enhance ship tracking and allow for search and rescue operations. [Sciancalepore et al. 2022]

AIS communications use the Very High Frequency (VHF) band, specifically, the 161.975 MHz (Channel A) and 162.025 MHZ (Channel B). [Sciancalepore et al. 2022] "The modulation scheme is the Gaussian filtered Minimum Shift Keying (GMSK), with a bit rate of 9,600 bits/sec. The theoretical maximum transmission range that can be reached by commercial AIS transceivers is approximately 70 km, even if weather factors can reduce it to 40 km." [Sciancalepore et al. 2022] The AIS system uses the Self-Organizing Time Division Multiple Access (STDMA) technology to meet the high broadcast rate. [Farahnakian, Heikkonen, and Nevalainen 2022] To try and overcome this issue, Low-Earth Orbit (LEO) satellites have been fit with AIS transceivers, leading to Space based AIS (SAT-AIS) communication paradigm, which allows the extension of the transmission range up to 400 km. [Sciancalepore et al. 2022]

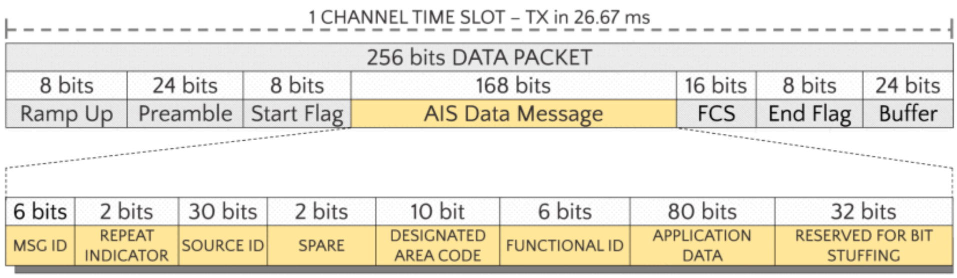


Figure 2.2: AIS message structure according to [Sciancalepore et al. 2022]

AIS messages are 256 bits and start with the first 8 bits reserved to turn the transceiver on. [Recommendation 2014] The following 24 bits are reserved for the *Preamble*, which are in turn used to identify an upcoming AIS message, and to synchronize with the symbols emitted by the transmitter. The following 8 bits are the *Start Flag*, which are used to indicate the beginning of the actual AIS message, which in turn has a size of 168 bits. The *AIS Data Message* itself can be subdivided in several parts. In Figure [2.2], an example of a type 8 AIS message is evidenced, which is a binary broadcast message. Other types of AIS messages can be found in the United States of America Department of Homeland Security, defined per ITY-R M.1371 and IEC Standards in [USCG 2023]. The *AIS Data Message* has a 6 bits *MSG ID* which contains the message type, the *Repeat Indicator* with 2 bits of information, indicating how many times a message has been retransmitted, the *Source ID* Maritime Mobile Service Identity (MMSI) with 30 bits of information uniquely identifying the transceiver emitting the message, the *Spare Field* with 2 bits is a placeholder for future information. The *Designated Area Code* of 10 bits is the jurisdiction code, the *Functional ID* of 6 bits specifies the message sub-type, the 80 bits *Application Data* is dedicated to binary data, and, at last, the *Reserved for Bit Stuffing* area of 32 bits is devoted to bit stuffing. Outside the *AIS Data Message* section, the *FCS* (*Frame Check Sequence*) field provides error detection via the Cyclic Redundancy Check (CRC) polynomial of 16 bits. The *End Flag* indicates the end of the message's transmission, and the *Buffer* field, with 24 bits of data, is adopted for bit-stuffing, distance delay, and syncrhonization jitter. [Sciancalepore et al. 2022] The transmission of an AIS message, from the beggining of the preamble to the end flag, is always less than 26,667 ms. [Sciancalepore et al. 2022]

"Each AIS transceiver periodically acquires the knowledge of the AIS entities in the radio neighborhood, by passively listening to one of the two AIS channels for 1 minute", [Sciancalepore et al. 2022] which is divided in 2250 slots of 26,667 ms. Standard indicates that an AIS transceiver can continuously transmit data for a maximum number of consecutive slots, based on the message class. While class A transceivers can transmit continuously for 5 slots, class B devices can occupy a maximum of 3 slots. [Sciancalepore et al. 2022] [USCG 2023]

In summary, AIS is a system for vessel tracking and identification, both between vessels, ports, and other maritime infrastructures. These systems are critical for safe and efficient maritime navigation and communication and are even mandatory to some specific vessels according to the relevant regulations and enforced laws.

2.5 OVERSEE Project

Important note: some of the information presented in this sub chapter does not have a reference as it comes from the writer of this thesis' personal experience in working with the project at Critical Software.

According to [CSW 2020], OVERSEE is an award-winning project developed by Critical Software in partnership with the Marinha Portuguesa. Navies, coast guards, and other maritime agencies have had access to several vessel tracking and monitoring systems for a very long time, however, these systems are poorly integrated with user experiences that often leave a lot to be desired. Without a good overlook to what is happening at sea, it becomes difficult to appropriately respond to complex situations.

OVERSEE solves the above mentioned issue by providing integrated maritime operational picture, bring together a wide array of data sources together in a user-friendly user interface, [2.3]. OVERSEE provides not only the interface with the real-time vessels display, but also provides Maritime Safety Information (MSI) records, personalized alarms for example for a specific vessel tracking, an alerts dashboard, containing distress calls sent by vessels that require assistance, incidents report with its respective dashboard, and a dashboard to control search and rescue missions.

One of the secrets behind the success of the OVERSEE project stems from its customer-oriented approach, where surveys are regularly conducted among the relevant stakeholders to understand the nature behind maritime agencies problems and concerns. Besides this, CSW also kept in mind the legislative, industry and best practice requirements as well as all the relevant user interface standards. [CSW 2020]

OVERSEE has countlessly proven its concept by providing maritime agencies' the ability to rapidly respond and allocate and task resources to emergencies. Apart from Marinha Portuguesa, OVERSEE is used by other entities such as the Irish Coast Guard (IRGC), supporting their efforts to handle 2600 emergencies per year. [CSW 2020] In the first year of OVERSEE's deployment, the IRGC was able to save 405 lives with its help, an increase of a third in comparison to the previous year. Another substantial benefit provided by OVERSEE in this time span was the ability to reduce IRCG's wasted resources by successfully identifying 15% of the incidents as false alarms. [CSW 2020]

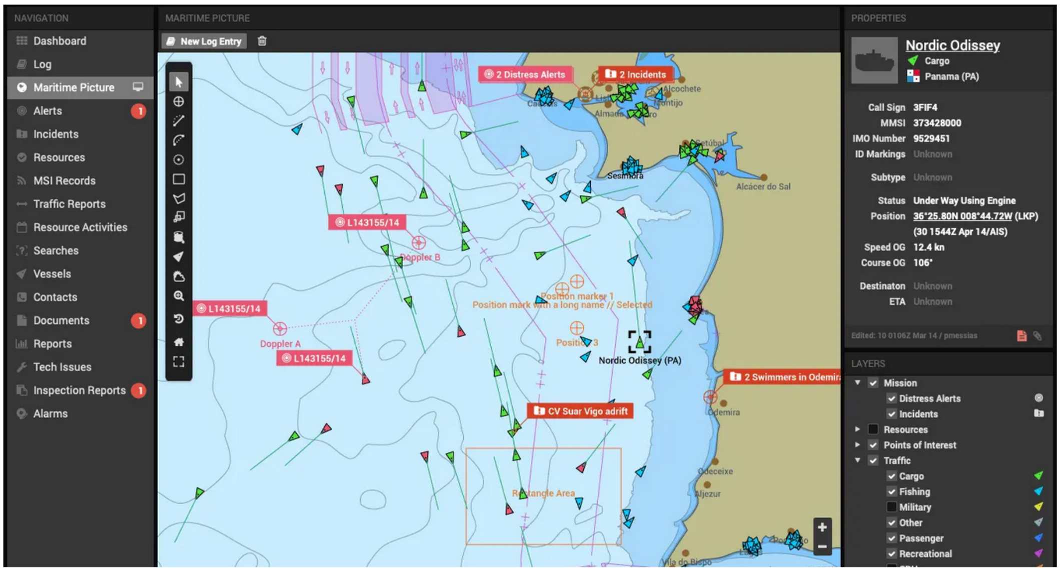


Figure 2.3: The OVERSEE project's interface

The above Figure [2.3] pictures the user interface used by the tool. In the picture it is possible to distinguish some of the features such as the real-time view of vessels along the Portuguese coast, as well as the identification of a specific vessel on the right panel.

Similar applications provide real time maritime traffic overview, such as MarineTraffic [MarineTraffic 2023], which have their own coastal receiving stations. Another example of an application that also gathers these kind of information, and reports, for example, shipping delays, is Windward AI [WindwardAI 2023].

The main advantage OVERSEE has over these applications is its customizability and the fact that it is taylor made to the specific client it was intended to, in this case, both Marinha Portuguesa and the Irish Coast Guard. It also provides many more specific details these other applications do not provide, such as air traffic, live video feed for some vessels, custom made rules to detect patterns in some vessels, vessel's position history, hydrography, as well as the previously discussed capabilities OVERSEE provides.

Figure 2.4: Maritime Operations Centre in Portugal using a release version of OVERSEE

OVERSEE has many services including **Search and Rescue (SAR)**, which is the search for and provision of aid to people who are in distress or imminent danger. The general field of search and rescue includes many specialty sub fields, typically determined by the type of terrain the search is conducted over. These include mountain rescue; ground search and rescue, including the use of search and rescue dogs; urban search and rescue in cities; combat search and rescue on the battlefield and air sea rescue over water. [Personal Communication, May, 2019] SAR operations are regulated by **Safety of Life at Sea (SOLAS)**.

The **Incident Command System (ICS)** is a management tool growing in international use for managing any emergency event. It consists of procedures for organizing personnel, facilities, equipment, and communications at the scene of an emergency. ICS is intended to quickly blend numerous organizations into an effective response organization for any type and magnitude of emergency. ICS is a highly flexible concept for managing emergency events involving multiple jurisdictions and multiple agencies, such as major disasters or events involving hazardous materials. Similar systems should be used when ICS is not available. [Personal Communication, May, 2019]

As previously mentioned, OVERSEE has **Vessel traffic services**. VTS are shore side systems which range from the provision of simple information messages to ships, such as position of other traffic or meteorological hazard warnings, to extensive management of traffic within a port or waterway. Generally, ships entering a VTS area report to the authorities, usually by radio, and may be tracked by the VTS control centre. Ships must keep watch on a specific frequency for navigational or other warnings, while they may be contacted directly by the VTS operator if there is risk of an incident or, in areas where traffic flow is regulated, to be given advice on when to proceed. SOLAS Chapter V (Safety of Navigation) states that governments may establish VTS when, in their opinion, the volume of traffic or the degree of risk justifies such services. [Personal Communication, May, 2019]

Maritime inspections are the verification of the implementation of the EU maritime safety and security legislation, which remains an essential task of the Portuguese Navy. There are several reasons for verifying how this legislation is implemented in practice, including: detecting gaps in the overall safety system; promoting a harmonised approach across the European Union; and improving the efficiency and effectiveness of the measures in place. Inspections are carried out on the basis of separate EU legal acts relating to the relevant area (standards for seafarers, maritime security, recognised organisations). The formula of the inspections depends on the area of activity. [Personal Communication, May, 2019]

Envorinmental Protection permits the detection of sea pollution. **Oil spills** are the release of a liquid petroleum hydrocarbon into the environment, especially the marine ecosystem, due to human activity, and is a form of pollution. The term is usually given to marine oil spills, where oil is released into the ocean or coastal waters. Trajectory maps can be produced using a NOAA (National Oceanic and Atmospheric Administration) developed computer model called GNOME (General NOAA Operational Modeling Environment), which helps to predict the movement of oil. GNOME can forecast the effects that currents, winds, and other physical processes have on the movement of oil in the ocean. During an oil spill, this model is updated daily based on field observations, aerial surveys, and new forecasts for ocean currents and winds. [Personal Communication, May, 2019]

The above mentioned functionalities are just some of the most important capabilities OVERSEE presents. For more information regarding this subject further reading of manuals [Personal Communication, May, 2019] and [Personal Communication, October, 2021] recommended.

2.5.1 Microservices

In the context of this project, it is relevant to talk about microservices as the developed solution will integrate the OVERSEE project using a microservice approach.

Microservices focus on software modularization. Prior to the introduction of the microservice architecture, software ws developed in a monolothic fashion. The term monolith implies all components are part of a single unit. In monoliths, everything is developed, deployed, and scaled as 1 unit. [Wolff 2016] The problems that arise with this type of architecture include the fact that teams need to be careful to not affect each other's work, every update the entire application needs to be redeployed, which implies serious efficiency concerns, and with this type of architecture, the entire application is much more prone to errors, with one error having the possibility to incapacitate the entire monolithic solution. [Janashia 2022] These problems bring more adversities, such as the entire release process taking longer, the entire application needs to be built and deployed each update, the entire application needs to be tested all over again, and the modules are all tangled, possibly creating problems.

With the increase in software complexity, a better solution was required, hence the rise of microservice architecture. Microservices are, simply put, a monolithic application broken down into smaller, independent services, or modules. [Janashia 2022]

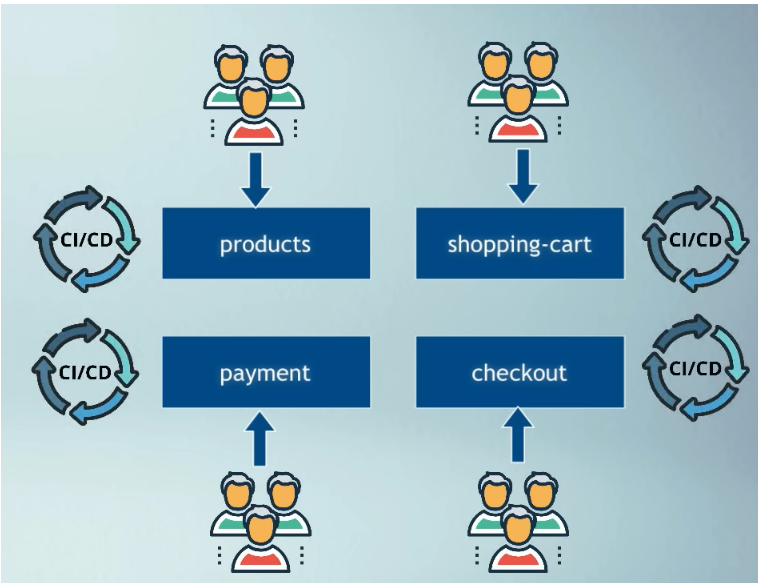


Figure 2.5: Example of an e-commerce software application using microservice architecture (retrived from [Janashia 2022])

In the above Figure [2.5], an example of an e-commerce software application using microservice architecure can be seen. Each module is split based on business functionalities and is self-contained and independent. The communication is done via API message calls, which are queued in message brokers (further explained in sub chapter 2.5.2). This allows teams to work independently, each being able to choose their own services' architectures and tech stacks. [Janashia 2022]

However, microservices also brought new challenges with them. The fact that microservices are distributed systems introduces a new layer of complexity, with each having their own repositories, databases, and development pipelines, making it difficult to monitor services distributed across different distributed servers. This, however, can be partly overcome with the recent introduction of tools that attempt to solve these orchestration issues, such as Kubernetes, an orchestration application developed by Google where multiple services deployed in containers can be managed through a user interface.

Microservices can have two kinds of repositores [Janashia 2022]:

- **Monorepo:** involves one repository that contains all the services related to one project;
- **Polyrepo:** each service is deployed on its own repository, emphasizing modularization.

Polyrepo has the advantage of further encapsulating each service with each one having their own repository but is harder to maintain, where **Monorepo** is easier to maintain, but projects are all maintained on the same repository which can defeat the purpose of a microservice by increasing the logic necessary to make sure only the relevant service for each update is built and deployed. [Janashia 2022]

Designing the correct microservice architecture has its challenges. One of the first challenges that ensures is what can be defined as a service within a project? To answer this question, [Wolff 2016] says that multiple subjects must be taken into account, namely the level of modularization intended, the technology used, the distributed communication used, and the team size.

In the scope of this project, microservices are very relevant as they allow for the seamless deployment of the new service being deployed without affecting the already developed services for the monolithic infrastructure.

2.5.2 Message Queue Pipeline Using Apache Kafka

Kafka is a distributed platform which runs on a cluster that can span multiple servers and data centers. [Vyas et al. 2022] Kafka was developed by LinkedIn in 2011 in an open-source fashion. [Shree et al. 2017] Kafka is a near real-time streaming distributed platform and has a producer API that works as a source for streaming services, which means it can receive data from heterogeneous sources. [Vyas et al. 2022] The application runs on a **broker** and consists of **"Topics"**. A topic is an object within the Kafka architecture that harnesses the data from the producer, to which the consumer subscribes, receiving the data in near real-time, allowing for over one million messages to be sent per second. Each topic is broken down in **"Partitions"**.

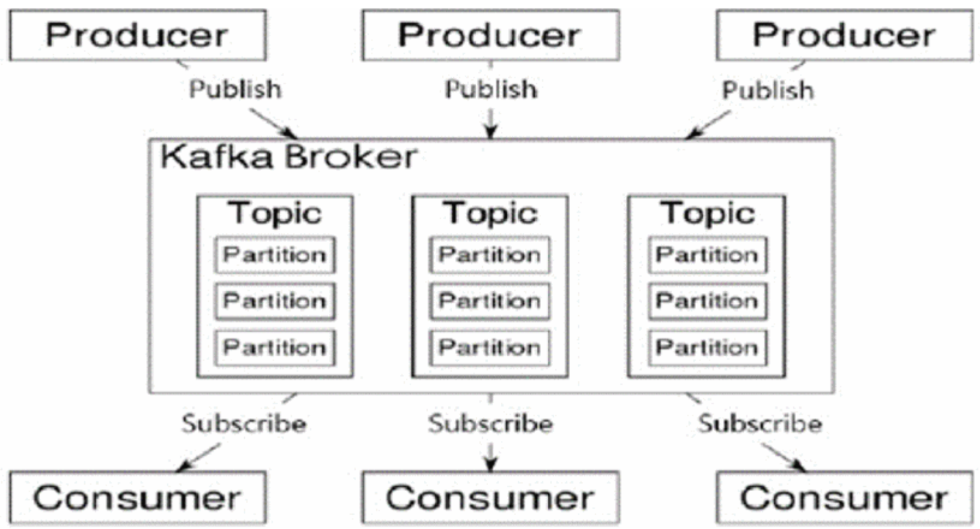


Figure 2.6: Kafka Architecture (retrieved from [Bang et al. 2018])

A partition is an ordered, immutable sequence of records that can be thought of as a commit log. Each partition is a collection of messages that are ordered by their offset, and once written to a partition, messages cannot be deleted or modified. A Kafka topic is divided into one or more partitions, and each partition is replicated across multiple brokers for fault tolerance. [Vyas et al. 2022] The number of partitions in a topic can be set at the time of creation, and it determines the maximum level of parallelism for consumers that read from the topic.

Producers write messages to a topic, and Kafka assigns the message to a specific partition based on the partitioning strategy specified by the producer. Consumers read messages from one or more partitions, and each consumer is assigned to a specific partition or set of partitions to read from. Consumers can read from a specific offset in a partition, and they can keep track of their progress by storing their offset in a separate Kafka topic called a "consumer offset topic". [Shree et al. 2017]

In summary, Kafka allows for the communication between the producer, in this case study, the OVERSEE project, to send partitioned messages in topics to the consumer, in this case, the microservice being developed. This is extremely important as Kafka allows for an extremely high number of messages to be sent each second, crucial in analysing such a large dataset that is being streamed by OVERSEE towards the microservice at hand.

2.6 Anomalous Behaviors

An anomalous behavior "can be described as a behavior that is not “normal” or, more specifically, not expected to occur during regular operation". [Wolsing et al. 2022] The objective of defining these anomalous behaviors is to clearly identify without ambiguity what an anomalous behavior should be considered in the context of maritime traffic. The following sub chapters will analyse with increased detail each individual behavior that is to be considered an anomalous behavior in the context of AIS maritime traffic.

2.6.1 Incorrect Anchorage

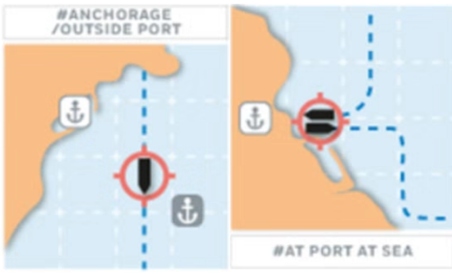


Figure 2.7: Incorrect Anchorage (image provided by CSW)

Anchorage outside port is an illegal activity and shall be considered as an anomalous behavior in the scope of this project. Anchorage is the activity of setting anchor, immobilizing the vessel, which is an activity that shall only be done at allowed ports.

In the left image of Figure [2.7] this behavior is evidenced by the anchorage of a vessel outside the allowed port, which is indicated by the white anchor. Instead, the vessel is anchored at an illegal position, indicated by the blue anchor icon.

Similar to anchorage outside port, setting anchor too close to another vessel, with the incorrect orientation, or on forbidden ports is also an illegal activity and shall be considered an anomaly. In the right image of Figure [2.7], the vessel is both anchored too close to another vessel, and is anchored in the incorrect orientation.

2.6.2 Drifting

Figure 2.8: Drifting Vessel (image provided by CSW)

Drifting is the behavior when a vessel is moving towards an orientation, but the reported orientation mismatches in the AIS message. An example of this is when a vessel is moving towards NE, as reported by its consecutive position reports in the AIS messages, but its orientation along these reports has been miss matching this orientation, for example if it is moving towards northeast, but its orientation is reported as facing north.

2.6.3 At Sea Encounter

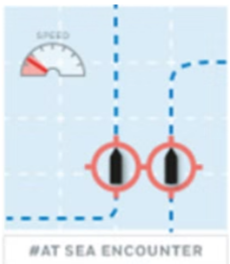


Figure 2.9: At Sea Encounter (image provided by CSW)

An encounter at sea is described by when two ships either get too close to each other while moving, or when they stop by each other at the middle of the sea. This activity is illegal because there is the possibility that they may be exchanging cargo, and potentially even illegal cargo, which is obviously considered a criminal activity, hence an anomalous behaviour. Moving too close to each other is also considered an anomalous behavior because of the danger of bumping into each other which can cause hull damage and subsequent safety concerns.

2.6.4 Position Report Anomalies

Position spoofing is, as the name indicates, the act of spoofing one's position, i. e. faking their actual position. This is considered illegal as it compromises the credibility of the actual position reports. This activity poses serious danger as it causes several safety concerns, such as the possibility of vessel colision. This can be evidenced in the left image of Figure [2.10]

Figure 2.10: Position Spoofing (image provided by CSW)

Not reporting the vessel position in the indicated time frame is considered an anomalous activity, as according to SOLAS regulation [IMO 2019], every relevant vessel shall be equipped with an AIS equipment capable of reporting their state at an indicated interval. Not doing so is considered an illegal activity, and therefore an anomalous behavior as evidenced on the right image of Figure [2.10]

2.6.5 Inside or Outside Restricted/Time Restricted Area

Some restricted areas at sea have rules that indicate where vessels are and are not allowed to transit. Entering these areas, or, in case of area restricted vessels, leaving these areas, is illegal and shall be considered an anomalous behavior.

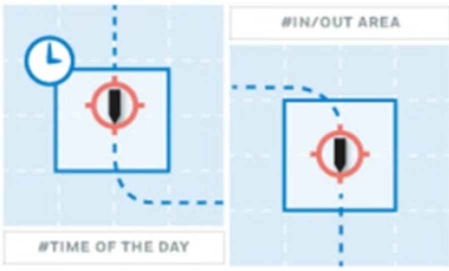


Figure 2.11: Inside/Outside Restricted Area (image provided by CSW)

Similar to the time restricted areas, some areas cannot be entered/left by vessels at particular times of the day, or during specific events. This is an anomalous behavior.

2.6.6 Sudden Change of Heading

Figure 2.12: Sudden Change of Heading (image provided by CSW)

Changing the vessel's heading, or orientation, suddenly, and without any proper justification, is considered a suspicious activity and is considered an anomalous behavior. A sudden change of orientation is considered when the vessel changes its orientation by over 45 degrees.

2.6.7 Sudden Change of Speed

Figure 2.13: Sudden Change of Speed (image provided by CSW)

A sudden change of speed is considered an anomalous behavior as it can possibly also relate to other types of anomalous behaviors, such as position spoofing. A sudden change of speed can also mean the vessel is running from something, which can both mean they are practicing criminal activity or are in distress.

2.6.8 Shore Related Anomalies

Heading to shore or off shore too quickly raises suspicion as it is often related to drug activity. Drug vessels are often reckless and excess in speed both in heading to and off shore, which may raise triggers to an anomalous behavior. This can be evidenced in the left image of Figure [2.14]

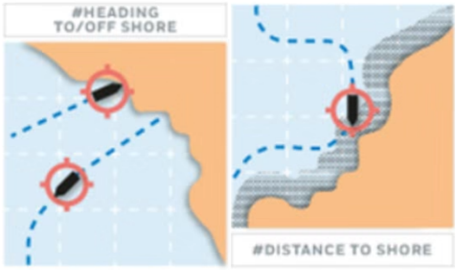


Figure 2.14: Shore Related Anomalies(image provided by CSW)

The behavior evidenced in the right image of Figure [2.14] shows a vessel that is reporting a position too close to shore and it is particularly dangerous to larger vessels as they can hit rocks and other materials in shallow waters.

