FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

OVERSEE: Identification Of Anomalous Vessel Behaviour

José Pedro Vieira Gomes

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Aníbal Matos

Second Supervisor: Paulo Gomes

August 9, 2018

Resumo

Com um crescente número de embarcações equipadas com um Sistema Automático de Identificação cada vez mais dados estão a ser gerados e a criar a oportunidade para novos estudos sobre o comportamento de embarcações marítimas.

Para resolver o problema de assegurar e proteger vastas quantidades de zonas marítimas seria interessante a ajuda de um sistema automático de deteção de navios com comportamentos anormais. Foi com esse objetivo em mente que ao longo do desenvolvimento deste projeto se tirou partido das capacidades de teoria relacionada com Memória Temporal Hierárquica (*Hierarchical Temporal Memory - HTM*) e respetivos algoritmos para identificar comportamentos anômalos em trajetórias de navios de forma a melhorar as capacidades de monitorização e a possibilidade de avisos mais oportunos e consequente plano de ação desde simples contato até à identificação e posterior missão de captura ou salvamento.

Ao longo desta dissertação um sistema baseado em *HTM* foi desenvolvido e aplicado à tarefa acima com bons resultados na modelação das trajetórias de embarcações ao longo da costa Portuguesa e na identificação de um subgrupo de trajetórias com comportamentos anômalos previamente identificados.

Abstract

With an increasingly number of ships equipped with an Automatic Identification System (AIS) more and more data is being generated and creating an opportunity for new studies of the maritime vessel behaviours.

The problem of securing and protecting vast expanses of the maritime zone could use the help of an automatic vessel anomalous behaviour system. With that goal in mind the development of this project took the capabilities of Hierarchical Temporal Memory (HTM) theory and respective algorithms to help identify anomalous behaviours on vessel trajectories improving sea monitoring capabilities and the possibility of more opportune warnings and subsequent action plans from simple contact or vessel identification to arrest or rescue missions.

Along this dissertation an HTM based system was developed and applied to the task above with good results on the establishment of a model of vessel trajectories on the Portuguese maritime zone and good performance on the identification of a subset of trajectories with anomalous behaviours previously identified.

Acknowledgements

I would like to thank Numenta Inc. which open sourced much of their work related with HTM with careful attention to the internet community which accompanied their work over the years and to which various materials were made available such as all kinds of videos, papers and implementations which were very important to the development of this dissertation.

Thanks to Aníbal Matos at FEUP for being available for any need throughout the process, specially on the preparation and project set-up.

Thanks to Paulo Gomes at Critical Software for pointing out the general direction to pursue all along.

And finally thanks to my friends and family which in the end of the day are the greatest support for a continuous arduous work.

José Pedro Gomes

Contents

List of Figures

List of Tables

Abbreviations and Symbols

AIS	Automatic Identification System
CBR	Case-Based Reasoning
SVM	Support Vector Machine
BN	Bayesian Network
VTMIS	Vessel Traffic Monitoring and Information Systems
MSA	Maritime Situational Awareness
VTS	Vessel Traffic Services
SOG	Speed Over Ground
COG	Course Over Ground
ROT	Rate Of Turn
KDD	Knowledge Discovery in Databases
MMSI	Maritime Mobile Satellite Identity
SDR	Sparse Distributed Representations
HTM	Hierarchical Temporal Memory
SP	Spatial Pooler
TM	Temporal Memory
GIS	Geographical Information System
QGIS	QuantumGIS
OSGeo	Open Source Geospatial Foundation

Chapter 1

Introduction

On our blue planet where a big part of the surface is covered by water, maritime transportation represents approximately 80 percent of the volume of global trade (Asaritoris *et al.*, 2013, [3]). This presents a big challenge on many fronts including efforts to maintain the security of all parts involved or environment concerns while the need to improve the control over all the traffic that is entailed in a sector with gigantic economy global repercussions is a continuous challenge.

1.1 Context

With the advance of technologies devices like the Automatic Identification System (AIS), a ship reporting system that brought great improvements to the Maritime Situational Awareness (MSA) and that was first developed for collision avoidance and which has lately transformed in the core of the efforts for better MSA. This device is now an international standard on communications between vessels or with terrestrial stations and brought an improvement on the general maritime security and control by helping vessels to avoid collisions and by assisting Vessel Traffic Services (VTS) on the control of vessels near the coast.

With the ever growing need for better methods of control of the coastline and the appearance of data sources like the AIS system there was a need to create a new integrated environment that could allow technicians which needed to have a easier access to the growing amounts of data for a better performance on tasks like identification of anomalous situations related to the enforcement of the law or environment protection and better access to the data which is critical on search and rescue missions. In partnership with the company Critical Software, the Portuguese navy, duty-bound to perform on those fields, developed a software called Oversee which integrates maritime information available into one ecosystem that presents this information to navy operators in real time in a way that allows them to make the best of the integrated data for faster response on all situations on the Portuguese coastline.

After the development of the software Oversee the result was that nowadays vessels are monitored by human technicians with the use of the new system and when there is a suspicious behaviour or emergency situation the operator in charge of monitoring starts the means needed to

understand the suspicious behaviour or to give the necessary support on the emergency response. Since this is executed manually it consumes a lot of human resources and doesn't guarantee that all the suspicious behaviours are detected which presents the possibility for improvement of the current system.

1.2 Motivation

With the ever growing number of data sources available is expectable that systems factoring this data for surveillance and providing useful inputs to users were more readily available than they actually are. Recently there's been a large investment on the development of systems that make use of the available data provided by AIS and radars to help on security of maritime space, and integration of Vessel Traffic Monitoring and Information Systems (VTMIS), such as the Oversee system, providing intelligence to this systems and making use of their already available information system architecture to improve on the information and automation capabilities provided to end users. This capabilities can range from simple detection and alarm on conditional trigger or even to the ability of learning with the user input and improving at each new detection, be it from the automatic system or user provided the goal is to achieve the best results today while in the future achieving a reliable piece of software able to aid operators accomplish their duty.

So with the idea that "being able to do it automatically would be much more efficient and less error prone", a new phase that extends the current Oversee system was started. For this using Machine Learning and the ability to identify these behaviours and learn from it is one of the paths that is going to be explored. This phase is already midway where related work was previously developed.

The first objective lies on understanding the different behaviours that would be of interest to understand and latter identify as a normal or anomalous vessel behaviour. There are several anomalous behaviours class possibilities identifiable while performing analysis of the AIS transmissions data which is one of the most important maritime reporting systems and from where very large amounts of data are available, examples of these behaviours classes are: deviation from standard routes, unexpected AIS activity, unexpected port arrival, close approach and zone entry (Lane *et al.*, 2010, [15]).

The second is pointed by the need of methods on identification of the behaviours formerly identified and which make the end solution backbone. Focusing on machine learning approaches were identified several techniques such as pattern classification techniques or application of Gaussian Processes for normality model creation. It's now important to expand the range of possibilities and better understand the different pros and cons of the techniques such as ability to adapt between using new data sets and using knowledge provided by system supervisors or experts on the preparation phase or along the system life, it's also important to contemplate data needs and performance which should relate with the different kinds of behaviour being identified.

The related work previously developed entails the understanding and preprocessing of the data available which culminated on using data from the AIS system with the main focus on the

use of GPS positions, source and destination points, Speed Over Ground (SOG) and Course Over Ground (COG). Using the AIS messages data and after preprocessing them in a way that makes the amount of data being used more manageable the first prototype was developed, an algorithm able to classify the course points in start and end point and way-points which refer to any point between the start and end point. A group of this points can create a track which, at that point of development, presented as an important unit for the identification of anomalous behaviour. The most recent work was done using case-based reasoning (CBR) and uses a track as case-unit, similar tracks are used as comparative cases to establish the normal behaviour which, on the other hand, allows the identification of anomalous tracks which present behaviours that deviate from the norm for tracks with similar characteristics.