It is worth noting that not all of the anomalous behavioral patterns are to be approached and detected by the solution that is being developed. The most interesting patterns that shall be analysed in further detail in the scope of this project shall be the position spoofing and route divertion, and the ones related to anomalous speed and heading changes

2.7 Time Series Data

Time series are sequences of data points recorded or observed for a time duration, usually at regular intervals. Each data point can have a specific timestamp, making it a chronological sequence. Some examples of time series data are stock prices, temperature measurements over the days, and entity geopositions. [Ferenti 2017] [Xu et al. 2020]

Key characteristics of time series data include:

- **Temporal Ordering**: Time series data is ordered chronologically, meaning each observation is associated with a timestamp, which is a crucial aspect of time series data, as it reflects the temporal evolution of the data.
- **Intervals**: Time series data has the possibility to have regular intervals between observations, depending on the nature of the data collection process. Irregular intervals stem from event-driven data collection processes, while regular intervals derive from data collection events that have well-defined intervals to be measured.
- **Trends**: Time series data often exhibit trends, indicating patterns in the data over time.

- **Seasonality**: Many time series exhibit seasonality, which refers to recurring patterns or fluctuations that follow a consistent time frame. For example, retail sales may exhibit seasonality with increased sales during holiday seasons.
- **Noise**: In time series data, there may be "noisy" data, which are irregularities, or outliers, that do not make a part of the underlying patterns, which can complicate the identification of patterns in the data.

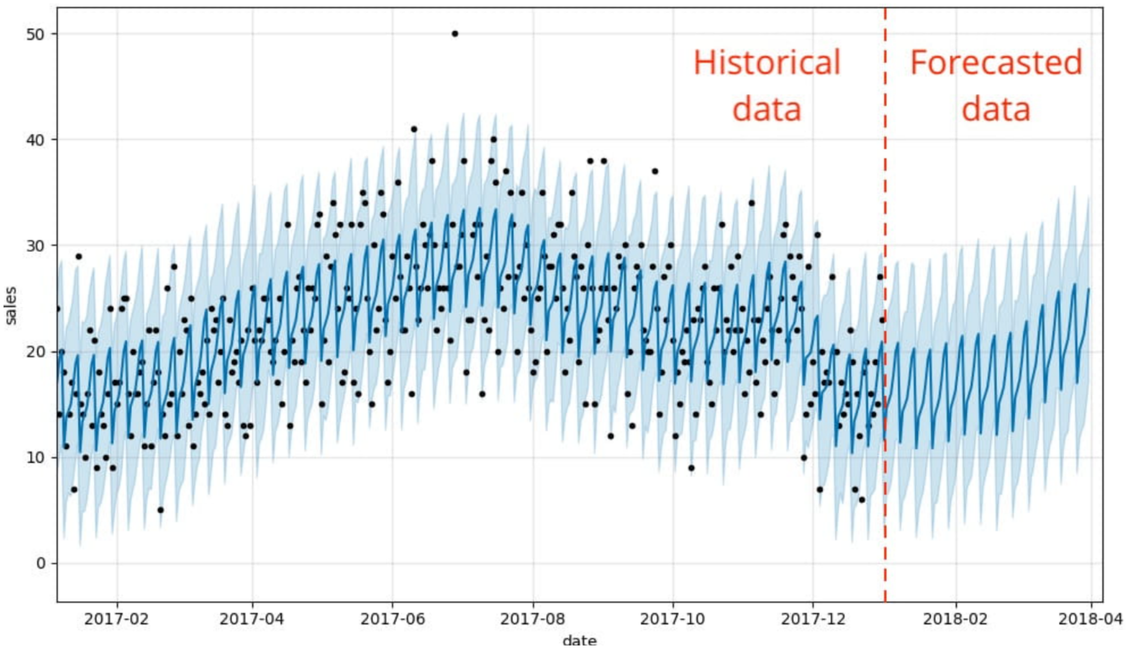


Figure 2.15: Time Series Data Visualization (Retrieved from [Kumar 2023])

Time series analysis requires the application of various techniques to understand and model the chronological patterns within the data, such as forecasting, which involves the prediction of future values based on historical observations, such as regression methods, exponential smoothing and machine learning algorithms. [Kumar 2023]

Time series data is used in several fields such as finance, defence, and ecological patterns monitoring. Developing solutions and methods to predict future data is crucial in these areas. In finance, it can lead to investment predictions that compromise the success of a business, while in defence it may influence geopolitical strategies decision making.

2.8 Machine Learning

Machine learning is a field of computer science that focuses on developing algorithms and models that can learn from and make predictions or decisions based on data. The goal of machine learning is to enable computers to automatically improve their performance on a specific task, without being explicitly programmed to do so. [Farahnakian, Heikkonen, and Nevalainen 2022]

Machine learning algorithms learn from data by detecting patterns and relationships in the data, which are used to make predictions or decisions on new data. There are various types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning, each of which has different methods for learning from data.[Dixit, S. Bhatia, and P. Bhatia 2022] [Samia, Soraya, and Malika 2022]

Supervised learning involves training a model on labeled data, where the desired output is already known, and the model learns to map inputs to outputs. Unsupervised learning involves training a model on unlabeled data, where the model learns to identify patterns or structure in the data without any predefined outputs. Reinforcement learning involves training a model to make decisions based on rewards and punishments received from its environment. [Dixit, S. Bhatia, and P. Bhatia 2022]

Machine learning has many applications, including image and speech recognition, natural language processing, recommendation systems, fraud detection, and autonomous vehicles, among many others. With the availability of large amounts of data and advancements in computing power and algorithms, machine learning is becoming increasingly important in many fields and industries. [Farahnakian, Heikkonen, and Nevalainen 2022]

One of ML's areas is Deep Learning (DL), which is becoming increasingly popular in many applications. [Samia, Soraya, and Malika 2022] The most popular algorithm involves Neural Networks (NN), which can be further subdivided in Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), each of which will be later on explained more in depth. [2.9]

"Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task." [Samia, Soraya, and Malika 2022] This approach is very popular as it saves significant computing power by reusing pre-trained models from similar projects.

ML can be divided in three main areas [Dixit, S. Bhatia, and P. Bhatia 2022] [Samia, Soraya, and Malika 2022] [Farahnakian, Heikkonen, and Nevalainen 2022]:

Supervised Learning: In supervised learning, the algorithm is trained on labeled data, which means that the input data has a corresponding output or label that the algorithm is trying to predict. The goal is to learn a mapping between the input and output data so that the model can accurately predict the output for new, unseen input data. The algorithm is given a set of input/output pairs and learns to generalize to new, unseen data. Examples of supervised learning include regression, classification, and object detection.

Unsupervised Learning: In unsupervised learning, the algorithm is trained on unlabeled data, which means that there is no corresponding output or label for the input data. The goal is to learn the underlying structure or patterns in the data. Clustering is one of the most common unsupervised learning tasks, where the algorithm groups similar data points together. Other examples of unsupervised learning include anomaly detection, dimensionality reduction, and generative modeling.

Reinforcement Learning: In reinforcement learning, the algorithm learns by interacting with an environment and receiving feedback in the form of rewards or penalties. The goal is to learn a policy that maximizes the expected cumulative reward over time. The algorithm learns through trial and error, by taking actions and observing the rewards received from the environment. Reinforcement learning is used in applications such as game playing, robotics, and control systems.

One key difference between supervised and unsupervised learning is the type of data used for training. Supervised learning requires labeled data, while unsupervised learning works with unlabeled data. Reinforcement learning, on the other hand, involves an agent interacting with an environment and receiving rewards or penalties.

Another difference is the type of task that each approach is best suited for. Supervised learning is well-suited for tasks where the goal is to predict a specific output for a given input. Unsupervised learning is better suited for tasks where the goal is to learn the structure or patterns in the data, without any predefined outputs. Reinforcement learning is best suited for problems where the agent must make a sequence of decisions to achieve a long-term goal, such as playing a game or controlling a robot.

In summary, supervised, unsupervised, and reinforcement learning are three different types of machine learning that differ in the type of data used for training, the goals of the learning process, and the types of problems they are best suited for.

2.8.1 Anomaly Detection Using Machine Learning Models

Anomaly detection is almost impossible to do by hand, due to the already discussed large amount of data [2.4], [2.5]. To analyse and detect these behaviors with a realistic amount of speed and accuracy, the best way is to use anomaly detection algorithms in the dominion of ML.

Some literature studies have been made, such as [Farahnakian, Heikkonen, and Nevalainen 2022], that analyse this possibility. Another example of a study done in the context of malicious network traffic is the one performed by [Rathod, Parekh, and Dholariya 2021], where ML models are based off to develop a solution that aims to detect these patterns in high traffic networks.

In summary, anomaly detection using ML models has been conceptualized and studied in other papers and literature surveys, but it is still very underdeveloped. It is relevant for this case study as it involves the underlying technology behind the development of the solution contained in this specific dissertation. Further analysis into the best models and algorithms for each specific scenario is observed in the following chapter [2.9].

2.9 Machine Learning Models and Algorithms

In this chapter, some ML algorithms will be explained in depth. Note that not all algorithms in the ML realm are approached and explained in this chapter as that would be worthy of a dissertation itself. The most relevant algorithms for this project are the ML algorithms able to handle sequential data in high amounts with a sufficient degree of efficiency and speed.

2.9.1 Neural Networks

Neural Networks (NN) are a class of machine learning models inspired by the structure and function of the human brain. They consist of interconnected nodes (neurons) that perform computations on input data, and learn from data by adjusting the strength of connections (weights) between neurons.

There are several types of Artificial Neural Networks [(Farahnakian, Heikkonen, and Nevalainen 2022)]:

- **Feedforward neural networks**: These are the simplest and most common type of neural network, in which information flows only in one direction, from input to output, without any loops or feedback. They are used for classification and regression problems, and can have one or more hidden layers.
- **Convolutional neural networks** (CNNs) are specialized neural networks made for processing images and videos with a spatially structured input. They employ pooling layers to minimize the spatial dimensions of the features and convolutional layers to extract features from localized areas of the input. For jobs like object detection, picture recognition, and video analysis, they are employed.
- **Recurrent neural networks** (RNNs) are neural systems that can analyze time series, text, and other sequential data types. They can model temporal relationships in the data by propagating information from earlier time steps to the current time step through recurrent connections. Inside RNNs, there are also Long Short-Term Memory (LSTM) algorithms, which are similar to RNNs, but include the fact that they keep a long-term memory unit, which allows the network to store information in the long term to allow for different outputs based on its memory and context of previous outputs.
- By learning the underlying distribution of the data, **Generative Adversarial Networks** (GANs) are neural networks that can create new data samples that mimic the training data. They are made up of a discriminator network that can tell the difference between authentic samples and fraudulent samples and a generator network that creates new samples and are employed for activities like data augmentation and the synthesis of images and videos. They are also employed for activities including speech recognition, video captioning, and natural language processing.

Below is an example of a simple NN [TIBCO 2016]:

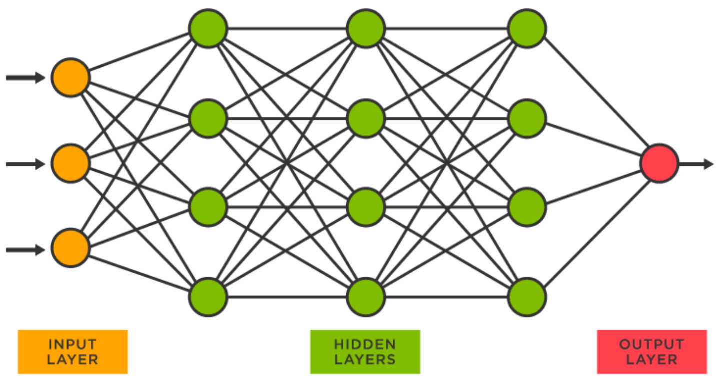


Figure 2.16: NN example according to [TIBCO 2016]

In the above example, There are three hidden layers, one input layer, and an output layer. The input data goes through the input layer, which is then passed on to the first hidden layer, where an activation function will be applied on the input data. Activation functions are mathematical functions that are applied to the output of each neuron in a neural network, in order to introduce non-linearity and enable the network to learn complex relationships in the data. Some examples of activation functions are the sigmoid function, the ReLU (Rectified Linear Unit) and the Tanh (Hyperbolic Tangent). After this, with each weight multiplied to the input data, the bias is applied at the end and the output data is obtained with the prediction data. Bias refers to a scalar value that is added to the weighted sum of inputs to each neuron. The bias value is typically learned along with the weights during the training process, and can have a significant impact on the performance of the neural network.

Each type of neural network has its own advantages and disadvantages. Some of the advantages of neural networks in general are:

- They can learn complex non-linear relationships in the data, and generalize to new data.
- They can handle high-dimensional and heterogeneous data, such as images, text, and audio.
- They can be trained in an end-to-end manner, without the need for manual feature engineering.

Some of the disadvantages of neural networks are:

- They can be computationally expensive and require large amounts of training data and computing resources.
- They can overfit the training data if the model is too complex or the training data is insufficient.
- They can be difficult to interpret and debug, due to their black-box nature and high dimensionality.

In summary, an NN is a type of ML model that is comprised by several distinct types of networks that are inspired by the function of the human brain. They consist of interconnected nodes (neurons) that perform computations on input data, and adjust their connections (weights), which are also based on human synapses, which are basically the connections between the neurons. NNs can learn complex non-linear relationships in the data, handle high-dimensional and heterogeneous data, and be trained in an end-to-end manner without the need for manual feature engineering. However, they can be computationally expensive, require large amounts of training data and computing resources, and can be difficult to interpret and debug due to their black-box nature and high dimensionality.

2.9.2 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a simple supervised ML algorithm with great performance in regression and classification tasks and is used by many real world applications and companies. KNN firsts measures the distance between the given test data and all training points, which then proceeds to elect the K number in which the point is closes to the test dataset, finally calculating the probability, in which the highest probability will be the chosen one. [Farahnakian, Heikkonen, and Nevalainen 2022]

K-Nearest Neighbors (KNN) is known to perform well on smaller and simpler data sets, but may not generalize as well to larger and more complex data sets and is also considered to be a relatively accurate model, but it may not perform as well on high-dimensional data sets. [Farahnakian, Heikkonen, and Nevalainen 2022]

In the case of a classification problem, the KNN algorithm finds the k closest neighbors to the new data point based on a distance metric, such as Euclidean distance or Manhattan distance. The most common class among these k neighbors is then assigned to the new data point as its predicted class label.[Farahnakian, Heikkonen, and Nevalainen 2022]

In the case of a regression problem, the KNN algorithm finds the k closest neighbors to the new data point and takes the average (or weighted average) of their output values as the predicted output value for the new data point. [Farahnakian, Heikkonen, and Nevalainen 2022]

KNN is a non-parametric algorithm, which means it does not make any assumptions about the distribution of the data. It is also a lazy learning algorithm, which means it does not require any training or model building, and simply stores the training data in memory for future predictions. [Farahnakian, Heikkonen, and Nevalainen 2022]

Since KNN is a non-parametric method, it makes no assumptions about how the data will be distributed. It is also a lazy learning algorithm, which merely keeps the training data in memory for future predictions without than requiring any training or model development. [Farahnakian, Heikkonen, and Nevalainen 2022]

KNN is simple to implement and can handle non-linear data and multi-class classification problems. However, its main limitation is that it can become computationally expensive and memory-intensive for large datasets and high-dimensional feature spaces, as it requires calculating distances between all pairs of data points. [Farahnakian, Heikkonen, and Nevalainen 2022]

2.9.3 Hidden Markov Model

Hidden Markov Model (HMM) is an ML algorithm that is used mainly in temporal datasets and is a popular unsupervised classification solution. HMMs have been used in many solutions such as in the areas of handwriting and speech patterns recognition. [Farahnakian, Heikkonen, and Nevalainen 2022]

HMM consists of two types of parameters: transition probabilities between hidden states, and emission probabilities for observed data given the hidden state. The model can be trained using the Baum-Welch algorithm, which is a variant of the Expectation-Maximization algorithm. [Toloue and Jahan 2018]

According to [Farahnakian, Heikkonen, and Nevalainen 2022], a semi-automated ship behavior detection using HMMs has been proposed, which uses a filtering technique called Density-Based Spatial Clustering of application with Noise (DBSCAN) to cluseter the well-established path and extract spatial regions. After this, "they reduced the resulting data by deleting outliers to easily pick features before clustering remaining patterns. Finally, Partitioning Around Medoids (PAM) algorithm, which is designed to explore and find a sequence of objects named (Medoids) is applied to the data to obtain a final clustering" [Farahnakian, Heikkonen, and Nevalainen 2022] "The main disadvantage of using algorithm-based medoids is that determining the number of cluster is too difficult." [Farahnakian, Heikkonen, and Nevalainen 2022] The example's flowchart is as follows:

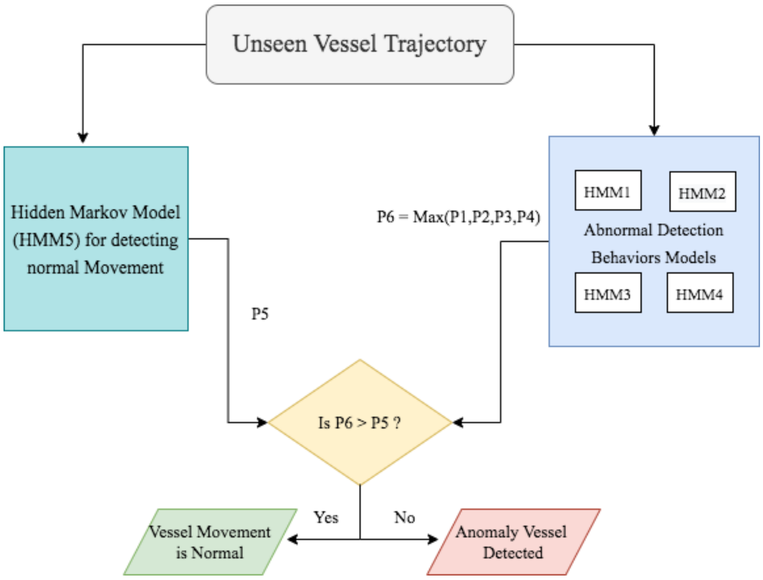


Figure 2.17: HMM example according to [Farahnakian, Heikkonen, and Nevalainen 2022]

To conclude, HMM has several advantages, such as the abbility to model complex temporal dependencies in sequential data, being able to handle noisy or missing data by incorporating prior knowledge and smoothing techniques and the ability to perform both classification and regression tasks, depending on the type of output.

However, HMM also has some limitations, such as assuming the observed data are conditionally independent given the hidden state, which may not be true in some cases. HMM can also be computationally expensive and memory-intensive for large datasets and longe sequences, especially for high-dimensional data. HMM is also sensitive to the initial parameters settings and can be prune to overfitting.

Overall, HMM is a powerful and flexible model for sequential data analysis, with various applications in speech recognition, natural language processing, and bioinformatics. However, it requires careful parameter tuning and regularization to avoid overfitting and ensure robustness.

2.9.4 Support Vector Machine

Support Vector Machine (SVM) is a supervised ML algorithm that analyses data and identifies patterns within datasets. SVM is usually used for regression, classification, and outliers detection. The SVM algortim's goal is to explore and find a hyperplane in two-dimensional and three-dimensional spaces that are capable of classifying data points. [Farahnakian, Heikkonen, and Nevalainen 2022] The purpose of SVM is to find the best possible decision boundary (or hyperplane) that can separate the data into different classes in the best way possible. It is known for its ability to handle high-dimensional data and its robustness to overfitting, which is when the model fits the training data too closely and performs poorly on new, unseen data. In the case of a binary classification problem, SVM tries to find the hyperplane that separates the data into two classes with the maximum margin, which is the distance between the hyperplane and the closest data points from each class. This margin is considered the "safest" distance from the hyperplane to the data points, and helps to reduce the risk of overfitting. [Farahnakian, Heikkonen, and Nevalainen 2022]

SVM can handle linear and nonlinear data by using different types of kernels, such as linear, polynomial, and radial basis function (RBF) kernels. The kernel function maps the input data to a higher-dimensional feature space, where the data points are more separable. SVM then finds the optimal hyperplane in this higher-dimensional space that separates the data with the maximum margin. [Farahnakian, Heikkonen, and Nevalainen 2022]

This technique has been used in the detection of anomalous behaviors and has shown experimental results of 99,81%. [Farahnakian, Heikkonen, and Nevalainen 2022]

The first step to implement this algorithm, is to divide the data in two sets, normal, and abnormal behaviors. After this, two more sets are created, random normal and random abnormal. "Then, the group of normal track data has been mixed with random anomaly dataset for training the model, and the combination of the group of anomaly track data with random normal dataset has been used for testing the SVMs classification model." [Farahnakian, Heikkonen, and Nevalainen 2022] After this, the system clusters the historical trajectories into labeled groups and creates the SVM based classifier, which then identifies the new trajectory prediction.

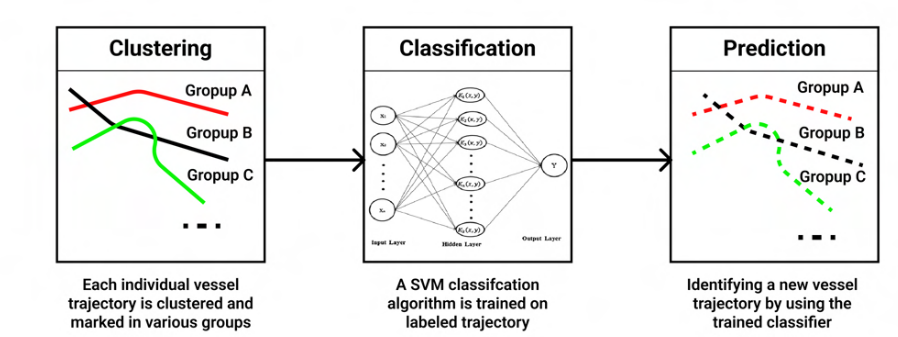


Figure 2.18: SVM example according to [Farahnakian, Heikkonen, and Nevalainen 2022]

Overall, SVM is a powerful machine learning algorithm that can be used for both classification and regression analysis, and is especially useful in cases where the data is high-dimensional and the goal is to find a clear separation between classes

Chapter 3

Value Analysis

This chapter focuses on the understanding and development of the concept of analysing the value of a solution and/or product, defined as "Value Analysis" in the scope of this project. This concept of value analysis is not new, it was first defined at General Electric Co. during World War II. During this time, there were shortages of both staff and products, and, when replacing these, Lawrence Miles et al. found out that, when substituting these resources, the new ones often brought more value and/or reduced costs when compared to the previous resources, giving birth to the concepts of "value analysis" and "value control". [Sharma et al. 2012]

"The primary objective of value analysis is to assess how to increase the value of an item or service at the lowest cost without sacrificing quality." [Nicola 2020a] Value Analysis is a systematic process of analysis and evaluation that requires planning, control, and coordination among the relevant parties. When analyzing the value of a product, the first concept that comes to mind is that said product must meet the requirements specified by the customer. To further assess the quality of this product, a way to quantify the degree of fitness between the requirements and the implementation must be established. To succeed, this degree of fitness must be as high as possible. To provide a benefit to the company, the analysis must trigger (in case it is necessary, which is most often) a process where reworks and redesigns are executed, to improve the correlation between the requirements and the implementation. [Nicola 2020a]

The improvement of a product must take into account the environment it is being performed as well as the different mindsets present in these domains. Internally, a business must perform these analyses and continuous improvements corroborated by feedback presented by relevant stakeholders to allow for the best practices that will improve both the pipelines, attitude, and value of the business in itself, improving safety and compliance requirements, product quality and better shaping and understanding customer and stakeholder requirements and desires, both specified, and hidden, by also allowing the understanding of the customers' values, culture and mentality, allowing for a better prediction of what the client wants when they specify a certain request in the product. As for the external factors, namely, the market for the product/service, value analysis helps identify the pricing practices, reduce the complexity of the pipelines and help structure the products and marketing around ever-maturing markets, which will also help to understand and, in some way, shape the marketing strategy around the evolution of the value analysis of the product/service. [Nicola 2020a]

Value analysis is especially important in the scope of this project, as it is centered around defense, one of the main application areas of value analysis, which is safety-critical, meaning it involves lives at risk, such as in search and rescue. Being a critical safety project means there are lives at risk in the direct operation of the software, which means there are lives that can be directly saved by the correct use of this software, which further boosts the necessity of quality control of this project, which is a concept intimately related to value analysis.