1.3 Problem Definition

As prior explained above this project comes to cover the needs of an automated system capable of detecting anomalous behaviours of maritime vessels on the Portuguese coastline improving the response time and reliability on the identification and necessary actions to handle the diverse situations that happen daily and that present a threat to the security of the country or simply to individuals in distress situations on Portuguese maritime space. This project comes as an expansion of the Oversee project being the "automatic" keyword to the development on this phase. Since this project is ongoing it's important to make clear some of the steps already done even if on a tentative way:

1. AIS messages preprocessing into more compact data sets without reducing relevant data and down sample of the data into a meaningful rate for better performance;

2. Establishment of three main classes:

    - Vessel: represents one vessel identification and contains all information on the current state like last location transmitted and navigation data, state of the vessel - Sailing, Stationary or Lost - and, of course, static information like Maritime Mobile Satellite Identity (MMSI) or ship type which includes code numbers for fishing, passenger, sailing, tanker, diving or even military vessels.
    - Way-point or waypoint: this stands for every geographic point received along the course and maintains the relevant data like position, state of the vessel at the moment, transmission time and possible indirect data like a Calculated Speed Over Ground that depends on the distance and time from the last transmission to obtain an average speed comparable with the SOG received from the vessel and which brings new information. This class can have four states: Track Start, Track End, Stationary Point and Way-point.
    - Track: this represents a group of way-points and is a trajectory performed by a specific vessel assigned to the track.

3. Successful application of data mining algorithm based on case-based reasoning to identify anomalous tracks and respective vessels while using restricted data set with definition of distance metrics for comparative analysis between cases (database of tracks as models).

Knowing what's already developed defines the future track of the solution to be achieved, at this point we have a proof of concept that using case-based reasoning can perform well on a real time identification of anomalous behaviours, since this is the case it's expected for the near future a continuous work on expanding the initial project. Since this expansion comes from the problems present on the initially found solution it's important to identify them:

- There's still problems on the identification of some specific situations that should be abnormal and which are still not being identified meaning the metrics are still inappropriate.
- The solution is limited to a single class of vessels which makes the differentiation of normal/abnormal behaviour a lot simpler. For the same reason the metrics used for the current case-based reasoning won't support the ability to identify normalcy on vessels with very contrasting behaviours between classes, for example a fishing boat will need very different metrics since there's probably little relation between the way a track for a cargo vessel can be modelled in contrast with a fishing one, this is expected since different classes of ships have various goals in areas of activity generally unrelated. There's the need to separate global metrics and local ones which should only be applied to specific ship classes.
- The CBR implementation isn't capable of making good use of the natural waypoint temporal sequence on modelling trajectories from vessel tracks which are modelled more as waypoint array then as real movement trajectories.
- CBR resulting models are not apt at generalization due to being limited by implementation which makes use of what are basically spatially distributed statistics to model normal behaviour.
- There's yet no way to easily classify (ab)normal tracks manually allowing the case-base improvement and testing the performance of the algorithms learning ability from expert knowledge which would represent an important advantage for future growth.

1.4 Goals

The current approach using case-based reasoning has identified with success anomalous behaviours and presents as a viable solution for the problem of automatic identification of anomalous vessels behaviours. Despite that it's still heavily restricted and using only a reduced amount of data and being applied on a restricted number of vessel classes. It's able to identify some anomalies but still not presenting the actual reasoning for the anomaly, while the later could be achieved by using expert knowledge to define the anomalous cases found, it's still too early for this but it is by itself an important characteristic that makes this approach very interesting. Since the current Oversee

system does manual identification there's already experts able to do the classification of cases which should create a positive feedback cycle able to improve the case database and subsequently the results performance. The application of case-reasoning brings the need to define comparative metrics still being improved for a better performance on the identification of anomalous behaviour and the reduction of false-positives.

At first this project had as goals to improve and generalize previous work developed on the case-based reasoning solution. The big goal of the project was the creation of new metrics or improvement of the existent ones while abating the restrictions previously established such as the application of the same algorithm on different classes of vessels which would present new needs on the identification of new (ab)normal cases. This would present challenges on the current defined characteristics of normalcy which may have to be adapted per class or create the need to implement specific metrics per class, the introduction of new data until then deemed as not pertinent that would improve the case characterisation or other complementary algorithms application. At some point during preparation of the this project a new approach to solve the problem of anomaly identification was identified. This approach uses a new algorithm introduced initially by Critical Software advisor as just a new possibility for analysis namely HTM.