Developing a pipeline for analysing value is crucial in the scope of value analysis, to better automate and produce artifacts that will help with the former. The first step is to gather the requirements of the product or service, harnessing a first iteration of what needs to be developed for the fulfillment of the requirements and identifying the customer needs. Developing a first MVP (Minimum Viable Product) and presenting it to the customer gives valuable feedback that will be used in the next step which is to select alternatives to the development of the next iterations, shaping the course of development and, potentially, reshaping the customers needs, by also allowing them to draw conclusions and mature on their idea of a quality product, sometimes shaping their idea and perception of what is value, which will be defined later in this chapter. Implementing the new, and possibly redefined, requirements allow for the business to iterate through versions of the initial product/service idea permitting a maturity model that further improves quality on what was the first idea of a quality product. To better help analyse these steps, some artifacts will be produced. These artifacts include [Nicola 2020b]:

1. **Quality Function Deployment (QFD)**: an artefact that is used to ensure the customers' needs are prioritized. Involves the House of Quality (HOQ) which helps correlate customer requirements with technical requirements;
2. **Brainstorming and mind map**: brainstorming allows for the visualization of the ideas that will favour the course of development of the project, while a mind map helps organized these ideas into a graph;
3. **SWOT Analysis** identifies the strengths, weaknesses, opportunities, and threats of the product/service, which help shape the business strategy plan;
4. **Business Model of CANVAS**: the Business Model Canvas is used to map out the key elements of a business model and how they relate to each other;
5. **Value Proposition CANVAS**: the value proposition CANVAS is used to design and develop new products, services, and business models that create value for customers;
6. **TOPSIS/AHP**: these are both multi-criteria decision-making methods, but they have some differences in their approach and focus. AHP is a hierarchical decision-making method that breaks down the decision problem into smaller, more manageable parts by creating a hierarchy of objectives. TOPSIS on the other hand is a non-hierarchical decision-making method that uses a geometric approach to evaluate alternatives based on a set of criteria.

Value is the perceived measurable quality of a certain product/service. Value is crucial in establishing a successful relationship between the business and the customer, as only by providing value, or making the customer perceive that they are getting value out of a product/service will they be happy and create a positive connection with the product/service, and, consequently, with the business associated with said product/service. Depending on the perspective, there will be different values. If taken from the business side, lowering risk, saving time, or being environmentally friendly may be some of the values that are more perceived as having a higher value, while on the other hand, if taken from the customer perspective, they may value more such things as usability, aesthetics or newness of the product/service. [Nicola 2020a]

Value proposition “is an overall view of a company’s bundle of products and services that are of value to the customer.” [Nicola 2020a] Value Proposition asks such questions as what the target audience is, or what is the uniqueness of the product. This helps justify the reason why the customer should purchase or subscribe to our product/service. In the following sub chapters, the concepts around value analysis will be further explained in more detail and this specific thesis’ value analysis will be performed, based on the product that is being developed, and, keeping the customers in mind as well as the business involved.

3.1 New Concept Development Model

When developing a new product or service, it is important to start on a very high level concept and work from a top down perspective, which means starting by establishing a high level vision of what is to be achieved, and specify what follows in a progressively narrow fashion. According to [Koen et al. 2001], the innovative process "can be divided into three processes: the fuzzy front end (FFE), the new product development (NPD) and the commercialization".

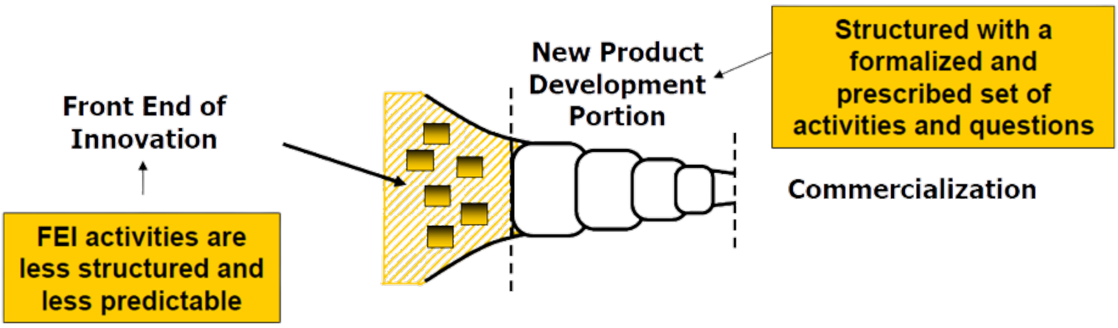


Figure 3.1: Front End of Innovation according to [Koen et al. 2001]

The Fuzzy Frontend is fuzzy because the products in this stage experience a high level of uncertainty and inconsistency as well as ambiguity and fast-paced redesigns. As the term "fuzzy" usually comes with a negative connotation, FFE is replaced with Front End of Innovation (FEI) by some authors. The FEI is where the designers conceive the higher-level concepts and models of what the future product will look and feel when finalized.

This step precedes the NPD, which is the step where developers create the product based on the decisions in the FEI. [Koen 2004]

The final step in the innovation process is commercialization. This step brings the product to the market, which allows customers to give feedback on the product so it can be redesigned based on this feedback, which makes the development go back to the drawing board, in this case, to the FEI, which allows for iterative development of the product.

The fact that the FEI is lacking of standard terminology and a set work pipeline, means it is unreliable and inconsistent. To resolve this issue, [Koen et al. 2001] proposed the New Concept Development Model (NCDM). The NCDM is a product/service development strategy that "provides a common language and definition of the key components of the Front End of Innovation". [Koen et al. 2001]

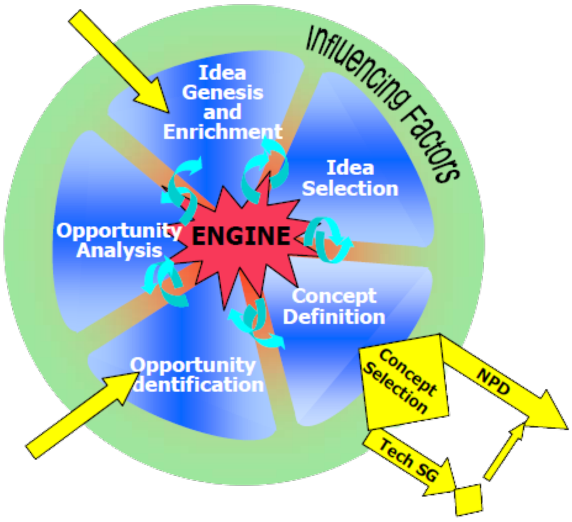


Figure 3.2: The New Concept Development Model (NCDM) according to [Koen 2004]

The NCDM consists of [Koen et al. 2001]:

1. **The interior**: specifies the five key elements in the FEI;
2. **The engine**: is fueled by the stakeholders, leadership and culture of the organization and is what manages and processes the five key elements;
3. **The influencing factors**: involves the environment around the company developing the product, such as the outside world, the science and the organizational capabilities.

In the above image, there is a connection between every element, indicated by the arrows flowing back and forth. "The circular shape is meant to suggest that ideas are expected to flow, circulate and iterate between and among all the five elements, in any order or combination, and may use one or more elements more than once." This back-and-forth iteration defends that the time of product development is shortened positively, even though it may delay the FEI development and cause it to iterate through several profound changes over the course of the lifecycle. [Koen et al. 2001]

3.1.1 Opportunity Identification

As reported by [Koen et al. 2001], identifying opportunities is where organizations identify situations that may be of interest to pursue. These allow for the management of the allocation of resources to new business areas that allow for innovation and expansion. These opportunities usually follow the company's culture, values, and vision. The opportunity can reshape the business model of the company entirely, or improve it upon itself. These opportunities either trigger the creation of a new product/service or the improvement of an existing one. Opportunities do not last forever, as they are usually taken advantage of by competition, making the organization lose its competitive edge. Furthermore, opportunities can be short-term countermeasures to a competitive threat, the possibility to achieve a competitive advantage, the analysis of market and customer trends, and a means to reduce the cost or improve the operations in a specific organizational area.

Sub chapter [1.3] shows there is a problem to be solved. Wherever there are problems to be solved, there are opportunities to solve them. Furthermore, sub chapter [2.5] analyses the already extant OVERSEE project. This project requires a way to trigger warnings about the detection of anomalous behaviors patterns in maritime traffic (see sub chapter [2.6]). This opportunity presents an array of possible solutions, which will be analysed in the following sub chapter [3.1.3] and selected in sub chapter [3.1.4]. Essentially, this is where the opportunity developed in the scope of this project is located. It tries to solve a problem that an already existing project has, by linking a new service to an already existing and proven one. This proves to be much more interesting for the client, as it only builds on top of an already existing concept, which will not require any significant changes. The fact that a state-of-the-art technology in the field of artificial intelligence is being used, means that there is also the possibility to provide significant feedback for the development of machine learning, by providing important data in the forms of trained data sets as well as analysis results and conclusions.

3.1.2 Opportunity Analysis

After identifying the opportunities, and before translating them into specific business and technology opportunities, these need to be analysed. "Extensive effort may be committed for focus groups, market studies, and/or scientific experiments. However, the amount of effort expended is dependent upon the attractiveness of the opportunity, the size of the future development effort, the fit with the business strategy and culture, and the risk tolerance of the decision-makers." [Koen et al. 2001] What this means is, after identifying the opportunities, they are assessed with the mission to evaluate their viability and feasibility. This involves steps such as identifying and analysing the environment around the opportunity, such as the political, social, environmental, and technological situation [Cox 2021] of the universe that revolves around the opportunity at the time of the assessment.

To analyse the opportunity, a SWOT analysis was generated. A SWOT analysis revolves around identifying Strengths, Weaknesses, Opportunities, and Threats related to the opportunity at hand to assess whether the business is viable and what are the constraints and difficulties as well as the opportunities and strengths that will be faced. [Hajizadeh 2019]

Applying the knowledge above to create the SWOT analysis, the first thing to be analysed is the environment in which the project stands. This system is implemented in maritime vessels, which means it is possible to integrate it with the existing OVERSEE project, which involves strong partners such as Critical Software and Marinha Portuguesa. The dimension of maritime traffic, further and more deeply discussed in chapters [1] and [2], allows for the availability of a very large dataset. The customer segment analysis provides the potential future clients that envision the presentation of the solution in international sales conventions, boosting the potentiality of the project further. The threats present in this environment are hackers, pirates, and other criminal activity inside the vessels that jam the AIS signals. They turn them off or otherwise damage them to obfuscate the signals, which implies turning them off and justifying them with bad weather conditions or spoofing the signals, faking the vessel's actual position. Zooming out to a broader scope, the increase in AI technology demand is an opportunity to both take advantage of this fact and contribute to scientific research.

Competition anomaly detection in maritime traffic isn't a preoccupation in the scope of this project. The complexity of this technology significantly reduces the competition, making this both a weakness and a strength. It's a weakness because it can become overwhelming to develop the solution, and it's a strength because of the existence of the OVERSEE technology.

Based on this analysis the following SWOT was generated:

Figure 3.3: Opportunity SWOT Analysis

3.1.3 Idea Genesis and Enrichment

"Genesis is the birth, development and maturation of the opportunity into a concrete idea." [Koen et al. 2001] Idea Genesis is an iterative process that involves going back and forth on the drawing board and thinking of steps to materialize the idea. Idea Genesis goes through many iterations where these steps are examined, studied, and discussed to define which are more worthwhile to develop. The best way to produce ideas is through brainstorming sessions and creating idea banks. Ideas also come in daily events during what is the course of a regular day. Ideas may come from typical general day occurrences, but they also often come from unusual requests and atypical experiments. Idea Genesis can trigger other steps, such as Idea Selection and Opportunity Analysis, which reinforces what was shown in Figure [3.2], further demonstrating that the NCDM may not always be linear, with steps going back and forth.

By using the brainstorming technique and upon further research on the subject [Farahnakian, Heikkonen, and Nevalainen 2022], the best approach is to develop a machine learning algorithm. The algorithm needs to comply with some prerequisites. These prerequisites include handling large datasets and sequential time-series data, generalization capability, accuracy, and complexity. The thought ideas are:

- **K-Nearest Neighbors**: simple and efficient algorithm good at generalizing small datasets;
- **Neural Networks**: complex algorithm able to handle large datasets and sequential data;

- **Support Vector Machine**: complex and accurate algorithm able to handle sequential data;
- **Hidden Markov Model**: this algorithm is usually used in time-series datasets.

3.1.4 Idea Selection

To proceed with the solution design and development, the most ideal solution will be selected from sub chapter [3.1.3]. Idea selection is a process in businesses that allows the selection of the most valuable processes and product ideas from a list of options. [Koen et al. 2001] To aid in the selection of the ideal solution, there are formal processes such as Analytical Hierarchy Process (AHP) and Technique for Order Performance by Similarity to Ideal Solution (TOPSIS). AHP is one of the main methods developed in discrete multi-criterion decision-making. AHP subdivides the main problem into lesser hierarchical sub-problems. TOPSIS helps decide the best solution by taking as inputs different alternatives into account and attributing a score to each on the distinct criterion. TOPSIS hypothesizes two artificial alternatives, the ideal alternative, and the negative ideal alternative. TOPSIS is more suitable for situations where the criteria are quantitative and the alternatives can be evaluated on a common scale. This is the reason why TOPSIS was chosen as the idea selection process for this project. [Nicola 2018] [Duffuaa 2009]

The criterion chosen as the driving factors for the selection were efficiency (quantitative benefit), complexity (cost attribute), generalization (qualitative benefit), adequation (qualitative benefit), and accuracy (quantitative benefit). Efficiency is the time constraint the algorithm takes to execute and draw conclusions. It is chief to make it a viable solution, otherwise, being too inefficient would not provide enough time to produce a feasible solution in time, as the times to train the model influence the quality of the product significantly. Complexity defines the degree of difficulty and intricacy in designing and developing the solution. If the solution is too complex, implementing it may not be realistic in the relevant time span. Generalization is the ability of a model to perform well on new, unseen data. This aspect is crucial as the objective of the solution is to make accurate predictions based on constant new real world data. Adequation refers to the algorithm's relevance in the context of the analysis of the type of data that is presented in the scope of the inputs of the project. As was discussed in chapter [2.4], AIS messages, which are the inputs to this solution, are sequential. This requires an algorithm that is able to take into account the fact that there must be a memory created for the analysis of the sequential data, not only the latest inputs. Accuracy refers to the quality of the results obtained from the model's predictions. It is a relevant criteria as the algorithm must provide with accurate results. A high level of accuracy means that the model is able to make correct predictions for a high percentage of the cases. Not providing accurate results means the solution would be untrustworthy, hence impracticable.

Following TOPSIS, and with the 4 alternatives and 5 criterion established, $m = 4$ and $n = 5$. Let x_{ij} be the score of option i with respect to criterion j and $X = x_{ij}$, a 4*4 matrix. [Samia, Soraya, and Malika 2022]

Table 3.1: Matrix X with alternative-criteria relationships

	Efficiency	Complexity	Generalization	Adequation	Accuracy
KNN	9	5	7	6	8
NN	6	8	9	10	10
SVM	8	7	7	7	9
HMM	7	8	6	9	8

The values provided on Table [3.1] follow a rationale made in conformance with reading [Farahnakian, Heikkonen, and Nevalainen 2022], where it was evidenced that the SVM model, although very efficient and quite accurate, with 99,81% accuracy in the case study, it did not comply with the fact that, due to the large amount of data received from each vessel following each position's report, there will be a large amount of data, which this model has evidenced to have trouble handling, hence the lower adequacy score. The KNN method evidenced a lower accuracy of 0.93%, while also proving to be the lest complex of the discussed implementations. However, it lacks adequacy due to its lower generalization as compared to the other implementations. The HMM model presents an accuracy of 96%, which is acceptable, but lower than SVM or NN, which will be discussed afterward. The low generalization score stems from the fact that this implementation requires all anomalous behaviors to be well defined in advance, which, as will be discussed in later chapters, isn't the case with the available dataset. Finally, a Neural Network approach was given these values due to the fact that, while being a quite complex and slower implementation compared to the other implementations, it demonstrates a good generalization capability and an accuracy of over 99%.

The next step is to normalize the decision matrix to convert the values into non-dimensional values, so that a comparison is possible. The normalization follows the formula:

$$r_{ij} = \frac{x_{ij}}{(\sum_i x_{ij}^2)^{\frac{1}{2}}}, \text{ for } i = 1, \ldots, m; \ j = 1, \ldots, n \tag{3.1}$$

This step was divided in two. First:

Table 3.2: Matrix with $(\sum_i x_{ij}^2)^{\frac{1}{2}}$ result

	Efficiency	Complexity	Generalization	Adequation	Accuracy
KNN	81	25	49	36	64
NN	36	64	81	100	100
SVM	64	49	49	49	81
HMM	49	64	36	81	64
$\sum_i x_{ij}^2$	230	202	215	266	309
$(\sum_i x_{ij}^2)^{\frac{1}{2}}$	15,17	14,21	14,66	16,31	17,58

After obtaining the results in table [3.2], for each value x_{ij} in table [3.1], equation [3.1] is applied to obtain r_{ij}, which represents the normalized scores:

Table 3.3: Matrix with r_{ij} values for each x_{ij} value

	Efficiency	Complexity	Generalization	Adequation	Accuracy
KNN	0,59	0,35	0,48	0,37	0,46
NN	0,40	0,56	0,61	0,61	0,57
SVM	0,53	0,49	0,48	0,43	0,51
HMM	0,46	0,56	0,41	0,55	0,46

Because not all criterion account for the same level of importance when taking the decision, different weights were applied to each criteria. The most important criteria is the adequation. If the algorithm is not adequate to solve the problem, it shall not even be considered for the solution development, hence, it has a weight of 0,3. Generalization, efficiency and accuracy have a weight of 0,2, as they are also very important to obtain good results in a realistic time frame with a high level of trustworthiness. Complexity has a weight of 0,2, as it does not directly affect the performance of the algorithm, but rather the implementation difficulty. With these weights in mind, we shall calculate the weighted normalized decision matrix, v_{ij} by multiplying the weights with each matrix value, with the formula:

$$v_{ij} = r_{ij} w_{ij} \tag{3.2}$$

Table 3.4: Matrix with v_{ij} values for each x_{ij} value

	Efficiency	Complexity	Generalization	Adequation	Accuracy
KNN	0,118	0,035	0,096	0,111	0,092
NN	0,080	0,056	0,122	0,183	0,114
SVM	0,106	0,049	0,096	0,129	0,102
HMM	0,092	0,056	0,082	0,165	0,092

The next step is to select the ideal solution for each criterion in table [3.4], i.e. the highest value in the efficiency, generalization, adequation and accuracy columns, and the lowest value in the complexity column. The ideal solution is represented by array A^*. Let J be the set of benefit attributes where more is better and J' the set of negative attributes where less is better:

$$A^* = \{v_1^*, ..., v_n^*\}, \text{ where } v_j^* = \{max_i(v_{ij}) \text{ if } j \in J; min_i(v_{ij}) \text{ if } j \in J'\} \tag{3.3}$$

To select the negative ideal solution, i.e., the alternative with the worst attributes, according each criterion, the highest value is selected where the worst attribute in the set of benefit attributes the highest value and the lowest value is selected where the worst attribute is the lowest value. Let $A^{'}$ represent the array of negative ideal solutions:

$$A^{'} = \{v_1^{'}, ..., v_n^{'}\}, \text{ where } v_j^{'} = \{min_i(v_{ij}) \text{ if } j \in J; max_i(v_{ij}) \text{ if } j \in J'\} \tag{3.4}$$

With this in mind, the ideal values are highlighted in green in table [3.4], while the negative ideal values are highlighted in red. With $A^* = \{0,118; 0,035; 0,122; 0,183; 0,114\}$ and

$A^{'} = \{0,080; 0,056; 0,082; 0,111; 0,092\}$, we are able to determine the separation from the ideal solution. Let S_i^* be the separation from the ideal solution for each alternative:

$$S_i^* = [\sum(v_{ij}^* - v_{ij})^2]^{\frac{1}{2}} \tag{3.5}$$

Table 3.5: Separation from ideal solution (A^*), S_i^*

	Efficiency	Complexity	Generalization	Adequation	Accuracy	$\sum(v_{ij}^* - v_{ij})^2$	$[\sum(v_{ij}^* - v_{ij})^2]^{\frac{1}{2}}$
KNN	0	0	0,000676	0,005184	0,000484	0,006344	0,080
NN	0,001444	0,000441	0	0	0	0,001885	0,043
SVM	0,000144	0,000196	0,000676	0,002916	0,000144	0,064000	0,064
HMM	0,000676	0,000441	0,001600	0,000324	0,005184	0,008225	0,091

With this, $S_i^* = \{0,080; 0,043; 0,064; 0,091\}$.

The separation from the negative ideal solution, $S_i^{'}$, is obtained with:

$$S_i^{'} = [\sum(v_{ij}^{'} - v_{ij})^2]^{\frac{1}{2}} \tag{3.6}$$

Table 3.6: Separation from negative ideal solution ($A^{'}$), $S_i^{'}$

	Efficiency	Complexity	Generalization	Adequation	Accuracy	$sum(v_{ij}^{'} - v_{ij})^2$	$[\sum(v_{ij}^{'} - v_{ij})^2]^{\frac{1}{2}}$
KNN	0,001444	0,000441	0,000196	0	0	0,002081	0,046
NN	0	0	0,0016	0,005184	0,000484	0,007268	0,085
SVM	0,000676	0,000049	0,000196	0,000324	0,0001	0,001345	0,037
HMM	0,000144	0	0	0,002916	0	0,00306	0,055

With this, $S_i^{'} = \{0,046; 0,085; 0,037; 0,055\}$. Having both the separation from the ideal solution and the negative ideal solution, we can calculate the relative closeness to the ideal solution, C_i^*:

$$C_i^* = \frac{S_i^{'}}{S_i^* + S_i^{'}} \tag{3.7}$$

Table 3.7: Relative closeness to ideal solution, C_i^*

	$\frac{S_i^{'}}{S_i^* + S_i^{'}}$	C_i^*
KNN	0,040 / (0,080 + 0,040)	0,333
NN	0,085 / (0,043 + 0,085)	0,664
SVM	0,037 / (0,064 + 0,037)	0,366
HMM	0,055 / (0,091 + 0,055)	0,377

In summary, the TOPSIS analysis assists in taking the right decision among a set of solutions, based on a set of criterion. In this analysis, the alternative solutions were KNN, NN, SVM and HMM, and the criterion were picked based on the expected outcomes of this project, which consist of efficiency, complexity, generalization capability, adequation, and accuracy. When analysing table [3.7], the closest solution (highest value of C_i^*, highlighted in gray) is Neural Networks. Neural Networks provide a high degree of generalization, accuracy and are adequate for analysing sequential, time-series datasets, making this solution a good fit.

This answer correlates to what was analysed with the paper by [Farahnakian, Heikkonen, and Nevalainen 2022], which comes to a similar conclusion. This type of solution is more thoroughly scrutinized in chapter [2].

3.1.5 Concept Definition

Concept Definition is the final element of the NCDM. This step involves the development of a business model and study the use cases of the opportunity. This step precedes the New Product and Process Development (NPPD). This step is similar to what is done on the initial stages of the NPPD, being skipped entirely in some occasions. As the effort escalates, usually the risk tends to decrease, as the processes and technologies are more well defined. [Koen et al. 2001] In the scope of this project, the business plan and technology development analysis will be performed on chapter [3.2], where the Value Proposition CANVAS (Figure [3.5]) and the Business Model CANVAS (Figure [3.6]) will be inspected and assessed in further detail.

3.2 Value

Value can be subdivided into different perspectives of what the term value means, which was also discussed in chapter [3], the beginning of this sub chapter's chapter. "The term 'customer value' is used within the marketing literature to portray both what is derived by the customer from the supplier, and also what is derived by the supplier from the customer." [Woodall 2003] This type of value will be assessed in sub chapter [3.2.3]. Different customers will have different value perspectives. Perceived value illustrates the value given to the same product/service by different customers. This means different customers perceive distinct values for the same products. [Nicola 2020a] Another type of value is the scientific value that is retained and that can be used in the future by the scientific area that explores the subjects discussed and developed in the context of this project. Having this in mind, the following sub chapters are critical in understanding what will be valued by the specific client of this project as well as the scientific value this may supply to the scientific community.

3.2.1 Value Chain

The value chain is a business management concept that describes the activities involved in producing and delivering a product or service, from the acquisition of raw materials to the final delivery of the product or service to the customer. [Porter 1985]

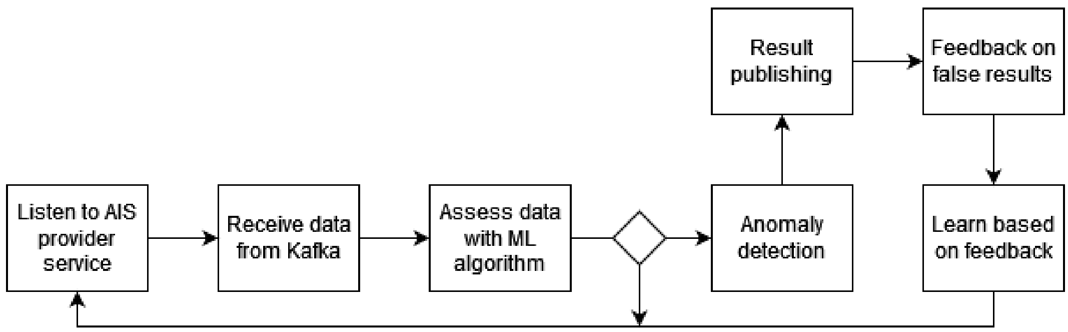


Figure 3.4: Value Chain

Each of figure [3.4]'s steps provides a different perceived value to the company, as well as the scientific community. This model helps increase the productivity and efficiency of each step individually involved in the operations incriminated in the obtainment of the end result. The most valuable steps for the company are the first, fourth, and last steps. The first step is important to integrate the new microservice with the already existing tool, while the fourth step comprises the results and conclusions drawn from the results gauging. The reason the first step is so important results from the fact that in the manner that this project is being developed, the already existing tool will provide even greater value to the client. The last step is also of great importance as it allows for the reinforcement learning paradigm application to the model so that it can improve its results based on the user's feedback. This is of great value to the customer as it allows them to obtain more robust results, which are invaluable in better detecting anomalies and assisting these occurrences.