This algorithm or group of algorithms which build the Hierarchical Temporal Memory (HTM) theoretical framework and which after analysis were considered to have the necessary characteristics for the development of a new HTM based solution had as core feature the capacity of providing temporal memory. Previously as future possibilities on the CBR system improvements using temporal data was one of the appointed enhancements since cases were restricted by the implementation based more on a point basis then on a track basis since the sequence of way-points wasn't particularly important even if track related information was being used on metrics like distance to track start. Using HTM provides the ability to describe tracks as specific sequences of waypoints with well defined characteristics which grouped with good generalization should improve the normal trajectories modelling. The models achieved by using HTM are expected to improve the ability to discern anomalies between tracks using not just the same information used previously which was assigned to each point but also the inherent information that is the sequence defined by these points.

The final goal stands on using Hierarchical Temporal Memory to create an application capable of identifying anomalous behaviour on maritime vessels and to better understand the use, restrictions, concerns and capabilities of it's algorithms and theory at this project end.

1.5 Solution Approach

The first step of the solution should be defined as the development of a software system capable of retrieving data and processing it using HTM to create a model of the normal data. The initial system should be simple and use HTM to model vessel trajectories by using their simpler and meaningful feature, the GPS coordinates. After successful data modelling, anomalies should be

identified and the results analysed and parameters adapted to setup a basic system capable of modelling and anomaly identification even if with low performance.

After that new iterations of KDD should start with the need to analyse why performance is beyond the expected, find the relevant data which can improve characterization of ship class tracks and evaluate performance globally with other classes and if a crippling of global performance is found a new metric should be evaluated until the possibilities are exhausted. The next steps should encompass proceeding with analysis of different classes evaluating how the metric performs for each class and identifying the classes where worst performance is obtained.

As a complex project developed along years of work where the final result is not well defined with rather simple ideas of the possibilities present in a solution but where the way to achieve these results needs to be explored along the way it's not feasible to architect a solution with a big plan for all phases and for a final result. This is the kind of project where agile development techniques shine. Agile where short cycles accompanied by discussion of obtained results with planning for new short cycle of development should be the best way to achieve results. The solution should be incrementally developed from the then current point onward trying to improve on the problems detected and finding new ones while solving the challenges and gaining a better insight of the true capabilities and limitations of the methods already in use.

The solution should then be created incrementally by refining the application until a bottleneck is achieved with exhausted possibilities on the current methods exploration, only then should be necessary to find a new breakthrough with focus on solving the limitations found during re-evaluation of the results obtained or introducing new features found relevant to the application at that point of the development cycle.

Chapter 2

Fundamental Concepts

This section presents relevant information for the development of a solution and to better understand the present problem and previously work developed which should bring a new grasp on solution possibilities and on different ways to tackle future problems and challenges from previous experiences.

2.1 Data Analysis

With the crescent amounts of data availability a well defined approach to the analysis of large pools of data is crucial. In this project the data available mainly from AIS system is presented in raw text messages which include a lot of data from which knowledge needs to be extracted, this message represents in and of itself a huge amount of information with lots of important knowledge to be extracted but it's also filled with irrelevant data that won't bring anything to the expected results while it represents really huge amounts of resources on computational time and on the hardware needed to manage all this data. In this project the need of real time analysis of data from one of the biggest maritime areas in Europe, the Portuguese coast, creates an even more important emphasis on good application of knowledge extraction methodologies.

Knowledge Discovery in Databases (KDD) (Fayyad, Piatetsky-Shapiro, and Smyth, 1996, [7]) focus exactly on the development of methods and techniques for making sense of data, during this process the main mission is to transform raw data which, like in this case, represents humongous amounts of data into models that are capable of represent the entire set of data and could prove useful to predict future values or tendencies or simply help to abate the relevance of missing data. KDD is an iterative process and entails several steps 2.1:

1. Understand the domain data being treated, background knowledge and identifying expected process results;
2. Select a data set with all the necessary variables to apply the process;
3. Clean and pre-process the data treating noise, missing data and dealing with time-sequence data;

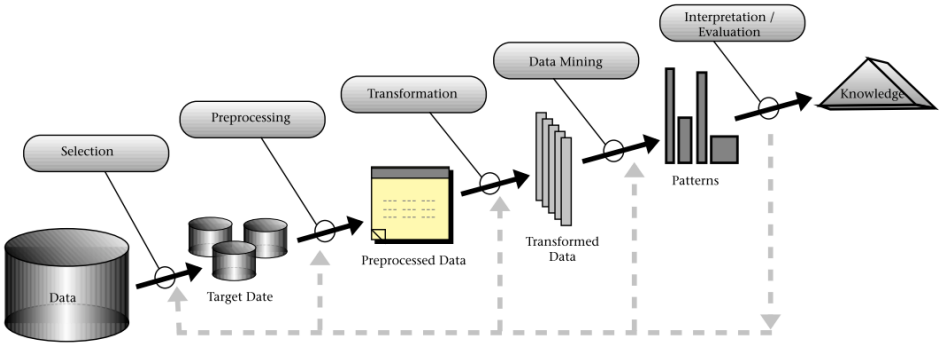


Figure 2.1: An Overview of the Steps That Compose the KDD Process (Fayyad, Piatetsky-Shapiro, and Smyth, 1996, [7])

4. Reduce and project data depending on the goal of task to achieve dimensionality reduction or transformation effectively reducing the variable number or establishing invariant representations for the data;

5. Map the goals of KDD process to data-mining methods, for example, summarization, classification, regression, clustering;

6. Exploratory analysis and selection of hypothesis to test, the selection of data mining algorithms to be applied with the respective parameter tuning and taking in consideration that selected algorithms must match expected goals, resulting models can present different characteristics like predictive capabilities or variable difficulty understanding the model;

7. Data mining: searching for patterns of interest and particular representation form or set;

8. Interpretation of the mined patterns with possibility of returning to previous steps for further iteration. Visualization of the extracted patterns, models or data from the models can present useful insights;

9. Act on the discovered knowledge: direct use, incorporate into another system for further action or simple documentation and report to interested targets. This step can also include the verification and resolution of potential conflicts with expectations based on previously believed or extracted knowledge;

Now that we have a methodology to use for the data analysis let's focus on the seventh step, data mining. There's two main categories of methods related with the expected goals, they are prediction and description. The prediction category emphasis's methods that give insight into unknown future values for specific variables using present data. Description is about discovering interpretable patterns from the existing data. Of course, as sometimes both actions are required it's common to find methods that can fit both categories.

Prediction and description can be achieved using several approaches (Fayyad, Piatetsky-Shapiro, and Smyth, 1996, [7]):

- Classification: learning a function that maps a data item into one of several predefined classes. An example is the classification of objects in image databases.
- Regression is learning a function that maps a data item into real-valued prediction variable, this means a correlation between two or more variables exists and the prediction depends on this correlation. An example is the estimation of a patient survival probability using the results of a set of diagnostic tests.
- Clustering is a descriptive task where one seeks to identify a finite set of categories (clusters) to describe the data. A simple example can be the clustering of the way-points of vessels where the direction of movement is well defined which generally can achieve a simple map of navigation lanes appointed by international regulations for specific destinations.
- Summarization involves methods for finding a compact description for a subset of data. An example is the use of mean and standard deviation to describe a data set.
- Dependency modelling focuses on finding significant dependencies between variables.
- Change and deviation detection focuses on discovering changes between data and previously measured or normative values.

2.2 Trajectory Mining

(Mazimpaka and Timpf, 2016, [18]) article defines a trajectory as a set of points where each point is represented by a spatial location, the time-stamp at which the point occurred and possibly other information that contextualizes the point history. The article proposes two mining methods:

- Primary methods which generally fall into two types of algorithms previously mentioned: clustering and classification. Clustering algorithms being unsupervised have the advantage of not requiring labelled data. On the article algorithms like ST-DBSCAN (Birant and Kut, 2007, [4]) which are an extension of the to be discussed DBSCAN algorithm on the next section. Another important alternative mentioned is the TraClus clustering algorithm (Lee *et al.*, 2008b, [16]) which instead of entire trajectories uses only trajectory sections.
- Secondary methods fall in three types:
  - Pattern mining which tries to discover movement patterns in trajectories;
  - Outlier detection which tries to discover trajectories not complying with the expected routes, which requires previous knowledge of what is the expected behaviour, this is a very interesting approach to solve the problem of anomalous behaviour in trajectories being limited by the lack of classified normality data.