For the scientific community, the most valuable steps are the data assessment (step 3), and the last step. These are more important to the scientific community since they provide more data results and conclusions from the already existing algorithms, hence allowing for their potential improvement.

In sum, the client is more focused on the integration with the already existing tools, and on obtaining results to meet their needs, while the scientific community is more captivated by the process and the development of the algorithms and operations themselves so that the already existing tools and processes can be improved.

3.2.2 Value Proposition

"A value proposition describes how a company's offer differs from those of its competitors and explains why customers buy from the company." [Lindič and Marques da Silva 2011] Customers look for benefits in products. The Value Proposition helps differentiate the product from this company from the rest because the reality is that there are usually other alternatives. So the value proposition consists of what we offer to the customers, what type of value or benefit is associated with the offer, and to whom the offer is being made. The value proposition forms the core of the business model [District 2012] "The value proposition is the central element of the value model (...) it describes the logic of what products and services are offered to the customer." [Petrovic and Kittl 2003] The process of creating the value proposition consists of three steps. This three-step process involves problem discovery, identifying which features the solution must have and designing the business model for the proposed solution. The first step was overseen in sub chapter [1.3], where the problem is discussed. The second and third steps involve identifying what the requirements for the solution to be viable must exist, consequently helping to create the business model, which, in the scope of this project, will be the Value Proposition CANVAS (Figure [3.5]).

The Value Proposition CANVAS (VP CANVAS) is subdivided into two sections. The organization's value proposition for the client (value map), and the client profile. The value map consists of the product and its benefits on the organization's side, which they will create. The customer profile indicates the current pains of the client that the product will fix, as well as the gains they will get from the product. Each of these two sections is further subdivided into three different sections each [International 2023]

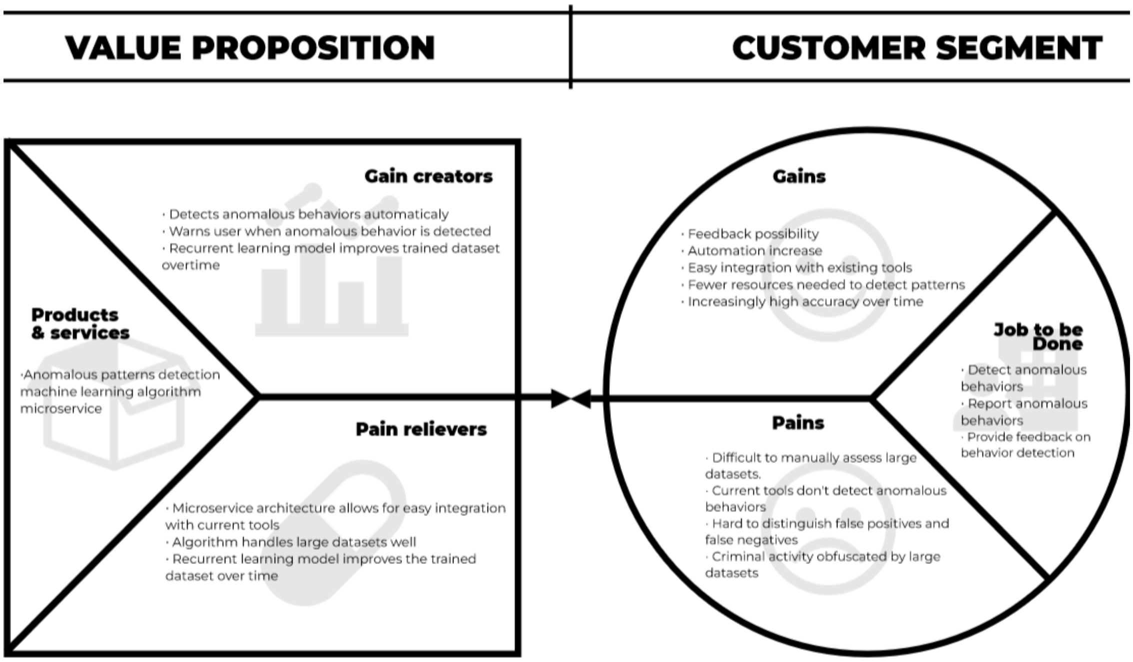


Figure 3.5: Value Proposition CANVAS

- **Value Proposition**:

  Pain Relievers how the product will solve the customer's pains (problems);

  Gain Creators is "how the product or service creates customer gains and how it offers added value to the customer" [International 2023];

  Products & Services that will create the previously mentioned pain relievers and gain creators.

- **Customer Segment**:

  Gains are the benefits which the customer needs and that is nice for them to see in the product/service, sometimes even as extras;

  Pains are the difficulties the customer feels and goes through in the process of achieving the result aimed by the product/service;

  Job to be Done is the objectives the customers are trying to meet with their current operations, and that the product/service presented in the Value Proposition is going to solve.

3.2.3 Value for the Client

It is relevant for this project to study this value concept more in-depth as it is directed towards specific companies, in this case, Critical Software and Marinha Portuguesa, which represent the customer. Value for the customer (VC) represents what the customer value in a product or service. To better understand this, a business model CANVAS was developed (3.6). The business model CANVAS visually represents the overview of the company's operations, such as the revenue streams, key activities, and customer segments, among others. [Hersztowski 2020] This helps understand what the client values in terms of the specific product, and helps relate one another, i.e. the product and the customer values.

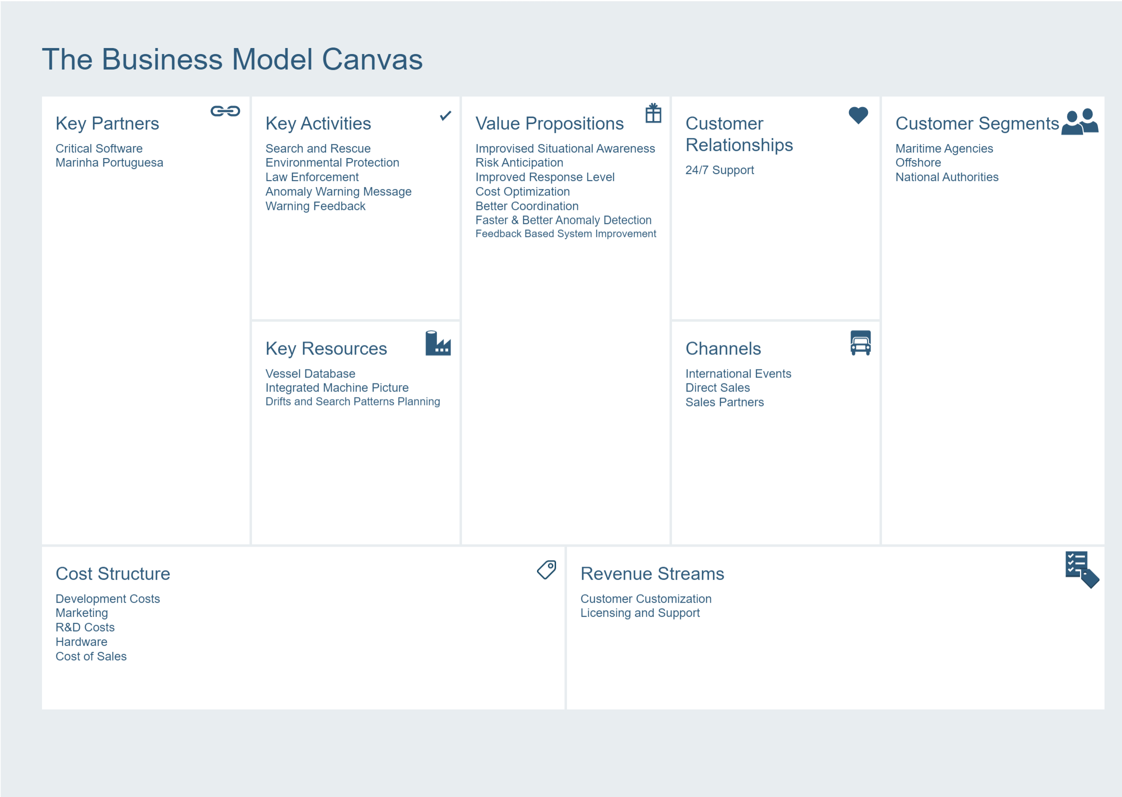


Figure 3.6: Business Model CANVAS

In this instance, the involved clients are Critical Software (CSW) and Marinha Portuguesa. For CSW, the main importance and value of this product is the possibility to utilize their strong presence in international events and their strong sales partners to maximize profit by selling this tool to various potential customers. For Marinha Portuguesa, the main value of this tool is that it allows them to optimize their operations, which, in this case, comprise search and rescue, environmental protection, law enforcement, and anomaly handling (closely related to law enforcement), which can be visualised by the "Key Activities" sector in Figure [3.6].

3.2.4 Scientific Value

As has already been mentioned in sub chapter [3.2.1], from the scientific community's perspective, the most important part of this project, is the development and processing. The means to get to the end result is very important for the community because it allows others to base their work around what is already done, as well as potentially improve their own work. Another important value for the scientific community that is provided by this project is the number of results obtained. Results like these are invaluable to understanding these types of large datasets analysing algorithms. Training large datasets can be complicated, and drawing as many different results from various sources is very valuable to draw conclusions and comparing other work.

3.3 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that ensures the client's needs are met throughout the product development lifecycle. [Crawford and Benedetto 2003] QFD is a system that helps design the product without forgetting the values the customer has in mind so that the product can be shaped around those values. QFD involves the entire company as well as all members of the producer or supplier organization. [Roberts 2007] To understand what QFD means, an analysis of each keyword comprising this term can be done in the following way, as was explained by [Roberts 2007]:

- **Quality** ensures the product meets the customer's needs;
- **Function** describes what must be done;
- **Deployment** derives from the Japanese extension or broadening of activities, which refers to who will do it and when.

The QFD system is currently used by many successful companies around the world such as Toyota, GM, AT&T, and Exxon. This system ensures products are delivered at a faster and lower cost with a customer-driven mindset in place, providing a tracking system for future improvements. [Roberts 2007]

The House of Quality (HOQ) ensures a good QFD, so the first step is to identify customer's high-level requirements. Customer requirements can be understood by what the client expects the product to achieve. This data can be obtained by primary sources, enquiring the customer directly, or by secondary sources, which involve statistics, commercial reports, and business newspapers and trade magazines. [Roberts 2007] The following step is to gather the technical requirements. These requirements are not always identified by customers. They are either technical or regulatory requirements such as legislation, safety requirements, among others. [Roberts 2007] The next step is to attribute weights to each requirement and deploy them on the HOQ. Having the customer requirements in rows and the technical requirements in columns allows us to create a relationship between them with a degree of closeness.

3.3.1 House of Quality

With Figure [3.7], the most important requirements for the customer are the algorithm's understanding of what is an anomaly and what isn't, the warning of said anomaly detection, and the algorithm learning curve itself. With this analysis, the direction of development is established, as it directs towards the initial development of the algorithm itself.

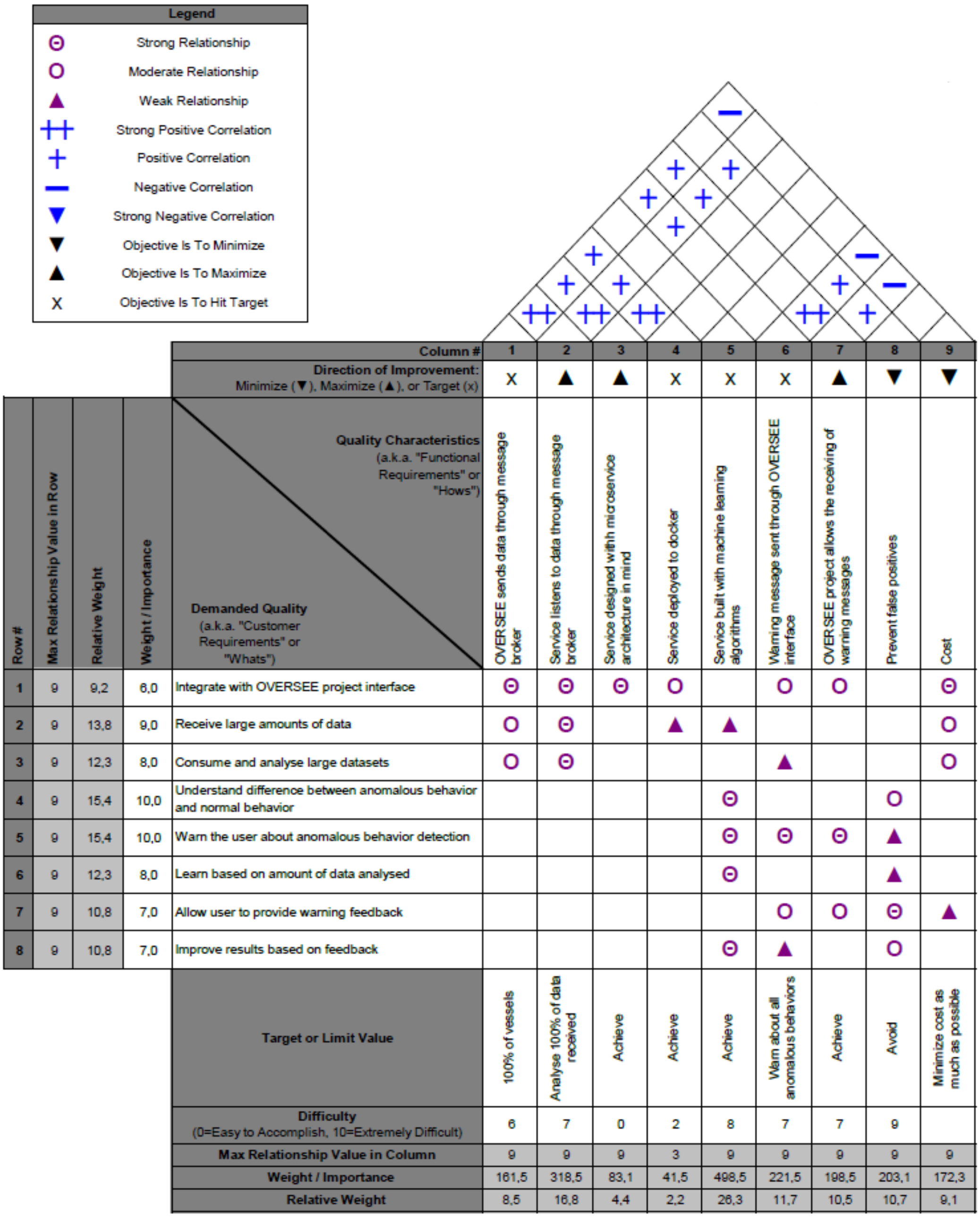


Figure 3.7: House of Quality

Chapter 4

Solution Specification and Design

This section aims to specify the elements, technologies, tools, and processes that will be involved in the development of the solution in itself. Just like it can be seen in Figure [3.4], the solution consists in developing a micro service aiming to solve the problem and fulfill the objectives specified in sub chapter [1.4]

4.1 Chosen Solution

The chosen solution was based on the needs of the stakeholder, keeping in mind the objectives required to be fulfilled to achieve the intended end goal. On Chapter [3.1.4], it was evidenced that the best solution to the proposed problem is to use an LSTM network to predict the future positions of the vessels to later assess their normality.

The objectives and standards of the solution have also been written into the QEF (See Chapter 6.2.2 and Appendix A) as the solution's requirements which will be the guidelines that shall be followed in each phase of the solution's development life cycle.

The development life cycle will be split in different sections:

- **Data Collection**: this step revolves around analysing the data that is available in the scope of this project, and the subsequent preprocessing of it, which is the step where the data is prepared for the future network training stage.
- **Network Design**: designing how the best network architecture will be found and used is key to obtaining optimal results. Another important step is the data visualization, which forms an important step to drawing conclusions.
- **Integration with OVERSEE**: after the best network architecture is found, it is crucial to integrate the solution with OVERSEE to prove its effectiveness and provide a good proof of concept to pave the path for a future fully working production implementation.
- **Anomalous Behavior Detection**: after having the positions predictions, it is necessary to implement a solution where behaviors that deviate normal patterns are identified and reported for further human assessment.

4.2 Data

The first step to the design of any solution that involves machine learning and neural networks, is to identify, gather, and assess the available data that constitutes what will be fed into the network to both train it, and make future predictions. In the case of this solution, the data we are looking for are AIS messages stored in text format that.

4.2.1 Preprocessing

Before feeding the data into the network, it is necessary for it to be decoded, as the AIS binary message will not provide the network the necessary information for it to predict the vessel's next position, so it will be necessary to decode each received message and store it in a CSV file with the necessary information, each CSV containing the information of one vessel ordered by the time it was reported from oldest to most recent.

```
!AIVDM,1,1,,A,142M:EQP001thgBEbRg=Fwwp0000,0*33,1424048399
```

Name	Value	Description
Packet Type	AIVDM	
CHANNEL	A	
Message Type	1	Scheduled Position Report
Repeat Indicator		Default
User ID	271010390	
Navigation Status	1	At anchor
Rate of Turn (ROT)	-729	
Speed Over Ground (SOG)		
Position Accuracy		An unaugmented GNSS fix with accuracy > 10 m
Longitude	27.254575	East
Latitude	37.86186	North
Course Over Ground (COG)	341.9	
True Heading (HDG)	511	Not available (default)
Time Stamp	60	Time stamp is not available (default)
Reserved for regional		Not available (default)
RAIM flag		RAIM not in use (default)
Communication State		Sync state: UTC Direct; Slot Timeout: This was the last transmission in this slot; Slot offset: 0
Communication Sync State		Sync state: UTC Direct
Communication Slot Timeout		Slot Timeout: This was the last transmission in this slot

Figure 4.1: AIS message with its respective decoded information using pyais' decoder (version 2.5.0)

To develop this decoder, the different types of AIS messages and their respective provided information must be analysed (See Chapter 2.4). The type of AIS message that will be used in this project are AIS messages of type 1, which represent a vessel's position report. [USCG 2023] Within the information provided by this type of message, the information that matters in the scope of this project is the geological position indicators **latitude**, and **longitude**, as well as the **speed** and **heading**. The speed and heading are important in the scope of this project as they will facilitate the prediction of the next position with the speed helping to find the distancing between reported positions, and the heading being the orientation of the vessel's bow, i.e., the "front" of the vessel, in relation to the geographical north. The latitude is indicated in degrees as an angle that ranges from $-90°$ at the south pole to $90°$ at the north pole, longitude is indicated also in degrees, as an angular measurement ranging from $0°$ at the Prime Meridian to $+180°$ eastward and $-180°$ westward. Speed is represented in knots, the knot is a unit of speed equal to one nautical mile per hour, which is a unit of measurement used in maritime and aviation contexts and is defined as one minute of latitude.

4.3 Network Design

With the data ready to feed into the network to start predicting the following positions, the network design will start to be implemented by choosing the frameworks and strategies. Firstly, with the data consisting of time series, and with what was analysed in Chapter 3.1.4, the model will be implemented using an LSTM. To design the network architecture, the libraries TensorFlow and Keras will be used. The network shall expect a CSV file named after each respective vessel's MMSI (see Chapter 2.4) with the previously discussed data. (See Chapter 4.2.1)

4.3.1 Preprocessing

The data that is received is converted from **latitude/longitude**, to **distance/bearing**. This step is performed based on an approach seen in [Violos et al. 2020], where it is stated that distance/bearing is used as a way to improve accuracy, as compared to its geolocation representation in latitude/longitude. This data will then be stored as a **Numpy** dataset. This dataset will be split in three subsets, the train set, the validation set and the test set.

4.3.2 Genetic Algorithm

To find the best network architecture, a similar approach to [Violos et al. 2020] will be used, which involves the usage of a genetic algorithm to find the optimal solution (Figure 4.2).

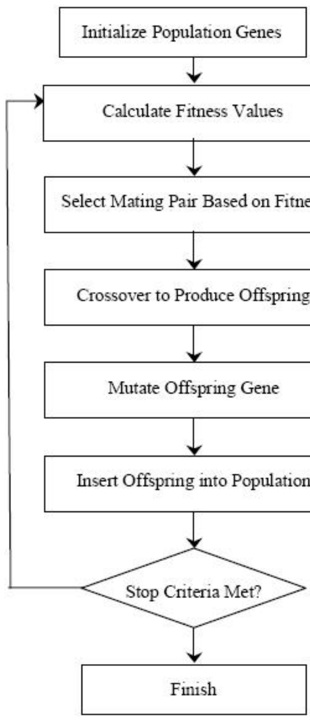


Figure 4.2: Genetic Algorithm Architecture (Retrieved from al-Hddli, Ghiduk, and El-Telbany 2010)

The genetic algorithm used will randomly create a population of networks, which will then be trained and evaluated in order to find its fitness score. Based on the fitness score, a select percentage of individuals is chosen, which in this case will be the top percentage of networks with the best fitness score. In this case, the best fitness is considered the networks with the lower scores, as these are calculated with the distance between the predicted and the actual positions reported by the vessels.

With the best fit networks selected, the next step involves crossing them over to create offspring. These offspring will then be selected randomly to mutate a random allele. These offspring are then inserted back into the network population. With this step finished, we check if the stop criterion are met, which, in this case, is the number of generations that have been crossed over.

4.3.3 Postprocessing, Results Gathering and Visualization

Result visualization and analysis is a fundamental step in developing accurate machine learning solutions. The only way to understand whether the development life cycle is heading towards the correct direction is by having a good visualization of the results given the current state of the network architecture and the fed data.

In the case of this solution in specific, both plotting graphs, and visualizing the obtained results against the actual positions on a map are important.

To understand the evolution of the genetic algorithm, the best scores from each generation need to be kept so that they can later be plotted. This best score is the distance between the actual position and the predicted position, which will be calculated using the **geopy** library.

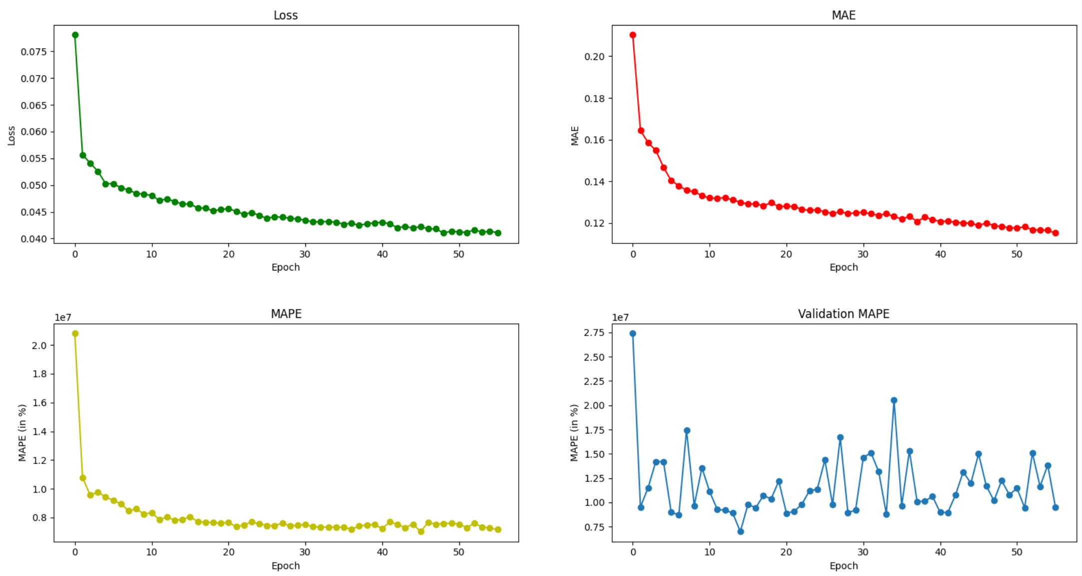


Figure 4.3: Example of graph plotting from a network run by the solution

To draw conclusions about the current network architecture, the necessary data to be analysed are:

- **Loss**: The loss, or loss function, is a measure of how well a machine learning model is performing. It quantifies the difference between the predicted values and the actual values in the training data.
- **Mean Absolute Error (MAE)**: MAE is a metric that measures the average absolute difference between predicted values and true values. It is calculated with $\frac{1}{n}\sum_{i=1}^{n}\text{abs}(yi_{true} - yi_{pred})$
- **Mean Absolute Percentage Error (MAPE)**: MAPE is a metric that measures the average percentage difference between predicted values and true values. It is calculated with $\frac{1}{n}\sum_{i=1}^{n}\text{abs}(\frac{yi_{true}-yi_{pred}}{yi_{true}})*100$.

These will be calculated automatically with the **fit** function from the Keras library, and plotted with the **matplotlib** library.

To visualize the results on a map against the actual positions, after obtaining the predictions, these sets of points will be used with the **Folium** library to plot the map with these positions. The postprocessing phase is important here, where the obtained results on distance/bearing format, will need to be converted back to latitude/longitude, so that they can be pinpointed on the map.

4.4 Kafka and Integration with OVERSEE

To allow the continuous running of the application, there must be a communication with the OVERSEE application in order to receive a constant stream of AIS messages. To achieve this, and since, as was mentioned in Chapter 2.5, OVERSEE uses Kafka to communicate and receive messages from its external sources, Kafka will be used to perform this communication and to serve as a message broker, since it is an optimal solution to handle large streams of data in an ordered manner. (Chapter 2.5.2) This will be achieved following the architecture:

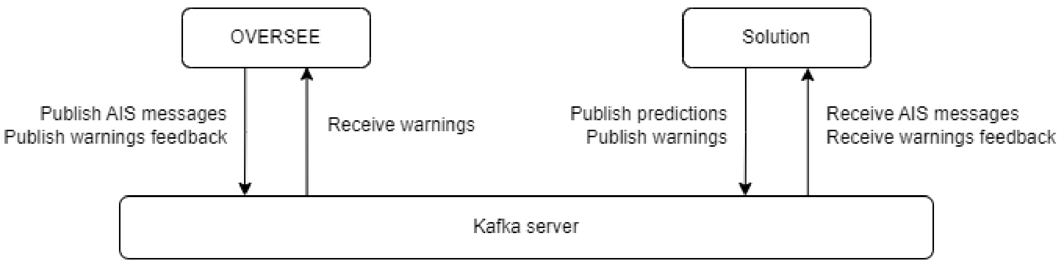


Figure 4.4: Integration Architecture

This Kafka server shall be running in a Docker image hosted inside a CSW's Virtual Machine. OVERSEE shall be able to publish the AIS messages to a Kafka topic, which will be received by the solution that is being developed. After processing this information, the solution shall produce as outputs the predictions for each vessel that it receives, as well as warnings that shall indicate anomalous behaviors, which will be received by OVERSEE. After the user assesses the given warning, if they deem necessary, a feedback report shall be sent back to the solution through another Kafka topic, that shall be received by the solution.

4.5 Anomalous Behavior detection

The final step is to identify the behaviors that are considered anomalous. To approach this, the first step is to gather information on what behaviors are considered anomalous, attributing a label to identify these behaviors. This step is necessary to develop the next step, which is to create a network that will identify these behaviors and fit the scores based on the given labels. Unfortunately, we run into a problem in this step. There is no available data currently to pinpoint these behaviors and create labels on these types of behaviors. (See Chapter 7.1) So this step will be marked as future work (See Chapter 7.2), and a different approach will be taken.

To assess whether a position is considered as anomalous or not, distance between the predicted positions, which were calculated in previous steps, and the true positions will be calculated using **geopy**, as mentioned previously. With this value, a distance shall be considered as anomalous if the distance of that prediction is higher than the median of the distances times 2.5. This value comes from the rationale that 2 shall be given as the range from which the distance may vary, and 0.5 to include tolerance to the network error. These warning reports shall be sent to OVERSEE through the Kafka server relevant topic.

4.5.1 Reinforcement Learning

After the reports are made, feedback from the user follows on whether these reports shall be considered as false positives. This step is called **reinforcement learning**. (See Chapter 2.8)

The user shall only give feedback to the network if the report was a false positive. The network shall use this information to rework and reassess the model based on this new information that was provided by the user.

Chapter 5

Solution Development

Note: some of the code snippets have been suppressed as they were too big to fit in this document. To see the code in full refere to Appendix B.

This chapter explains the procedures that were followed to achieve the successful development of the solution.

This phase can be divided in distinct moments:

- **Choosing the technologies**: the phase where the technologies which will be used to develop the solution.
- **Prioritizing**: on this phase, a kanban board was created in Trello, to manage the tasks that had to be concluded, the ones that were ongoing, the blocked tasks, and the finished tasks.
- **Gathering the Necessary Data**: this step involves gathering all the necessary data to fulfill the necessary dataset which will be used to feed the network aswell as predict the consecutive results.
- **Preprocessing the Data**: preprocessing the data so that accurate measurements can partake.
- **Designing the Optimal Network**: this phase is the most complex one, where the optimal network has to be found through an optimization algorithm.
- **Gathering Results**: is the phase where the predictions are aggregated and metrics can be taken to evaluate the quality of the work done, as well as providing a visualization of what is being predicted.
- **Integrating With Kafka**: this final step allows the solution to run on a continuous basis, so that it can later be integrated with OVERSEE.

5.1 Technologies Used

The selection of the technologies to be used followed the analysis and investigation of the field of study involving the development of the solution. (See Chapter 2) The necessary tools to develop the solution involve code editors, version control systems, libraries, integration tools, and other support tools.

5.1.1 Version Control System

The first step to start the development phase, was choosing a version control system, and a host for the repository. This is an important step as it allows for simple rollback to previous versions of the project, in case there are any loss, errors, or corruption of files. It is also important to track history and changes to the project files to identify probable reasons behind problems faced in the future.

The chosen versioning system was **Git** in conjunction with **GitHub** as the remote repository host. This choice stems from the experience of the involved with the platform. GitHub not only facilitates version control but also offers a streamlined workflow, allowing for seamless integration of feedback, as well as easing the task of sharing the work with the supervisors and relevant stakeholders.

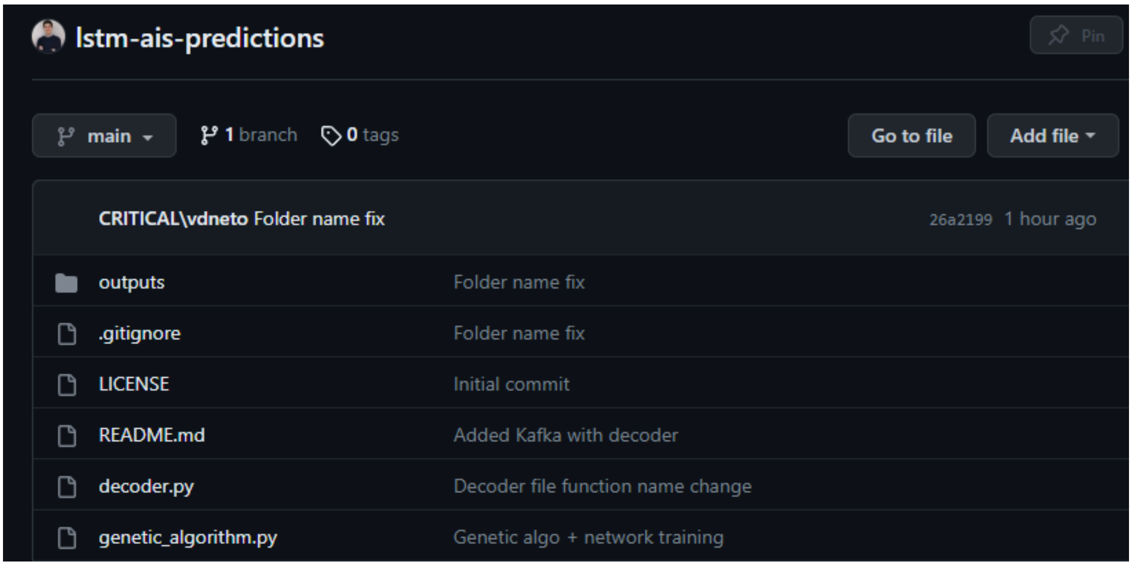


Figure 5.1: Example of some of the folders involved in the project

The project repository can be accessed from Appendix B and includes a "REAMDE.md" file with documentation on how to run the project.

5.1.2 Code Editor

The chosen code editor was **Visual Studio Code**. The reasoning behind this choice stems from this editor being free and open source, aligning with the principles of the development of any academic work. Other reasons include the integrated terminal, which allows for the running of the scripts and its lightweight and speed capabilities. The fact that **VSCode** also comes with an integrated version control system allows for a seamless integration with Git, the chosen control version, which simplifies the version tracking, commits and branching, which streamlines the version control system negating the need for the use of external tools.

5.1.3 Architecture and Libraries

The source code used to develop this solution is **Python**. Python has a rich ecosystem of libraries and frameworks that allow for good structuring of machine learning and deep learning tasks. The existence of very powerful and complete libraries such as **TensorFlow**, **Scikit-Learn** and **Numpy**, make Python, arguably, the most competitive programming language to write machine learning solutions. Its extreme popularity in data science and machine learning derive from its extensive library support, external tools support, and ease of prototyping and experimentation, make it conducive for using in such a project.

The most important libraries used in this project are:

- **TensorFlow and Keras**: TensorFlow is an open source library that is widely used in the creation of machine learning solutions, especially training neural networks to detect and decipher patterns and correlations. Keras is another machine learning library that runs on top of TensorFlow and allows for rapid experimentation with deep neural networks. Keras excels at encapsulating lower level code, easing the process of creating and training models.
- **Numpy**: Numpy is used for matrices and vectors processing, which permits the execution of high level mathematical functions.
- **Scikit-Learn**: Scikit-Learn, also known as sklearn, was chosen for this project as it allows for complementary functions, such as **MinMaxScaler** to be used to scale the data for network training.
- **Pandas**: Pandas is Python library used to manipulate and analyse big data. It has been used in this project to aggregate the datasets into Pandas "dataframes" so that these matrices can more easily be manipulated.
- **Pyais**: This library allows for the decoding of AIS messages (See Chapter 2.4) to gather the necessary information form them.
- **Geopy**: Geopy is used to calculate distances from a given position to the predicted position.
- **Kafka-Python**: Used to allow for the communication with the Kafka server, both for producing information as well as to consume it.
- **Folium**: Map plotting library used to visualize and compare the predicted data against the ground truth.
- **Matplotlib**: Data used to plotting graphs so that the taken metrics from the training and testing phases can be visualized and compared against one another in graphs.

The main project organization followed the following structure (Figure 5.2):

The folder **"outputs"** contains the subfolders:

- **decodedCSVsList**: contains the results from the decoder in CSV format, with the latitude, longitude, speed, and heading of each decoded AIS message.
- **MetricsOutput**: contains the ouputs from training a model, namely, the genetic algorithm's evolution history, and the metrics taken from fitting the model with the given dataset.
- **ModelsOutput**: contains the trained Keras models.
- **predictionOutput**: outputs from the prediction function, namely, the map, the position predictions, and the distances between the predictions and the actual positions.
- **ScalersOutput**: the scalers used to scale the data during the network's training phase preprocessing.

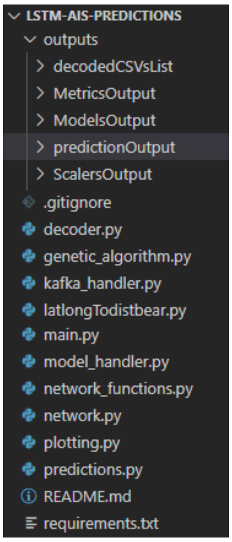


Figure 5.2: Project Structure

Each solution's function is encapsulated in each of the Python files in the structure, the *decoder* being in charge of decoding and saving the AIS messages with the relevant information, namely, the latitude, longitude, speed an heading. The *genetic_algorithm* and *model_handler* files handle the networks' optimization functions through a genetic algorithm, while the *network* and *network_functions* handle the network population creation, as well as the dataset preprocessing, and network training and validation phases. *kafka_handler* handles the listening to the Kafka server in order to receive new positions in real time, and *plotting* and *predictions* files handle the next positions predictions and plotting the metrics and information to graphs and a map. *latlongTodistbear* is an algorithm based on work conducted at the National Technical University of Athens [Violos et al. 2020] which serves the purpose of converting two geographical points in latitude/longitude to distance and bearing.

5.1.4 Integration Tools

Integration tools are tools that are used to deploy and connect the solution to the production server of OVERSEE. Since OVERSEE uses **Kafka**, one of the tools used will be Kafka. To run the Kafka server, the server will be deployed in a **Docker Image** inside a **Virtual Machine** hosted by Critical Software. This Virtual Machine will be deployed using Oracle's **VirtualBox**.

5.1.5 Support Tools

Support tools are tools that are used to provide means of supporting the development of the work, not directly influencing or taking part of the solution's development. **Overeaf** was used to write the theoretical part using LaTeX. **Grammarly** and **ChatGPT** to provide support with the writting and grammar checking. **Trello** was used as a Kanban board to help understand the ongoing tasks, as well as **Excel**.

5.2 Data Gathering

Developing a Neural Network solution requires a very crucial step, which is the gathering of the data that will provide the means of training and validating the network itself. The data gathered (See Appendix C) to first develop the solution is historic data from Marinha Portuguesa, from the months of January and February 2015.

This data consists of a series of log files with AIS position reports from vessels, where the messages are aggregated and ordered from oldest to newest in a series of files, where each file contains the aggregated positions of all vessels' messages from one hour. January 2015 had 31 days, while February 2015 had 28 days which means there are a total of 1416 files.

5.2.1 Continuous Data

After the connection to Kafka (See Chapter 5.6), the solution is now able to handle the processing of concurrent data (See Chapter 5.3). This is possible using Kafka topics.

5.3 Data Preprocessing

To preprocess the data, first, we need to decode the received AIS message:

```python
def decode_single_message(msg):
    total_dictionary = defaultdict(dict)
    try:
        decoded_message = msg.decode().asdict()
        if decoded_message["msg_type"] == 1:
            if decoded_message["mmsi"] in total_dictionary.keys():
                total_dictionary[decoded_message["mmsi"]].update({ list(
    total_dictionary[decoded_message["mmsi"]].keys())[-1] + 1 : { "Speed"
    : decoded_message["speed"], "Latitude": decoded_message["lat"], "
    Longitude": decoded_message["lon"],"Heading": decoded_message["
    heading"] } } )
            else:
                total_dictionary[decoded_message["mmsi"]].update(...)
    except:
        print("Error reading message")
    save_dictionary_to_csv(total_dictionary, csvSavePath)
    return decoded_message["mmsi"]
```

Listing 5.1: Decode AIS message (Python)

Where we first declare a dictionary where the message information will be stored with the relevant decoded information, namely, the latitude, longitude, speed and heading. This dictionary is then saved with:

```
def save_dictionary_to_csv(total_dictionary, outputPath):
    for mmsi, messageList in total_dictionary.items():
        (...)
        with open(outputPath + str(mmsi)+".csv", 'a') as fp:
        (...)
            for messageCount, message in messageList.items():
                fp.write(str(message["Latitude"])+ "," + str(message["
    Longitude"])+ "," + str(message["Speed"] )+ "," +str(message["Heading
    "])+ "\n")
```

Listing 5.2: Saving the messages to a CSV file (Python)

This function saves the dictionary with the previously mentioned information to a CSV file, so that it can be easily handled by the Pandas library later on.

5.4 Network Design and Genetic Algorithm

To initialize the network design and training, the first step is to create the **Network** class, which will allow for the creation of the network object.

```
class Network():
    def __init__(self, nn_param_choices=None):
        '''
        Initializes the Network with its default parameters:
            - accuracy: The accuracy of the network which will be used
    to decided whether it is fit enough or not
            - nn_param_choices: dict of random parameters to be chosen
    from to create the network
            - network: the network parameters
        '''
        (...)
    def create_set(self, network):
        self.network = network
    def train(self, dataset, bestScore, currentMMSI):
        '''
        Train the network with a dataset
        '''
        (...)
    def print_network(self):
        """
        Shows network information
        """
        (...)
```

Listing 5.3: Network Class (Python)

First, we need to initalize the genetic algorithm, which needs a number of generations, and a population. The population number is the number of networks that will be randomly generated and evolved through the generations:

```
bestScore = 100000.0
# Number of times to evole the population
generations = 50
# Number of networks in each generation
population = 15

networks = []
for _ in range(population):
    # Create a random network.
    network = Network(nn_param_choices)
    for key in network.nn_param_choices:
        network.network[key] = random.choice(network.nn_param_choices[key])
    # Add the network to our population.
    networks.append(network)
```

Listing 5.4: Population Creation (Python)

Where *nn_param_choices* are a set of random LSTM network parameters that will be randomly selected for the population creation, and *bestScore* is used to later evaluate each network so that the best can be kept.

After this step, the networks training begins:

```
def train_network(network, dataset, bestScore, currentMMSI):
    trainX, trainY, testX, testY, scaler_X, scaler_Y, Coords = preprocess_data(dataset)
    model, history  = compile_model(network, trainX, trainY)
    error, bestScore = evaluate_model(model, testX, testY, scaler_X, scaler_Y, Coords, bestScore, history, network, currentMMSI)
    return error, bestScore
```

Listing 5.5: Network Training (Python)

This step involves three substeps, the **preprocessing** of the received data, where it is first converted to distance/bearing, from latitude/longitude, and split in two sets of data:

The function used to convert from latitude/longitude to distance/bearing, based on the solution by [MovableTypeLtd 2023]:

$$a = sin^2(\frac{\Delta\varphi}{2} + cos(\varphi_1) * cos(\varphi_2) * sen^2(\frac{\Delta\lambda}{2})) \tag{5.1}$$

Where φ is the latitudes and λ are the longitudes.

$$c = 2 * arctan^2(\sqrt{a}, \sqrt{(1-a)}) \tag{5.2}$$

$$Distance = R * c \tag{5.3}$$

$$Bearing = arctan^2(sin(\Delta\lambda) * cos(\varphi_2), cos(\varphi_1) * sin(\varphi_2) - sin(\varphi_1) * cos(\varphi_2) * cos(\Delta\lambda)) \tag{5.4}$$

```python
def preprocess_data(dataset):
    # Gets the coordinates in dist/bear as a vector of dist/bear coordinates like the original file, but in dist/bear
    dataset_X = transform_dataset(dataset)
    # Delete the first line of the dataset into a new dataset
    dataset_Y = np.delete(dataset_X, 0, 0)
    # Delete the last line of the dataset
    dataset_X = np.delete(dataset_X, -1, 0)
    # Original dataset coordinates
    Coords = pd.read_csv(dataset, engine='python').values.astype('float32')
    # Pushes the first element of the array to last and every other element goes up 1 position, ex: [0,1,2,3,4] -> [1,2,3,4,0]
    Coords=np.roll(Coords, -1, axis=0)
    # Get the train and test datasets divided in g_training_percentage for training, and the rest for testing
    train_size = int(len(dataset_X) * g_training_percentage)
    trainX, testX = dataset_X[0:train_size,:], dataset_X[train_size:len(dataset_X)+1,:]
    trainY, testY = dataset_Y[0:train_size,:], dataset_Y[train_size:len(dataset_X)+1,:]
    Coords = Coords[train_size:len(dataset_X)+1,:]
    # Fit the data to be between 0 and 1
    scaler_X = MinMaxScaler(feature_range=(0, 1))
    scaler_X.fit(trainX)
    trainX = scaler_X.transform(trainX)
    testX = scaler_X.transform(testX)
    # Do the same to the other dataset
    (...)
    return trainX, trainY, testX, testY, scaler_X, scaler_Y, Coords
```

Listing 5.6: Network Preprocessing (Python)

In the above algorithm, the test data is prepared by creating two datasets, the "dataset_X", which is the data used for training and testing the model, and "dataset_Y", which will be our ground truth. They are both then divided in two different datasets, the training, and the testing, 80 and 20 percent respectively. This is important for hyper parameter tuning as well as to prevent overffiting, and avoiding data leakage. This allows the data to be tested on new, unseen data. After this division, the data is normalized from a scale of 0 to 1 using Scikit-learn's *MinMaxScaler*. This is important so that the optimizer can converge faster and more efficiently.

The prepared datasets as well as the scalers used to normalize them are returned, so as to avoid contamination of the datasets with previously seen data.

After this step, the following step is to compile and train the model. The first step is to define the network, beggining with Keras' *Sequential* model definition, where we will add each LSTM with its respective parameters received from the current network previously defined with randomized parameters:

```python
# Begin defining the model
model = Sequential()
# Add each LSTM to the model
if lstms==1:
    model.add(LSTM(units1, input_shape=g_input_shape, activation=
    lstm_activation1, recurrent_activation=recurrent_activation1,
    implementation=implementation1))
elif lstms==2:
    model.add(LSTM(units1, input_shape=g_input_shape, activation=
    lstm_activation1, recurrent_activation=recurrent_activation1,
    implementation=implementation1, return_sequences=True))

    model.add(LSTM(units2, activation=lstm_activation2,
    recurrent_activation=recurrent_activation2, implementation=
    implementation2))
# Add the neurons to each layer
for i in range(nb_layers):
    model.add(Dense(nb_neurons[i], activation=activation[i]))
model.add(Dense(4))
```

Listing 5.7: Network Compilation and Training (Python)

Following this step, we compile the model with the *mean_squared_error*, the relevant loss function, and metrics to be taken, in this case, the Mean Absolute Error, and the Mean Absolute Percentage Error, after this, we call the *fit* function from Keras to train the model with the previously preprocessed datasets (5.6), and the desired number of batch size and epochs. The data is again divided in a 90/10 percent ratio where the 10% is for validation data.This is important to further improve the network's accuracy by introducing previously unseen data. The fit history is saved for posterior plotting for results visualization.

```python
model.compile(loss=g_loss_function, optimizer=optimizer, metrics=[
    metrics.MeanAbsoluteError(), metrics.MeanAbsolutePercentageError()])

trainX=trainX.reshape(len(trainX),1,4)

history = model.fit(trainX[:int(len(trainX)*(1-g_validation_percentage))
    ], trainY[:int(len(trainX)*(1-validation_percentage))],
validation_data=(trainX[int(len(trainX)*(1-validation_percentage)):],
    trainY[int(len(trainX)*(1-validation_percentage)):]), epochs=g_epochs
    , batch_size=g_batch_size, verbose=g_myverbose, callbacks=[
    earlyStopper])

return model, history
```

Listing 5.8: Network Compilation and Training (Python)

With the obtained model, now we need to evaluate its capabilities of making good predictions. For this, we developed the *evaluate* function to calculate the difference between the inferred positions and the true positions, which will give us the test score, so that the score of each network can be compared with each generation's population to assess which of the best network architectures is the best. First, the previously divided dataset will be used with the *predict* function from Keras to get the predicted following positions, which are then transformed back from their normalized counterparts:

```
testPredict = scaler_Y.inverse_transform(model.predict(testX))
testY = scaler_Y.inverse_transform(testY)
```

Listing 5.9: Following Positions Predictions (Python)

With the obtained predictions we now need to compare these results with the ground truth. For this, we need to start by postprocessing the data by converting it back to latitude/longitude, from distance/bearing. For this, we use the following equations [MovableTypeLtd 2023]:

$$lat_2 = arcsin(sin(lat_1)*cos(\frac{dPred}{R})+cos(lat1)*sin(\frac{dPred}{R})*cos(brngPred)) \quad (5.5)$$

$$W = sin(brngPred) * sin(\frac{dPred}{R}) * cos(lat1) \quad (5.6)$$

$$Z = cos(\frac{dPred}{R}) - sin(lat1) * sin(lat2) \quad (5.7)$$

$$long_2 = lon_1 + arctan^2(W, Z) \quad (5.8)$$

Where lat_2 and lon_2 represent the latitude and longitude of the predicted positions, and lat_1 and lon_1 represent the previous point. R is the earth radius, and *dPred* and *brngPred* are the predicted distance and bearing.

The test score is then measured by calculating the average distance between the predicted positions and the actual positions. If the current network's score is better (in this case lower, as lower distance means a better result), then it saves (or overwrites, in case one already exists) the model and its respective scalers and history.

After each network has been trained and evaluated, they are sorted and ran through the genetic algorithm's next step, which is **evolving** the population. In this step, the networks are sorted by score in a list called *graded*. The *RETAIN_PERCENTAGE* is the percentage of networks that are going to be kept for the next generation and consist, in this case, of the top 40% networks with the best scores. To maintain some genetic diversity, and not only the top scores, we also select randomly some of the networks from the networks that weren't being kept, in this case, with a 10% chance of being kept (See Chapter 6):

```
retain_length = int(len(graded)*RETAIN_PERCENTAGE)

# The parents are the kept networks
parents = graded[:retain_length]

# Retain some of the parents networks from the ones that aren't going to
    be kept
for individual in graded[retain_length:]:
    if SELECT_PERCENTAGE > random.random():
        parents.append(individual)
```

Listing 5.10: Selecting the Networks to be Kept by the Genetic Algorithm (Python)

After retaining the best networks, the next step of the genetic algorithm is to breed them and produce offspring until we have the desired amount of networks. This breeding step iterates through the parameters of each parent and randomly selects one of the parameters to be kept by the produced offspring:

```python
def breed(mother, father):
    offspring = []
    for _ in range(2):
        child = {}
        for param in nn_param_choices:
            child[param] = random.choice([mother.network[param], father.network[param]])
        (...)
        if MUTATE_PERCENTAGE > random.random():
            network = mutate(network)
        offspring.append(network)
    return offspring
```

Listing 5.11: Breeding the Networks (Python)

Code snippet 5.11 represents this step, where two offspring are being produced from each parent network pair. Before returning the produced offspring, we mutate some of their alleles randomly to obtain more genetic diversity. This step is important as it improves the algorithm's global searching capability, helping prevent premature convergence, which would make the algorithm stuck in local optimums. The chance of being mutated is given by *MUTATE_PERCENTAGE*, in this case, a chance of 10% (See Chapter 6)

```python
def mutate(network):
    # Choose a random key.
    mutation = random.choice(list(nn_param_choices.keys()))
    # Mutate one of the params.
    network.network[mutation] = random.choice(nn_param_choices[mutation])
    return network
```

Listing 5.12: Mutating the Networks (Python)

After all generations have passed, the list of networks produced over the generations needs to be sorted from best score to worst, so that the top network with the best score is kept.

5.5 Results Gathering

With the trained model and its respective saved scalers, we can start predicting the following positions by loading the model and giving it new unseen positions:

```
# Load the model and preprocess it
model = keras.models.load_model(model_to_be_loaded + str(currentMMSI) +
    '.keras')
np.set_printoptions(formatter={'float_kind':'{:f}'.format})
transformedPredictionDataset = preprocess(dataPath + str(currentMMSI) +
    '.csv', True)
# Load the saved scalers
scaler_X = joblib.load(scalerXFilename + str(currentMMSI) + 'X.save')
scaler_Y = joblib.load(scalerYFilename + str(currentMMSI) + 'Y.save')
```

Listing 5.13: Loading the Saved Model and Respective Scalers for Predictions (Python)

The next step is to make the predictions and reverse transform them using the *Y Scaler*, which is the target scaler that was saved during the model training phase, applied to reverse transform (with MinMaxScaler) the predicted dataset back to distance/bearing format:

```
 # Transform the predicted output
transformedPredictionDataset = scaler_X.transform(
    transformedPredictionDataset)
# Inverse transform the predicted results for printing
testPredict = scaler_Y.inverse_transform(model.predict(
    transformedPredictionDataset))
```

Listing 5.14: Model Predictions and Results Reverse Transformation using Target Scaler (Python)

With the predicted values in distance bearing, it is then necessary to convert them back to latitude/longitude so that the results can be assessed and the map plotted:

```
# Load ground truth
trueCoordinates = pd.read_csv(g_testingDataPath + str(currentMMSI) + '.
    csv', engine='python').values.astype('float32')
# Convert predictions to latitude/longitude, from distance/bearing
resultsList = convert_to_latitude_longitude(testPredict,trueCoordinates)
```

Listing 5.15: Results PostProcessing (Python)

The *resultsList* is a list of sublists, where each sublist has three values, the predicted position, the true position, and the distance between them in kilometers.

It is then necessary to assess whether there are anomalous behaviors in the predicted values

```
# Calculate average predicted distances
average = 0
for prediction in resultsList:
     average += prediction[DISTANCE]
average = average/len(resultsList)

# Report anomalous behavior if one exists
for value in  resultsList:
     if value[DISTANCE] >= (average * 2.5) and value[DISTANCE] >
    THRESHOLD:
          producer = KafkaProducer(bootstrap_servers=[SERVER], api_version
    =API_VERSION)
          producer.send((...))
```

Listing 5.16: Anomalous Behavior Assessment (Python)

In this case, only anomalous positions where the distance between the true and predicted positions are too large are being analysed,. This approach stems from the challenges in the dataset faced where there were no labeled datasets with anomalous behaviors. (See Chapter 7.1) It was considered that a position would be reported as an anomalous behavior if the predicted position is higher than the average position distance times 2.5, and also larger than a given threshold of 3 nautical miles. This thresholds were given to allow for some tolerance as these AIS reports may not be 100% accurate, and 3 nautical miles is not a very distant prediction from the actual true value.

5.5.1 Results Plotting

Plotting the results was done with the **matplotlib**, where the results being plotted involve the model's *loss*, mean absolute percentage error, *MAPE*, the model's mean absolute percentage error, **MAPE**, and the genetic algorithm's best score evolution throughout the generations. This can be done for each of these metrics with:

```python
# Saved metrics file
path = METRICS_PATH + str(currentMMSI) + '.save'
    with open(path, "rb") as file_pi:
        history = pickle.load(file_pi)
# Plot multiple graphs
figure, axis = plt.subplots(LINES, COLUMNS)
axis[0, 0].plot(history['loss'], marker='o', color='g')
axis[0, 0].set_title("Loss")
(...)
plt.show()
```

Listing 5.17: Plotting a Graph with Mathplotlib (Python)

5.5.2 Map Plotting

```python
def create_map(resultsList, currentMMSI):
    # Correct coordinates
    coordinates = pd.read_csv(VESSEL_POS_FILE + str(currentMMSI) + '.csv
', engine='python').values.astype('float32')
    (...)
    # The points from the correct coordinates
    points = []
    for i in range(len(latitudesList)):
        points.append([latitudesList[i],longitudesLits[i]])
    # Define map
    map = folium.Map(location=[latitudesList[0],longitudesLits[0]],
    zoom_start=13)
    # Create the markers with the correct points and add them to the map
    for index, lat in enumerate(latitudesList):
        folium.Marker([lat,longitudesLits[index]],popup=('Position{} \n'
    .format(str(index+1))),icon= createNumberMarker(color='blue',number=
    index+1)).add_to(map)
    # Add the lines to the correct points
    folium.PolyLine(points, color='blue', dash_array='5').add_to(map)
    # Repeat for the inferred values
    (...)
    # Save the map as html
    map.save(MAP_SAVE_PATH + str(currentMMSI) + '.html')
```

Listing 5.18: Plotting the Map with Folium (Python)

To plot the map, we use **Foliun**. Folium is a Python library for creating interactive and visually appealing maps. It is built on top of the Leaflet JavaScript library. On snippet 5.18, we begin by loading the correct coordinates and divide them into two lists, one with the latitude, and the other with the longitude informations, creating a list of points to be added to the map. We then define the folium map, with the beginning position being the first position of the list. After this, we add the positions to the map as *Markers*, where each *Marker* will then be connected to the previous one and to the next one using Folium *PolyLines*, which are segmented lines used to better visualize the route of the vessel throughout the *Markers* on the map. The final step is to save the map as an HTML file so it can be easily opened in a browser to visualize the map with the added *Markers* connected with *PolyLines*.

5.6 Integration with Kafka

As explained previously in Chapter 2.5.2, Kafka is a message broker that allows for easy communication with OVERSEE. The first step to install Kafka, was to make sure **Java** is installed, as Kafka is developed using **Scala** and **Java**. After this, Kafka is installed via their official website, and its installation folder put in the root folder of the provided virtual machine.

Starting Kafka requires **Zookeeper** to be running as well, which can be performed using a command in the command line inside the installation folder:

```
.\windows\zookeeper-server-start.bat .\config\zookeeper.properties
```

followed by:

```
.\bin\windows\kafka-server-start.bat .\config\server.properties
```

to start Kafka. These servers will need to be running concurrently on the virtual machine. To connect to Kafka with our solution, we use the **Kafka-Python** library. This library can be used to create a Kafka **Consumer**, which will allow for the receiving of positions from the relevant server **topic**:

```python
 # Create Kafka consumer and listen to server
consumer = KafkaConsumer(KAFKA_TOPIC, bootstrap_servers=[KAFKA_SERVER],
    api_version=KAFKA_API_VERSION)
for message in consumer:
    (...)
consumer.close()
```

Listing 5.19: Creating a Kafka Python Consumer and Listening to the Server (Python)

Each received message from the server will trigger a message decoding step, which will be saved to the respective vessel's CSV file, where each 100 lines, the model is retrained, while if different from 100 lines, a prediction for the position will be made, and the results saved and evaluated. A map is also created, or updated with the new position if it already exists, each time the solution receives a new position. (Snippet 5.20). As mentioned in Chapter 5.5, snippet 5.16, our Kafka also needs to send reports through a topic to OVERSEE with the anomalous positions warning. This can be done by creating a topic with:

```
kafka-topics.bat --create --bootstrap-server SERVER //
--topic anomalousBehaviors
```

The consumer will then listen to the server and begin receiving anomalous positions reports. We can test this on our local machine if we connect to the server with:

```
kafka-console-consumer.bat --topic anomalousBehaviors //

--bootstrap-server localhost:9092 --from-beginning
```

```python
# (...) While inside Kafka listening step
# Decode message
currentMMSI = decode_single_message(message)

# Get number of lines
with open(CSV_FILE_PATH+ str(currentMMSI)+".csv", 'r') as decodedFile:
    numLines = len(decodedFile.readlines())
# Create a model every 100 lines
if numLines%100 == 1:
    start_model_creation(currentMMSI)
elif numLines > REQUIRED_LINES:
    resultsList = predict_position(currentMMSI)
    create_map(resultsList, currentMMSI)
```

Listing 5.20: Logic while Listening to the Kafka Server (Python)

5.7 Faced Problems and Final Project

The final project can be found on GitHub (See Appendix B).

As will be discussed in Chapter 7.1, it was not possible at the time to have the solution deployed on a Docker images due to timing and performance constraints, so it is currently on GitHub available for testing and tuning. Another problem faced was the lack of labeled datasets so that anomalous position reports could easily be identified.

In sum, the biggest problem faced during the development phase of this project was the lack of a good labeled dataset to identify what was being investigated in this project, and an alternative had to be developed. This compromised the reinforcement learning phase, which was completely postponed as future work.

Chapter 6

Experimentation and Evaluation

Experimentation and evaluation are concepts followed by researchers which allow them to make smart decisions, pick paths to take, think critically and take conclusions. According to [Cohen 1991], most papers and scientific research lack key points regarding experimenting and trying the solution. Empirical work in scientific research links the developed solution with the theory behind the investigation. Experimentation consists in obtaining results while evaluation assesses them, concluding, and possibly redirecting the solution implementation. In machine learning, evaluating the outcomes is paramount in fine-tuning parameters. This chapter focuses on defining the experimental design plan as well as the processes defined in the solution evaluation and its analysis, after which the obtained results will be presented and analysed.

6.1 Experimental Design

The experimental design is the process of planning and conducting an experiment to test a specific hypothesis or answer a research question. According to [Gomes 2016] an experimental design includes:

- **Claims or hypotheses** is the theory or problem the research is trying to corroborate or solve;
- **Experimental and control conditions** provide a degree of isolation so that the solution can be tested against unseen data;
- **Variables** can be dependent or independent, and will affect results based on their attributed values ;
- **Test apparatus** is the equipment and software used to execute the experimentation process;
- **Protocol to run the experiment** defines the steps involved in running the experiment.

Typically, the first testing actions involve testing the apparatus, not the hypotheses, with a pilot experiment. The pilot experiment helps to adjust variables, see whether the protocol works and to provide initial data so that an initial statistical analysis can be calculated. It is necessary to select the data that will be used in this campaign. Usually, for testing purposes, any data can be used, even if unlabeled, as the network's testing phase is not yet being processed. This is called a sample, which is a subset drawn from the population, which represents the complete set of data to study. [Gomes 2016] The experimental life cycle involves exploration, hypothesis construction, experimenting, data analysis, and conclusions. Exploration involves a deep knowledge of the datasets that are used and their structure.

Without this, it is impossible to teach the model how to identify the patterns sought after. Exploratory data analysis involves a lot of statistics that can be calculated from particular values of the data that is being utilized. In the case of this experiment, the data is structurally divided in several parameters, such as the vessel ID, position, and speed, which have different types of measurements. In the mentioned examples, the vessel ID is a nominal categorical parameter, which provides a name, numerically in this case with no scale implied. The vessel position and speed are quantitative continuous parameters, which means they scale in increments. Some of the calculations that can be done to organize the data are the mean, median, standard deviation, etc.

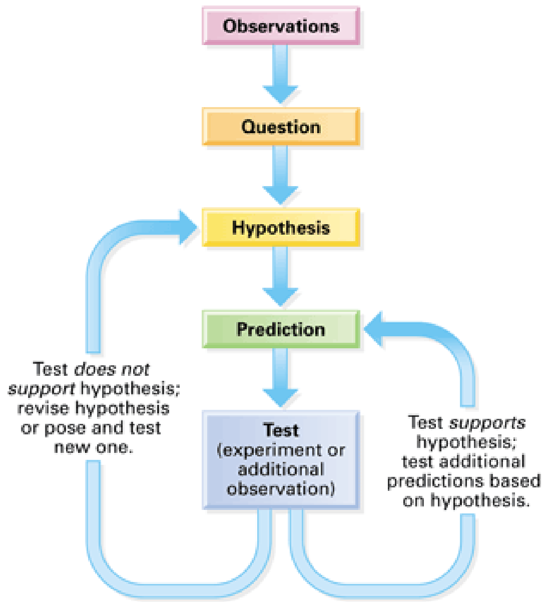


Figure 6.1: Hypothesis Testing according to [Gomes 2016]

To test the hypothesis, there must be a prediction of what is going to happen, and then that prediction is tested to corroborate what is expected. If the test checks out with what was predicted, this means the test supports the hypothesis and additional predictions based on the hypothesis can be made to be tested against. If the test outputs do not meet the predictions, this means the test does not support the hypothesis and it must be revised or completely scratched for a new one.

6.1.1 Hypothesis

In the context of this project, the problem explored in chapter [1.3] states that a way to detect anomalous behaviors in maritime traffic is required by the OVERSEE project. To satisfy this need, a Machine Learning solution based on LSTMs algorithms will be used to fulfil this problem. This solution intends on proving that an LSTM algorithm can accurately provide warnings on the detection of anomalous behaviors with a high accuracy, so the hypothesis being investigated in this project is:

Can a Machine Learning Solution based in LSTMs algorithms provide accurate results in the detection of anomalous behaviors in maritime traffic?

6.1.2 Experimental Conditions

Overfitting is caused by training the model in such a way that it becomes too specific to the data provided to train it. Overfitting caused generalization problems, in which the model is not able to accurately predict outputs based on new, unseen data. This is a problem, especially on the context of this project, as its goal is to continuously receive new data 24/7. To solve the overfitting problem, the method used is cross-validation. Cross-validation is a technique that evaluates the performance of a model by dividing the available data into multiple portions. [2] K-fold cross validation works by dividing the data in k-folds. In this specific case, the data will be divided in 3 folds, one for the training, one for testing, and one for the validation. The training portion is used to train the network, i. e. teach what it needs to flag as an anomalous behavior or not. This portion represents the largest fold of the cross-validation, with 80% as the chosen value as a large amount of data is required to train the network. The remaining 20% of the available data will be used as testing data. This testing portion of the data allows for generalization testing of the network, to evaluate how the model behaves in the presence of new, unseen data. The validation portion consists of 10% of the 80% chosen for the training phase which will be used for validating the model. Furthermore, when testing the model, after it has been defined, new, unseen data will be fed to the model for predicting the following results, which we expect to achieve a certain degree of accuracy. (See sub chapter [6.2.4])

6.1.3 Test Apparatus

Testing will be done using VSCode and Google Colab. The language used in these tests is Python, with the TensorFlow library. Google Colab will be used to mount the data together with the model so that it can train with access to the portion of the data made available. The data utilized for this is obtained by decoding given AIS messages (provided by Marinha Portuguesa) into CSV format, so that they can be digested by the network.

6.1.4 Experimental Protocol

This step aims to designate and elucidate the steps to collect and analyze the data and some ethical considerations. The data is supplied by Marinha Portuguesa. The data provided dates back to the entire months of January and February 2015, and is organized in AIS message log files, each representing the time span of an hour at the given day, where each line represents a vessel's information at a given timestamp. The reason behind this time frame derives from privacy and confidentiality reasons. The data obtained derives from observations, the relevant vessels are obliged to report their information to nearby vessels and control radars. The sample size is very large, consisting of over 500000 (five hundred thousand) reported positions in the span of each hour, which amounts to more than 800 million position reports in the time frame available. Note that only a part of this dataset will be used for this step, due to limited computational resources. Weights and biases are the main variables involved in training the network. To provide an headstart, i. e. not initializing these values as 0, as that will consume much more computational power, transfer learning techniques will be used. Transfer learning techniques in Machine and Deep Learning consist in reusing other proven experiments' pre-trained models as the starting point [Brownlee 2017].

6.2 Solution Evaluation

Following the experimentation phase, the results produced must be evaluated to analyse its quality. [Gomes 2016] To evaluate the quality, the methodologies used include the creation of error measurement techniques, which will be explained in more detail on sub chapter [6.2.4]. Furthermore, to assess not only the quality of the algorithm itself, but to evaluate the adequation of the solution to properly meet the client's expectations, which were discussed on sub chapter [3], the Quality Evaluation Framework (QEF) was created in the scope of the project. The QEF intends to provide a metric to assess business process quality and requirement fulfillment, i. e. to what extent the proposed requirements were fulfilled. [Escudeiro and Bidarra 2008] This is explained in detail on sub capter [6.2.2].

6.2.1 Evaluation Goals

The principal objective to this chapter is to evaluate both the algorithm's quality, as well as the solution's overall quality, according to the client's expectations, further explained in sub chapter [3.2.3]. The algorithm's quality assessment affects the evaluated parameters. To assess the quality of the solution itself, the QEF is utilized to understand to what extent the client's needs are met (Further explained in sub chapter [6.2.2]).

6.2.2 Quantitative Evaluation Framework

Before implementing the solution, the requirements were established, in order to understand where the project was standing at each stage of development. Analysing the requirements after the solution has finished its development phase is a good way to evaluate a solution's quality based on the compliance with these. However, this originates a problem, there are no metrics or methodologies to exactify the quality of the solution based on each of the requirements fulfillment ratio.

The Quantitative Evaluation Framework (QEF), was proposed by Paula Escudeiro and José Bidarra in 2006 to solve this problem. [Escudeiro and Bidarra 2008] To evaluate the quality of a project, a measurement that quantifies the degree of reach of a quality characteristic is necessary, so to implement a quality characteristic, it is necessary to establish a metric capable of quantifying it and make a measurement to determine its fulfillment, resulting from the metric application. [Escudeiro and Bidarra 2008] The QEF model has as basis software engineering best practices relative to the creation and development of conceptual models to evaluate digital contents.

The QEF model proposes the inclusion of a quality benchmark based on :

- **Dimensions** Dimensions are the set of software components that will be assessed to determine the solution's quality and comprise factors, which comprise requirements. Each dimension has its factors, which have requirements that need to be fulfilled, and a set of metrics that intend to be used in the final calculations for the quality assessment. For example, the "Maintenance" factor can belong to the "Security and Maintenance" dimension, as they are closely related and cover requirements from this subset, such as requirements referring to the coding guidelines, and requirements dealing with the solution's documentation.
- **Factors** are further subdivisions of a dimension in the specific group of requirements that can be directly measured and place in the same subcategory based on their aims

and goals. For example, the above mentioned requirements can be placed on the same factor, as they already belong to the same dimension, which could be, for example, the "Maintenance" factor, as they both cover requirements that directly influence the ability to maintain the application during its lifetime.

- **Requirements** A software requirement is a description of the features and constraints that a software system must possess to satisfy the stakeholders' needs and expectations. It serves as the foundation for software development by providing a clear and unambiguous understanding of what the software is intended to achieve. In the context of the QEF, requirements are the atomic element that make up the factors within each dimension. Requirements can be defined as functional describing the system's functionalities, and non-functional, specifying qualities such as performance, design guidelines, and security.

To build the QEF for this solution, firsly, the requirements were specified as in Appendix A. For this, a brainstorming technique was used with the objectives of the solution in mind. After this, the next step was to classify these requirements into **Factors**, which ended up being the ones in the attachment previously mentioned. The next step was to classify these Factors into **Dimensions**, which ended up being separated into "Functionality", "Quality and Reliability", and "Security and Maintenance".

According to [Escudeiro and Bidarra 2008], each requirement must be attributed with its specific weight, which establishes the importance of each requirement in the solution's scope. Following this step, a page was created for each Dimension to explain what it means to complete each requirement's fulfillment percentage. This can have the values {0, 25, 50, 75 and 100}, with 0 being completely unfulfilled, and 100 meaning the requirement is fully implemented in the final solution.

Finally, the last step is to, at the end of the development life cycle, the column with the fulfillment percentage was filled with each requirement implementation percentage, i.e., the wfk % column. With this column filled, the quality metrics can begin to be calculated.

The first metric is the **Factor Weight** (W_{ij}), which is calculated by the count of the total requirements in a given factor, divided by the count of the total requirements in a given dimension:

$$W_{ij} = \frac{\sum rj}{\sum ri} \tag{6.1}$$

Where $\sum rj$ is the count of the total number of requirement in factor j and $\sum ri$ is the count of the total number of requirements in dimension i. Each factor is then evaluated with [Escudeiro and Bidarra 2008]:

$$Q_j = \frac{1}{\sum(rwjk)} * \sum(rwjk * wfk) \tag{6.2}$$

Where $rwjk$ is the weights of the requirements of the current factor, and wfk is the fulfillment percentage of the requirements of the current factor. With the calculated factor quality Q_j, the next calculation is the dimension's quality Q_i, which is given by the sum of the products between each factors' quality, Q_j, and its respective weight, W_{ij}:

$$Q_i = \sum(Q_j * W_{ij}) \tag{6.3}$$

To evaluate the global deviation, we use [Escudeiro and Bidarra 2008]:

$$D = \sqrt{\sum_i (1 - \frac{Q_i}{100})^2} \tag{6.4}$$

Where Q_i is each dimension's quality.

Finally, the system's quality is determined by [Escudeiro and Bidarra 2008]:

$$q = \frac{\log(\frac{1+Q_i}{2})}{3 * \log(2)} \tag{6.5}$$

In conclusion, the QEF measures the quality of the project in a quantitative fashion allowing for the proper validation of the solution by dividing the requirements in dimensions and factors, and attributing weights to them, where each requirement's fulfilment percentage will contribute towards the overall quality of the project, with the weight metric affecting how much each individual requirement will contribute towards this objective. [Escudeiro and Bidarra 2008]

6.2.3 Evaluated Parameters

Measuring the error present in an ML algorithm provides a good enlightenment to what can be improved over new iterations of the development lifecycle, methodologies detailed in sub chapter [6.2.4]. The calculated error is directly influenced by the algorithm's parameters and hyperparameters, which are adjusted according to it. The adjusted parameters include the biases and weights. Weights represent the importance of each input while making predictions, while biases are scalars that help tune systematic errors. These parameters are fine tuned accross the training epochs of the models. Hyperparameters, as opposed to the parameters, are chosen before the training begins, and are not learned from the data presented to the algorithm. The hyperparameters include the number of hidden layers, learning rate in gradient descent calculation on the backpropagation step, number of epochs, and the choice of activation function [Nyuytiymbiy 2022] These hyperparameters must be fine tuned according to the obtained results, as they affect the algorithm's performance and accuracy.

6.2.4 Evaluation Methodologies

To evaluate whether the obtained results are considered accurate or not, we must undertake and evaluate a series of metrics taken from the model's genetic algorithm evolution, the model's training phase, testing phase, as well as the predictions' results afterwards. To better visualize the data that is being undertaken and predicted, we will also plot it on a map to better understand what is happening.

The chosen loss function has been the **Mean Squared Error** (MSE). This metric has been chosen since it is a differentiable function, which is a crucial property for optimization algorithms such as the ones which use gradient descent, commonly used in regression problems such as this one. It is also convex, meaning it has a single global minimum, facilitating optimization, and is also sensitive to deviations, which is desirable in regression tasks, as it is important to penalize large errors in comparison with smaller ones. MSE is calculated with:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(ytrue_i - ypred_i)^2 \tag{6.6}$$

Where n is the number of samples, ytrue is the true position, and ypred is the predicted position.

Other metrics taken from the training history are the mean absolute error (MAE), and the mean absolute percentage error (MAPE), which means of calculation have been discussed in chapter [4.3.3].

Deriving from the fact that we don't have labels on the data, the way to detect anomalous behaviors has been decided to be based on the distance between the predicted positions and the true positions, and this threshold has been decided to be of 3 nautical miles (approximately 5550 meters), and a good mean absolute error percentage below 8%.

6.2.5 Sources of Information

To obtain the information mentioned in this chapter [6], several websites were used. To search for papers, Google Scholar came in handy. Google Scholar is a freely accessible web search engine that indexes the full text or metadata of scholarly literature across an array of publishing formats and disciplines. It allows users to search for academic and scholarly literature, including articles, theses, books, preprints, abstracts, conference proceedings and technical reports, from a variety of academic publishers, professional societies, online repositories, universities, and other web sites. It is often used by researchers and academics to find scholarly articles and papers on a particular topic. The most used websites to search for information, and that came up the most in Google Scholar, were ResearcheGate, ScienceDirect, and IEEE Xplore.

Researchers and scientists can share their work, work together, and connect with colleagues on the social networking site ResearchGate. With millions of members from all around the world, it was formed in 2008 and has since expanded to become one of the largest professional networks in the scientific community. Researchers can build profiles on ResearchGate, share their articles and research data, ask and respond to questions, and interact with other researchers who are working in related disciplines.

IEEE Xplore is a digital library that offers access to technical publications in the domains of electrical engineering, computer science, electronics, and related disciplines. These materials include journal articles, conference proceedings, and standards. The Institute of Electrical and Electronics Engineers (IEEE), the biggest technical professional association in the world committed to promoting technology for the benefit of humanity, is responsible for maintaining it. Since IEEE Xplore is a subscription-based service, a membership or institutional subscription is required to access the full-text content, so the ISEP's institutional account was used to access this content.

Digital database ScienceDirect offers access to a variety of scholarly and scientific research publications. From a variety of topic areas, including the physical sciences, engineering, biological sciences, health sciences, social sciences, and humanities, it offers full-text journals, books, and reference materials. Researchers, scholars, and students commonly utilize

ScienceDirect, one of the largest electronic repositories of scientific research anywhere in the world. Elsevier, the publisher, maintains it.

The investigation was conducted having the "3 step approach" in mind. First, the title, date, abstract and introduction are read. If these are relevant for the project, then the examination continues to the second step, analysing the body of the paper and the third step is to take notes of the relevant content included in said paper.

The used keywords to look for information regarding this chapter where: **Machine Learning**, **Maritime Traffic**, **Neural Networks**, **Machine Learning Evaluation**, **Machine Learning Optimization**, **Anomalous Behavior detection**.

To finalize, information present in the PDF and PowerPoint presentations in the context of the Tese / Dissertação / Estágio (TMDEI) curricular unit's modules lessons was also gathered and used in the development of this project.

All information is properly referenced in the Bibliography.

6.3 Results Analysis

This chapter intends to layout the obtained results, discuss them, and evaluate how they could be improved. As was discussed in sub chapter [6.2.4], metrics were taken for the genetic algorithm's generations evolution, the network training history with the loss function, MAE, and MAPE, the prediction's results, including its distance from the true positions, the MAE and MAPE. Tu finalize this chapter, a visualization of the predicted positions against the true positions on a map will also be provided for better insights.

6.3.1 Genetic Algorithm

To evaluate the quality of the genetic algorithm, we started with 400 generations, and a population of 15, with a retain rate of 40%, mutation rate of 20%, and selection rate of 10%, which equates to how many percent of the networks that were not supposed to be kept, are kept anyways, which means 10% of the remaining 60%. This number of generations provided sufficient generations to allow for the initial exploratory test to begin and figure how many generations were necessary. The results are presented as follows:

The lower the average score, the better, since this score represents the distance between the true positions and the predicted positions. From analysing this graph (Figure 6.2), two things were clear. Not so many generations were necessary, and the results were likely to be stuck on local optimums in the early evolution of the genetic algorithm. To fix these two issues, the generations were reduced to 150, and the mutation percentage reduced to 10%, values that were maintained throughout the course of development.

After changing these metrics, new results were drawn, in which we can conclude that most of the convergence happens before the 60 generations, which was the rationale as to why this value was again reduced on the final solution (See chapter 5.4). The mutation chance reduction also presents more "steps" in the line on the graph, which means the solution is reducing the exploratory space, finding the minimum more easily. (See Figure 6.3).

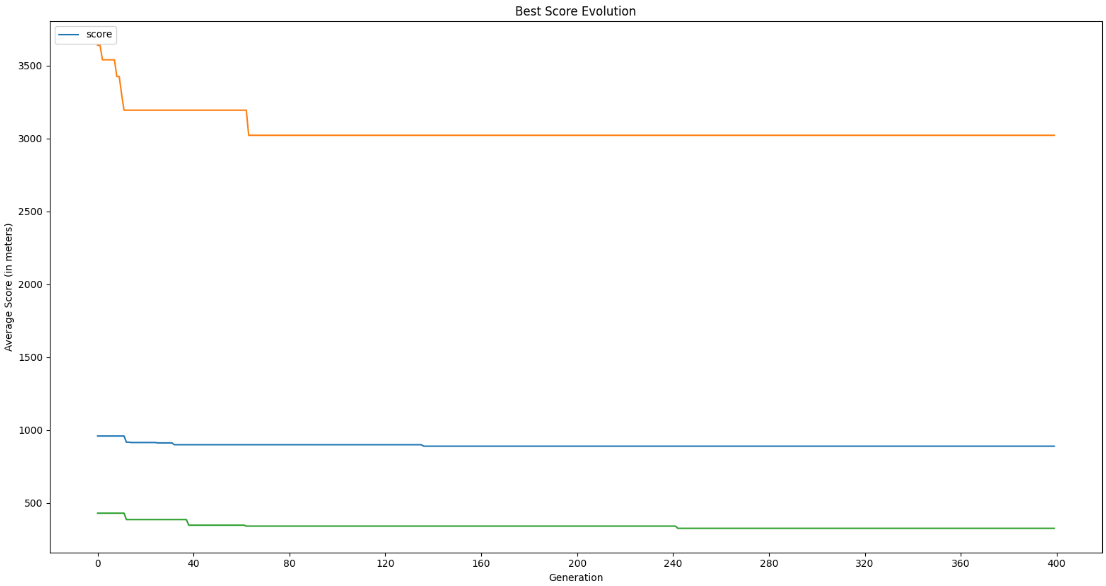


Figure 6.2: Score Evolution from 400 Generations

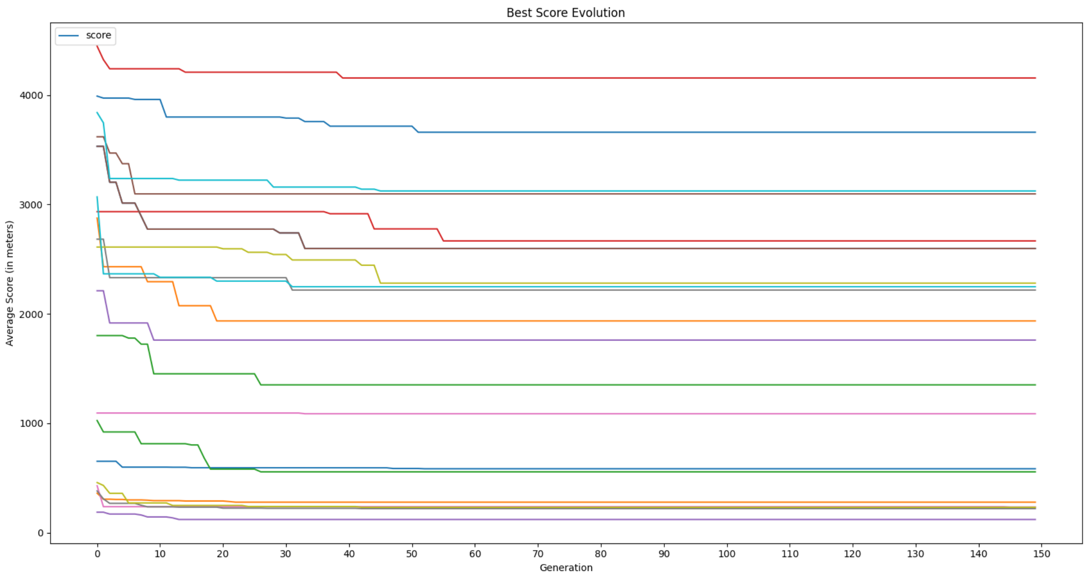


Figure 6.3: Score Evolution from 150 Generations

6.3.2 Network Training History

The saved models training histories need to be analysed with each respective metrics. MAE, MAPE and Loss function. Some examples follow:

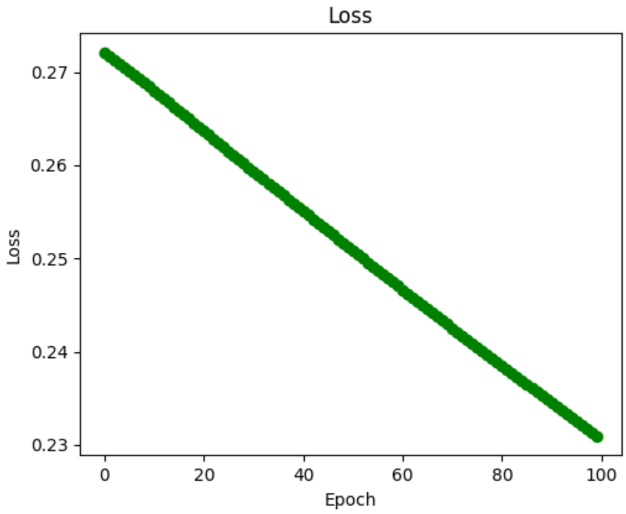


Figure 6.4: Loss function from 1_10_2023_21h20

The above model was trained with 100 values from 367321740 15 population 300 generations. The loss function result was faulty and so it was discarded since there was overfitting and contamination.

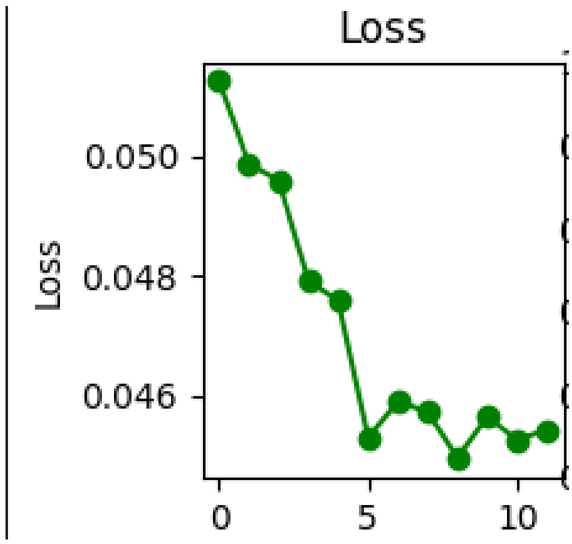


Figure 6.5: Loss function from 1_10_2023_13h00

The above model was trained with 2000 values from 367321740 15 population, 50 generations. The problem here was the early stopping, which was defined as 3, since it was too low, it was decided to increase it to 7 in later trainings.

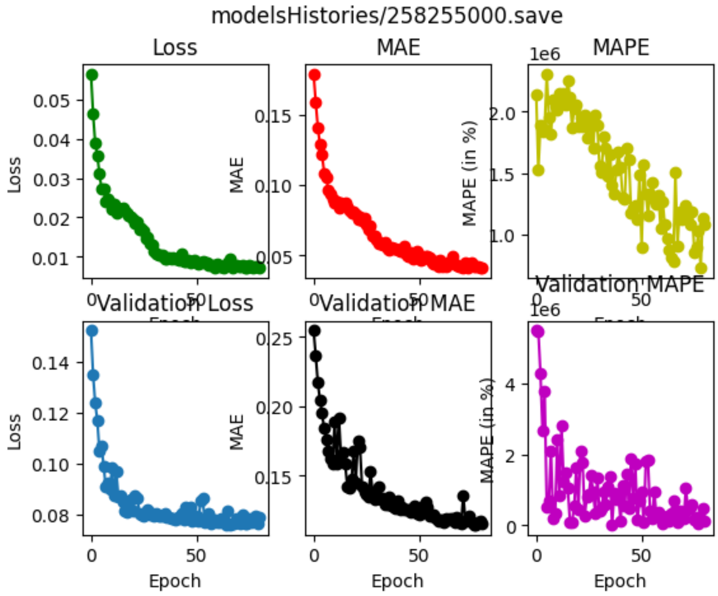


Figure 6.6: Loss function from 1_10_2023_21h20

The above model was trained with 100 values from 367321740 15 population, 300 generations. The same problem applied as previously, so the patience of the early stopping was increased to 10

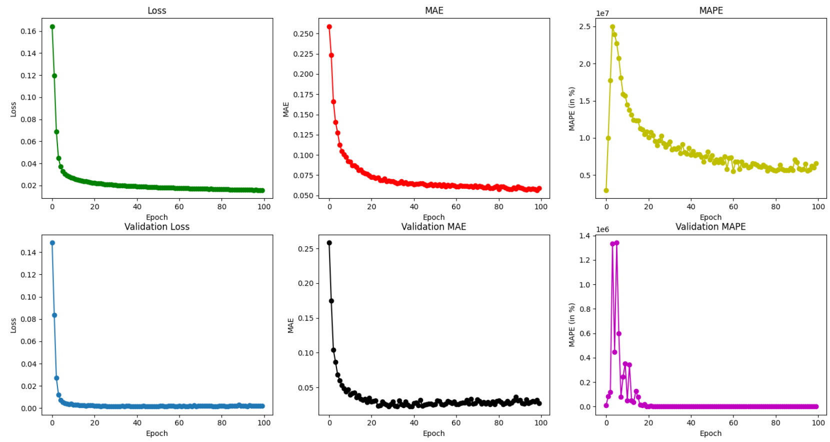


Figure 6.7: Loss function from 258255000

The above model was trained with 100 values from 258255000 15 population, 50 generations. The results from this model

Following the results, 100 epochs were used, in combination with a batch size of 1, accross 50 generations with a population of 15 and an early stopping with patience of 10, since lower than that wasn't enabling the model to train enough epochs,

6.3.3 Prediction Results

To test the predictions, we started by using one of the first models (named 30_09_2023, see Appendix E), where a MAE of 0,00007 from the predictions results was drawn, as well as a MAPE of 0,02%. It was clearly identified that the results were contaminated, as the scaleres being used were not the proper scalers, so this was fixed, and the later predictions started to be more reliable.

After this error was fixed, and having the previous errors also being fixed from what was talked previously, several prediction tests were conducted to draw MAE, and MAPE values from different datasets. After 30 different dataset prediction tests were conducted, we got a final result of an average MAE of 0,012, between the predicted latitudes and longitudes, and an average MAPE of 3,6% (See Appendix I). These values proved to be below the set 8% threshold to be considered good enough. An MAE of 0,012 across the positions averages a distance of 1330 meters, which is around 0,718142549 nautical miles. This proves a good prediction result.

6.3.4 Results Visualization on a Map

To complement the previously obtained results, the positions were put in a map so that better visualization could be taken (See Appendix H), some map examples follow where the blue line represents the true results, and the red line, the predicted results:

Figure 6.8: Map with vessel 566138000 dataset predictions

The above model used was trained with 100 values from the dataset, wit a population of 15, throughout 50 generations.

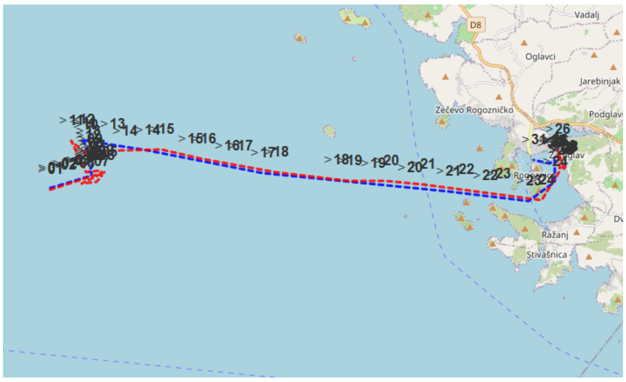


Figure 6.9: Map with vessel 138955240 dataset predictions

The above model used was trained with 500 values from the dataset, wit a population of 15, throughout 150 generations.

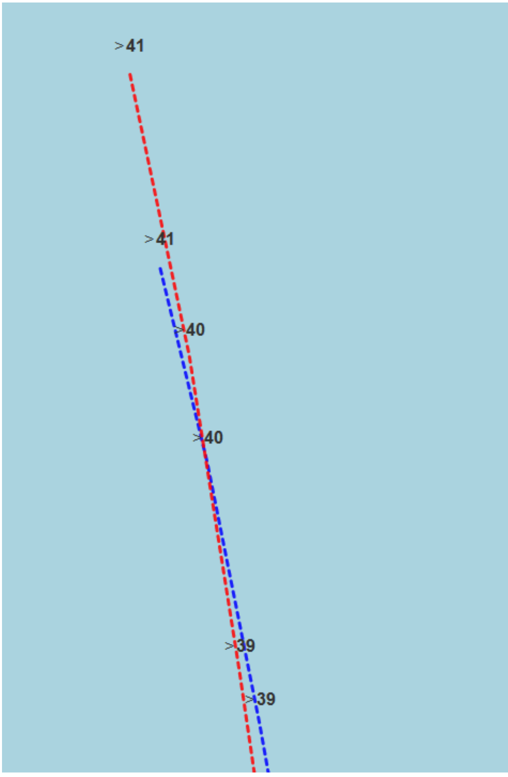


Figure 6.10: Map with vessel 672247000 dataset predictions

The above model used was trained with 1000 values from the dataset, wit a population of 15, throughout 75 generations.

Chapter 7

Conclusions, Challenges and Future Work

The development of this project was structured in a way that it can be divided in several different phases. The first phase followed the initial information collection regarding the technologies used and the area of development of this project.

Writing the Introduction and State of the Art provided insightful information regarding the current state of development of machine learning as well as the maritime field of study. This step was then followed by the development of the Value Analysis chapter, which viewed the gathering of information regarding what needed to be done to achieve the stakeholder's objectives. This facilitated the posterior development of the QEF, which allowed for a good visualization and tracking of what needed to be done achieve the previously defined objectives in the introduction as well as what needed to be done to try and solve the problem, also defined at this stage (See Chapter 1)

Nearly all objectives defined in the introduction chapter were achieved. A value analysis was done based on the customer's expectations and needs, which led to an investigation of the OVERSEE project, as well as gathering the requirements in conjunction with the customer.

To develop the solution it was also necessary to identify the extant anomalous behaviors so they can later be identified in the application, that was developed with a microservice approach in mind. With this, we moved to the network development phase, which led to the identification and analysis of the most adequate approaches. This allowed for the creation of a genetic algorithm to find the optimal network architecture, by training it with the gathered data.

The trained model then was able to predict the following positions, and report anomalous behaviors based on the comparison of the distances between the predicted position and the true positions. This approach stemmed from problems with the dataset, analysed in Sub Chapter 7.1, where no labeled data existed in order to create a good classification network.

Integration with OVERSEE was made possible with Kafka installed on a virtual machine, while deploying it in a Docker image presented its difficulties. This allows the solution to report anomalous behaviors through this message broker, so it could later receive feedback on these reports.

Reinforcement learning was supposed to be the final step but was postponed to future work due to time constraints. This step intended to improve the model over time, by providing it with user feedback on the warnings reports. This step would be hard to implement as the classification network development was not possible due to challenges in the data gathering.

The requirements specified in the QEF were almost all fulfilled, except a few as seen in Chapter 6.2.2. The requirements that were not fulfilled include the ones related to the constraints with the data, the Docker image, as well as the reinforcement learning, which was left for last due to the data problem, but ended up being postponed anyway due to time constraints.

To conclude, data proved to be one of, if not the most important factor in developing a good, quality neural network model, which crippled the development of this project, but we still overcame some difficulties with alternatives. Dividing the data in different groups to avoid overfitting and contamination of the model also proved to be a very important step as these were severely affecting the obtained results.

7.1 Faced Challenges and Constraints

As mentioned in Chapter 5.7, one of the biggest challenges of this whole operation was the lack of good labeled data. Due to the project presenting such a specific objective, it was hard to define what really would be considered an anomalous behavior, and what wouldn't. Stemming from this very problem, it was also very difficult to define thresholds, such as the threshold to what would be considered an anomalous distance between two geographical locations.

Another issue was the computing power necessary to produce good, viable models, as the amount of data was large, hence requiring some computing power.

On a more personal level, one of the biggest struggles was familiarizing with some of the libraries used, such as TensorFlow and Keras, which presented some challenges along the way. A significant amount of time was dedicated to understanding how these frameworks can be operated, and what good practices have to be taken, both in the data handling and preprocessing phases, as well as the other relevant phases, such as model training, and results gathering. On a more initial phase there was being undetected data contamination, which was compromising the results, leading to a time loss during that time frame.

Deploying the solution to a Docker image was another challenge faced due to constraints and challenges during the development phase. Firstly, this step was left as low priority, as it was not the most important phase of the project, which left little time for its implementation. This small time frame left for the development phase of this objective left no available support for the development of this step, which resulted in postponing this objective for future work.

Since the solution provides feedback when it detects a behavior that can be considered anomalous, there was the wish to provide feedback to the model prediction effort as to retrain it co-ordinance with the feedback received, i.e, if it was a false positive or not. This reinforcement learning objective was postponed since there wasn't any time left to develop this step and it is now referenced as future work at the time of the publication of this project.

Overcoming these challenges was difficult, but also gratifying, as it allowed for a more insightful understanding of the backbones of how neural networks function, as well as how microservices can be implemented and how beneficial that development approach is to a project as a whole.

7.2 Future Work

To further conclude unfinished objectives and mitigate faced challenges, the proposed improvements as future work stand:

- **Better data gathering on an initial phase** would allow for both a much more fluid solution specification and implementation phases. The lack of some key data, such as labeled anomalous behaviors datasets, compromised some of the objectives such as allowing for a different, more robust anomalous behavior detection using a dedicated classification neural network, which would potentially improve the results significantly. This approach needs a specialist both in machine learning, as well as in the maritime and OVERSEE project development to conduct a data labeling effort, in order to allow for a classification network to be implemented as part of the implementation of the solution.
- **Rethinking and reworking how anomalous behaviors are detected** are directly related to the point mentioned above. Without good labeled data, there is no possibility for a good classification network development, hence, compromising the implementation of this type of solution in mind.
- **More computing power** would have accelerated and improved the models' training phases, which would allow for a more thorough results gathering phase.
- **Docker deployment** would have allowed a more seamless integration with the OVERSEE project. Although the solution is able to run in a virtual machine hosted by the stakeholders, it would have been optimal to run it inside a Docker image so a more maintainable solution would allow for easier future changes.

Given these suggestions, and executing them properly, will ensure a better project development lifecycle, and a more robust and accurate solution on future iterations.

Bibliography

(2023). en. url: https://pypi.org/project/pyais/ (visited on 02/16/2023).

Bang, Jiwon et al. (Apr. 2018). "Design and implementation of a load shedding engine for solving starvation problems in Apache Kafka". In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–4. doi: 10.1109/NOMS.2018.8406306.

Brownlee, Jason (Dec. 2017). *A Gentle Introduction to Transfer Learning for Deep Learning*. en-US. url: https://machinelearningmastery.com/transfer-learning-for-deep-learning/.

Cohen, P.R. (1991). "A Survey of the Eighth National Conference on Artificial Intelligence: Pulling Together or Pulling Apart?" In: *AI Magazine*, pp. 16–41.

Corucci, L. et al. (Oct. 2012). "An integrated space-based passive system for maritime operational traffic surveillance". In: *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*, pp. 1–4. doi: 10.1109/ESTEL.2012.6400080.

Cox, John (2021). "The Higher Education Environment Driving Academic Library Strategy: A political, economic, social and technological (PEST) analysis". In: *The Journal of Academic Librarianship* 47.1. doi: 10.1016/j.acalib.2020.102219.

Crawford, Merle and Anthony Di Benedetto (2003). *New Products Management*. 7th ed. McGraw-Hill. isbn: 978-0072471632.

CSW (May 2020). "A Sea-Change for Maritime Agencies". In: url: https://criticalsoftware.com/multimedia/critical/en/IJL2JryVD-CSW_-_Case_Study_-_Oversee_-_A_Sea-Change_in_UxD.pdf.

District, MaRS Discovery (2012). "Crafting Your Value Proposition". In: *Fundamentals of Entrepreneurial Management*, pp. 2–8.

Dixit, Utkarsh, Suman Bhatia, and Pramod Bhatia (May 2022). "Comparison of Different Machine Learning Algorithms Based on Intrusion Detection System". In: *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*. Vol. 1, pp. 667–672. doi: 10.1109/COM-IT-CON54601.2022.9850515.

Dowling, Michael and Brian Lucey (Jan. 2023). "Chatgpt for (Finance) Research: The Bananarama conjecture". In: *Finance Research Letters*, p. 6. doi: 10.1016/j.frl.2023.103662.

Duffuaa, Salih (Nov. 2009). "TOPSIS Method". In: *Apontamentos das Aulas do Módulo Análise de Valor: Multi-Criteria Decision Making*, pp. 1–21.

Escudeiro, Paula and José Bidarra (Jan. 2008). "Quantitative Evaluation Framework (QEF)". In: *Conselho Editorial/Consejo Editorial*, p. 16. url: https://www.researchgate.net/publication/257579051_Quantitative_Evaluation_Framework_QEF.

Farahnakian, Farshad, Jukka Heikkonen, and Paavo Nevalainen (July 2022). "Abnormal Behaviour Detection by Using Machine Learning-Based Approaches in the Marine Environment: A Literature Survey". In: *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–11. doi: 10.1109/ICECET55527.2022.9872905.

Ferenti, Tamás (2017). "Biomedical applications of time series analysis". In: *2017 IEEE 30th Neumann Colloquium (NC)*, pp. 000083–000084. doi: 10.1109/NC.2017.8263256.

Glegg, G., R. Jefferson, and S. Fletcher (Mar. 2015). "Marine governance in the English Channel (la manche): Linking science and Management". In: *Marine Pollution Bulletin* 95.2, pp. 707–718. doi: 10.1016/j.marpolbul.2015.02.020.

Gomes, Elsa F. (Dec. 2016). "Experimentação e Avaliação". In.

Hajizadeh, Yasin (2019). "Machine learning in oil and gas; a SWOT analysis approach". In: *Journal of Petroleum Science and Engineering* 176, pp. 661–663. doi: 10.1016/j.petrol.2019.01.113.

al-Hddli, Hmidt, Ahmed Ghiduk, and Mohammed El-Telbany (June 2010). "The limitations of genetic algorithms in software testing". In: pp. 1–7. doi: 10.1109/AICCSA.2010.5586984.

Hersztowski, Paul (Mar. 2020). en. url: https://uigstudio.com/insights/what-is-a-business-model-canvas (visited on 01/28/2023).

IMO (July 2018). *Solas chapter V, Regulation 19 - Navigation Center*. url: https://www.navcen.uscg.gov/sites/default/files/pdf/AIS/SOLAS.V.19.2.1-5.pdf (visited on 01/16/2023).

– (2019). *AIS transponders*. url: https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx (visited on 01/16/2023).

International, B2B (2023). *What is the Value Proposition Canvas?* en. url: https://www.b2binternational.com/research/methods/faq/what-is-the-value-proposition-canvas/ (visited on 01/28/2023).

Janashia, Nana (2022). *Microservices explained - the What, Why and How?* url: https://www.youtube.com/watch?v=rv4LlmLmVWk (visited on 01/20/2023).

Koen, Peter (May 2004). "Understanding the front end: A common language and structured picture". In: *Understanding the Front End: A Common Language and Structured Picture*, pp. 3–27. url: https://kip.zcu.cz/kursy/imi/ERASMUS/KoenMay26UnderstandFrontEndPDMA_01.pdf.

Koen, Peter et al. (2001). "Providing clarity and a common language to the "Fuzzy front end"". In: *Research-Technology Management* 44.2, pp. 46–55. doi: 10.1080/08956308.2001.11671418.

Kumar, Ajitesh (2023). en. url: https://vitalflux.com/different-types-of-time-series-forecasting-models/ (visited on 03/31/2023).

Lindič, Jaka and Carlos Marques da Silva (2011). "Value proposition as a catalyst for a customer focused innovation". In: *Management Decision* 49.10, pp. 1694–1708. doi: 10.1108/00251741111183834.

MarineTraffic (2023). url: https://www.marinetraffic.com/ (visited on 09/01/2023).

MovableTypeLtd (2023). en. url: https://www.movable-type.co.uk/scripts/latlong.html (visited on 10/01/2023).

Nicola, Susana (Nov. 2018). "Método de Análise Hierárquica". In: *Apontamentos das Aulas do Módulo Análise de Valor*, pp. 9–48.

– (Nov. 2020a). "Análise de Valor". In: *Apontamentos das Aulas do Módulo Análise de Valor*, pp. 4–27.

– (Nov. 2020b). "FAST and QFD Techniques". In: *Apontamentos das Aulas do Módulo Análise de Valor*, pp. 2–25.

Nyuytiymbiy, Kizito (Mar. 2022). *Parameters and Hyperparameters in Machine Learning and Deep Learning*. en. url: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac.

OECD (2023). *Ocean shipping and shipbuilding - OECD*. en. url: https://www.oecd.org/ocean/topics/ocean-shipping/ (visited on 01/16/2023).

Perera, Lokukaluge, Paulo Oliveira, and C. Guedes Soares (Sept. 2012). "Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 13.3, pp. 1188–1200. issn: 1558-0016. doi: 10.1109/TITS.2012.2187282.

Petrovic, Otto and Christian Kittl (Oct. 2003). "Capturing the value proposition of a product or service". In: p. 2.

Placek, Martin (May 2022). *Topic: Container shipping*. url: https://www.statista.com/topics/1367/container-shipping/ (visited on 01/17/2023).

Porter, Michael E. (1985). "Competitive Strategy: The Core Concepts". In: *Competitive advantage*. Free, pp. 26–28. isbn: 0-02-925090-0.

Rathod, Viraj, Chandresh Parekh, and Dharati Dholariya (Sept. 2021). "AI & ML Based Anamoly Detection and Response Using Ember Dataset". In: *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 1–5. doi: 10.1109/ICRITO51393.2021.9596451.

Recommendation, ITURM (2014). "1371-5: Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band". In: *International Telecommunication Union Radiocommunication Sector (ITU-R): Geneva, Switzerland*.

Roberts, Paul (Jan. 2007). "Quality Function Deployment". In: *Product Excellence Using Six Sigma*, pp. 6–67.

Samia, Bougareche, Zehani Soraya, and Mimi Malika (May 2022). "Fashion Images Classification using Machine Learning, Deep Learning and Transfer Learning Models". In: *2022 7th International Conference on Image and Signal Processing and their Applications (ISPA)*, pp. 1–5. doi: 10.1109/ISPA54004.2022.9786364.

Sciancalepore, Savio et al. (July 2022). "Auth-AIS: Secure, Flexible, and Backward-Compatible Authentication of Vessels AIS Broadcasts". In: *IEEE Transactions on Dependable and Secure Computing* 19.4, pp. 2709–2726. issn: 1941-0018. doi: 10.1109/TDSC.2021.3069428.

Scotti, Filippo et al. (Oct. 2015). "Photonics-based dual-band radar demonstration for maritime traffic detection". In: *2015 International Topical Meeting on Microwave Photonics (MWP)*, pp. 1–4. doi: 10.1109/MWP.2015.7356729.

Serafino, Giovanni et al. (Mar. 2022). "Experimental Dual-band Coherent Photonics-based Radar Network with ISAR Imaging". In: *2022 IEEE Radar Conference (RadarConf22)*, pp. 1–6. doi: 10.1109/RadarConf2248738.2022.9763913.

Sharma et al. (Oct. 2012). "Achieving Success through Value Engineering: A Case Study". en. In: p. 1.

Shree, Rishika et al. (Aug. 2017). "KAFKA: The modern platform for data management and analysis in big data domain". In: *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, pp. 1–5. doi: 10.1109/TEL-NET.2017.8343593.

TIBCO (Jan. 2016). *What is a Neural Network?* url: https://www.tibco.com/reference-center/what-is-a-neural-network (visited on 02/02/2023).

Toloue, Kamran Fartash and Majid Vafaei Jahan (Feb. 2018). "Anomalous behavior detection of marine vessels based on Hidden Markov Model". In: *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pp. 10–12. doi: 10.1109/CFIS.2018.8336611.

USCG (2023). url: https://www.navcen.uscg.gov/types-of-ais (visited on 02/18/2023).

Violos, John et al. (Sept. 2020). "Next Position Prediction using LSTM Neural Networks". In: pp. 232–240. doi: 10.1145/3411408.3411426.

Vyas, Shubham et al. (Feb. 2022). "Performance Evaluation of Apache Kafka – A Modern Platform for Real Time Data Streaming". In: *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*. Vol. 2, pp. 465–470. doi: 10.1109/ICIPTM54933.2022.9754154.

WindwardAI (2023). url: https://https://windward.ai/ (visited on 09/01/2023).

Wolff, E. (2016). *Microservices: Flexible Software Architecture*. isbn: 978-0-134-60241-7.

Wolsing, Konrad et al. (Jan. 2022). "Anomaly Detection in Maritime AIS Tracks: A Review of Recent Approaches". In: *Journal of Marine Science and Engineering* 10.1, p. 112. issn: 2077-1312. doi: 10.3390/jmse10010112. url: http://dx.doi.org/10.3390/jmse10010112.

Woodall, Tony (2003). "An Attributional, Structural and Dispositional Analysis". In: *Conceptualising 'Value for the Customer'* 2003.12, p. 1. url: https://www.ams-web.org/page/OriginalAMSR/articles/woodall12-2003.pdf.

Xu, Xin et al. (2020). "LSTM-GAN-XGBOOST Based Anomaly Detection Algorithm for Time Series Data". In: *2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)*, pp. 334–339. doi: 10.1109/PHM-Jinan48558.2020.00066.

Appendix A

QEF

The fulfilled Quantitative Evaluation Framework can be accessed through:

https://docs.google.com/spreadsheets/d/1BvBGPnGBSQ3QQJD5ZIGzOUztRrrfn4SV
/edit?usp=drive_link&ouid=104102380370714844763&rtpof=true&sd=true

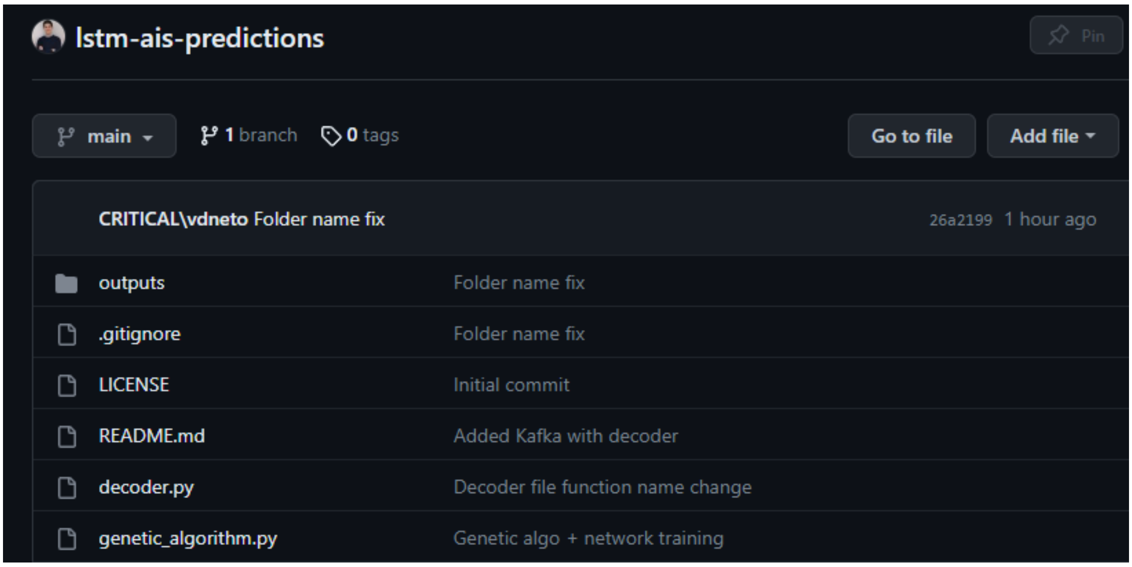
q	D	Qi	Dimension	Qj	Wij (Factor Weight j in	Factor	rujk (requirement weight k in Factor j) (2, 4, 6, 8, 10)	Requirement	wfk % requirement fulfillment k) [0,100]
93%	0,19	97,18	Functionality	100,00	0,17	Data	2,00	FD01 - The AIS messages shall be stored and provided in text files organized by date	100
							4,00	FD02 - Each vessel's information shall be stored in individual files	100
							8,00	FD03 - There shall be a big enough dataset containing vessel information	100
							4,00	FD04 - The dataset shall consist of AIS messages from vessels reports from a given consistent timespan	100
							10,00	FD05 - The AIS messages shall contain the relevant information for the proper ends (training the network)	100
				100,00	0,17	Preprocessing	10,00	FP01 - AIS decoder shall read the text files storing the provided data and decode them into human readable vessel br	100
							10,00	FP02 - The decoded messages shall provide the necessary vessel information	100
							8,00	FP03 - The decoded messages positions shall be received in Latitude/Longitude	100
							2,00	FP04 - The decoded messages shall be stored as comma separated values (CSV) files	100
							8,00	FP05 - The decoded messages' Latitude/Longitude positions shall be converted to Distance/Bearing	100
				100,00	0,24	Network	10,00	FN01 - The solution shall produce a keras model with the trained network	100
							4,00	FN02 - The algorithm shall use LSTM as its central paradigm	100
							2,00	FN03 - The solution shall be developed as an ANN using TensorFlow and Keras	100
							6,00	FN04 - The solution must distinguish an anomalous behavior from a behavior considered normal (Position must not be higher than the average of the predicted positions times 2,5 and must not be higher than 3 nautical miles)	100
							10,00	FN05 - The solution Shall use the produced model to predict the following position given the current position	100
							6,00	FN06 - The model shall use an algorithm to optimize its accuracy	100
							6,00	FN07 - The model shall evolve as it receives new positions	100
				100,00	0,21	Results	6,00	FR01 - The obtained results shall be plotted for better visualization	100
							2,00	FR02 - The obtained results must be reliable	100
							2,00	FR03 - The obtained results must be accurate	100
							10,00	FR04 - Metrics shall be taken from the results analysis to evaluate the solution's results quality	100
							10,00	FR05 - The obtained predictions shall be converted back to Latitude/Longitude from Distance/Bearing	100
							2,00	FR06 - The obtained predictions shall be compared with the actual positions on a plotted map for better visualization	100
				86,36	0,21	Integration and Communication	10,00	FI01 - The solution must be developed as a microservice	100
							4,00	FI02 - Kafka shall receive positions from OVERSEE in real time	75
							10,00	FI03 - The solution shall receive positions from Kafka topics	100
							4,00	FI04 - The solution shall be deployed using a Docker image	50
							10,00	FI05 - When an anomalous behavior is detected, it shall notify the respective user	100
							6,00	FI06 - Kafka shall perform its communication with the solution from within the Docker image installed in the virtual m	50
		87,48	Quality and Reliability	90,91	0,45	Speed and Optimization	4,00	QS01 - The received positions from Kafka shall be handled in real time	100
							4,00	QS02 - The service shall be able to run continuously	100
							8,00	QS03 - The triggered predictions and models training shall be done in an amount of time that makes sense in the relev	75
							4,00	QS04 - The service shall be able to listen to OVERSEE on a continuous basis	100
							2,00	QS05 - The network shall be optimized to an extent that makes sense relative to the available computing power	100
				100,00	0,18	Results Quality	8,00	QRQ01 - The solution shall be optimized to reduce the amount of false positives and false negatives	100
							10,00	QRQ02 - The dataset shall be divided in three different divisions, one for training, one for testing, and one for validati	100
				76,92	0,36	Reliability	2,00	QR01 - The service shall have a good performance based on the available resources and platforms	100
							10,00	QR02 - The solution shall be able to handle large amounts of data	100
							6,00	QR03 - The model shall evolve as it receives feedback from the user	0
							8,00	QR04 - There shall be a model for each vessel	100
		85,71	ecurity and Maintenanc	100,00	0,71	Maintenance	2,00	SM01 - The code shall follow the respective guidelines and be commented and structured well enough for its understa	100
							6,00	SM02 - The solution's documentation shall be properly given	100
							8,00	SM03 - The solution shall avoid runtime errors	100
							2,00	SM04 - The solution shall be developed following good programming practices	100
							2,00	SM05 - The deployment of the solution shall allow an easy maintenance of it	100
				50,00	0,29	Security	2,00	SS01 - The Docker image shall be deployed in a secured CSW's virtual machine	100
							2,00	SS02 - Kafka shall be installed in the respective secured virtual docker Image	0

Appendix B

Project Repository

The project repository can be accessed from:

https://github.com/VitorCoelhoNeto/lstm-ais-predictions



Appendix C

Dataset Used

Note: the dataset was too large (several gigabytes) to append, so a random selection of vessel data was selected, including the ones used for training experiments, to annex to this work.

These datasets can be accessed through:

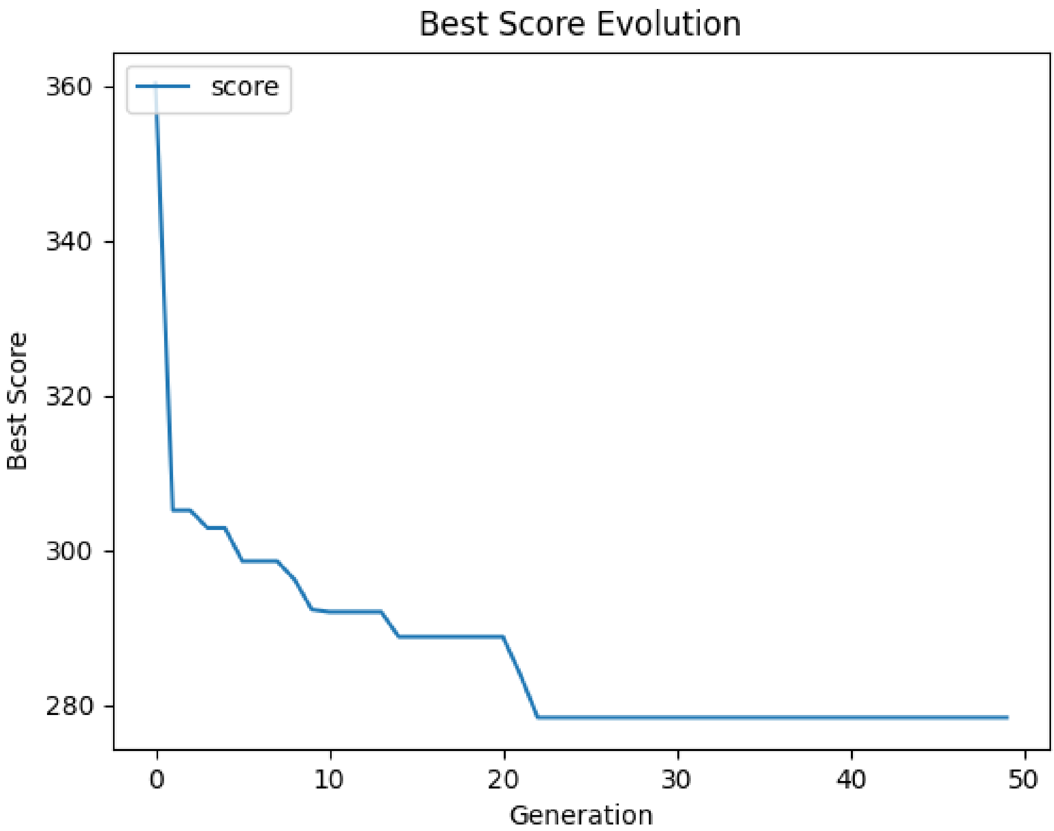
`https://drive.google.com/file/d/1rGDNHWg3shRoXd25aEA3l2G2S9XEHlxK/view?usp=drive_link`

Appendix D

Genetic Algorithm Results

Genetic algorithm's evolution scores can be accessed through:

`https://drive.google.com/file/d/1Tvsz78x2OkXCcyOlkxok-1qcBeV5zGP5/view?usp=drive_link`



Appendix E

Trained Models

Trained models accompanied by their respective information file. These files can be directly be traced back to their respective scalers via the file name.

https://drive.google.com/file/d/1yoVLebBTssgDZb6GrgmqUko8KBKEdtDB/view?usp=drive_link

Appendix F

Trained Models' Scalers

Trained models' scalers accompanied by their respective information file. These files can be directly be traced back to their respective trained models via the file name.

https://drive.google.com/file/d/13SAvFfBvD867_oUoxYgtoay_iMNfqLu6/view?usp=drive_link

Appendix G

Trained Models' Fit Metrics

Trained models' fit metrics accompanied by their respective information file. These files can be directly traced back to their respective trained models via the file name.

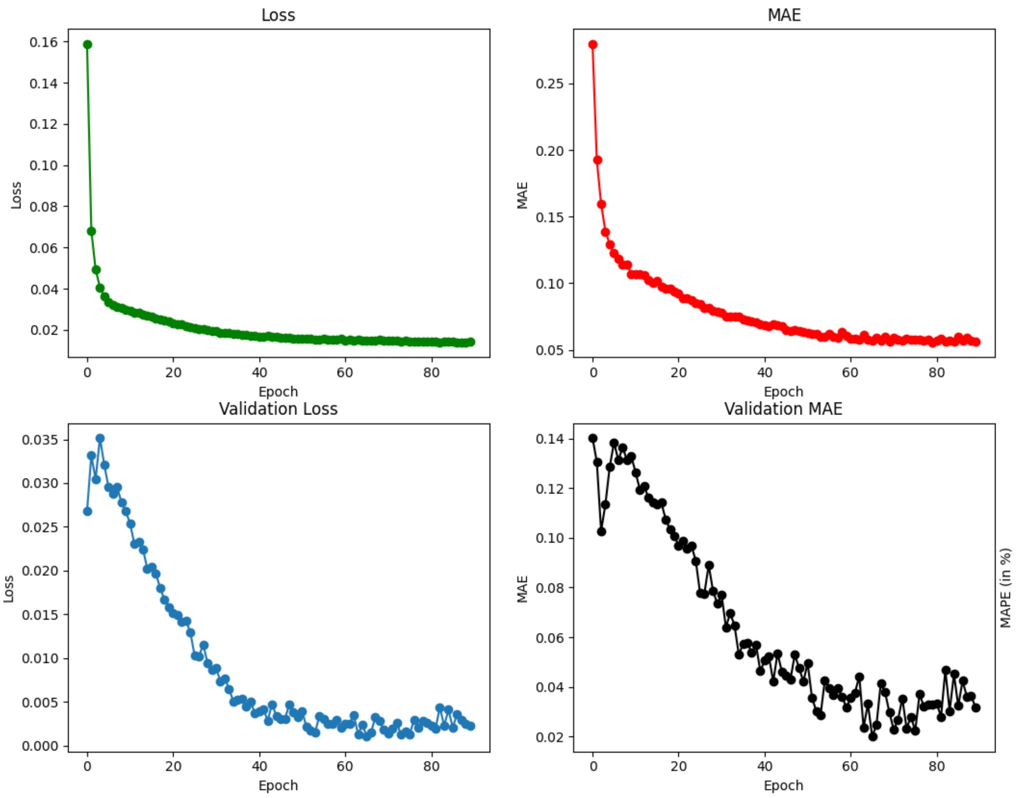
https://drive.google.com/file/d/17okzinwsiSC6nmoe_fl_n9XlrqFtJdZn/view?usp=drive_link

Included are also the figures in png format with the aggreggated graphs when relevant

https://drive.google.com/file/d/1AbJZRyQi5vGcYOOWccT7Ce85Yp-FKJ1j/view?usp=drive_link

Other miscellaneous test runs:

https://drive.google.com/file/d/1xnNvfrlnk8_ouInvRMoxdxzTfqEGBCoy/view?usp=drive_link



Appendix H

Map Predictions

Map predictions accompanied by their respective information file. These files can be directly traced back to their respective trained models via the information file. The true positions are represented by the blue routes, and the predicted positions by red routes.

https://drive.google.com/file/d/1FJC1tJrKcWAxbeAs7ARYSvYzA449SvF4/view?usp=drive_link

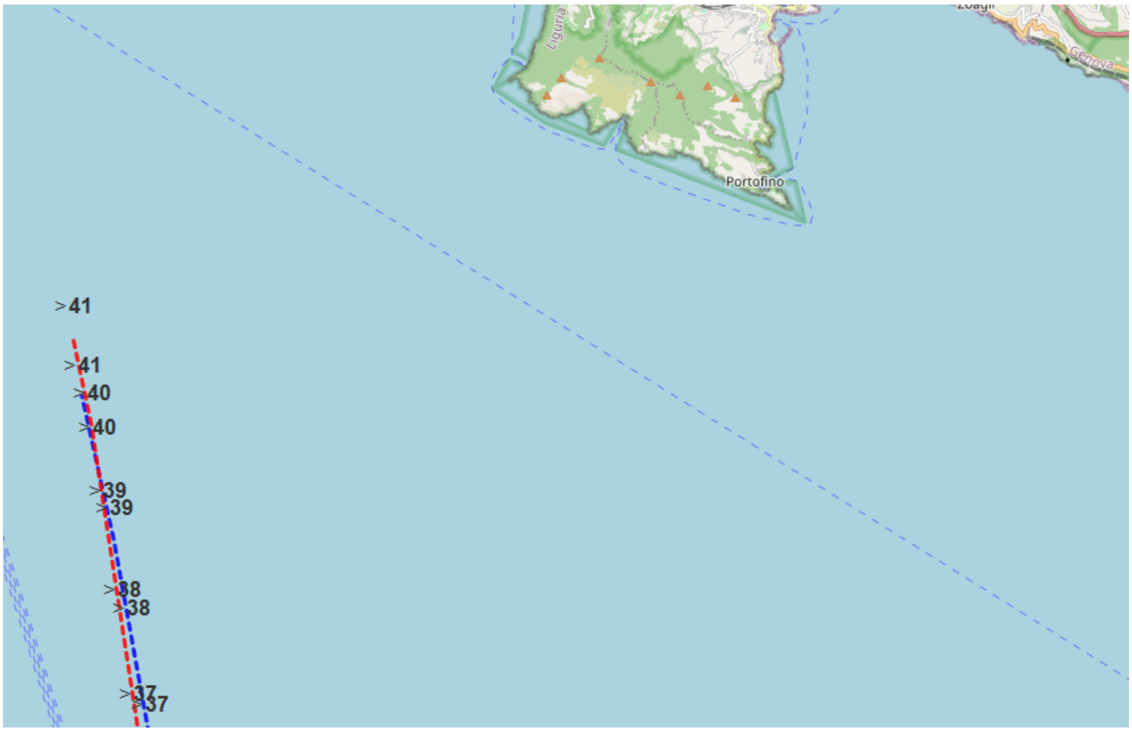


Figure H.1: Enter Caption

Appendix I

Predictions Results

Prediction results obtained from presenting the trained models with new information for predicting the subsequent positions.

https://drive.google.com/file/d/1rZT3ysWJLSH8NOgOk3tvzk8wCHTam-C6/view?usp=drive_link