– Prediction tries to discover the future location of objects based on already seen trajectories of them and expecting the same results.

2.3 Previously Employed Data Mining Methods

To extract knowledge from a dataset there's a multitude of methods that can be applied depending of data and resources availability and performance expectations. Previous work was developed to solve the problem introduced and revisited on this project which is summarized below.

2.3.1 Density Based Spatial Clustering of Applications with Noise

One of the most used algorithms applied to the set of data obtained from AIS systems is the Density Based Spatial Clustering of Applications with Noise (DBSCAN) which was also used in the past during the development of the first solution to the challenges appointed in this project to define the tracks, it uses the GPS coordinates as points and the SOG and COG to define a distance function to create clusters of points that can define a trajectory. This is the most common algorithm for density-based clustering. Given a set of objects, represented by points, this algorithm starts by labelling each one into three categories (Ester *et al.*, 1996, [6]):

- Core point, meaning that this point has a set of points within a given distance, which is called Eps and is parameter of the algorithm, and the cardinality of this set is greater than a given threshold called MinPts, also a parameter of the algorithm. Notice that the points in this set are said to be in the neighbourhood of the core point and are density-reachable from it (but the opposite may not be true). Also, the distance function commonly used is the Euclidean distance, but any function is supported;
- Border point, meaning that this point did not meet the criteria for becoming a core point but is in the neighbourhood of at least one core point;
- Noise point, meaning that this point did not meet the criteria for becoming neither a core or border point.

DBSCAN algorithm has important advantages that could make it an easy choice for the objectives, namely:

- Is great at separating clusters of high density versus clusters of low density within a given dataset
- Is great with handling outliers within the dataset

In the case of the tracks we expect to identify big clusters with several points and with the right distance function it should be easy to get clusters that describe tracks representing the trajectory from a start to an end point which could be a stop point or a high degree course change, while the rest of the points should be outliers pointing to fragmented tracks, stop points like ports or vessels

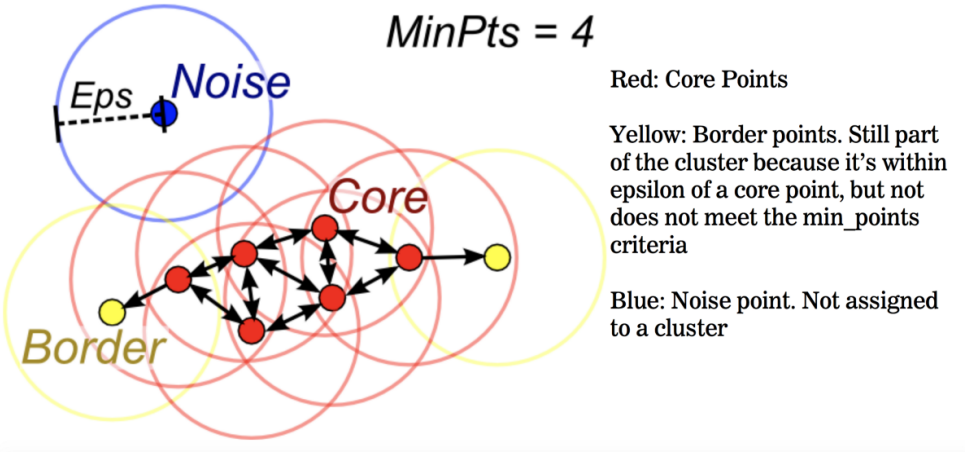


Figure 2.2: DBSCAN algorithm cluster example (Lutins, 2017, [17])

lost for a time-period. This method was later discarded since it's performance wasn't as good as the expected specially on the amount of resources needed to perform reasonably well under real time constraints.

2.3.2 Case-Based Reasoning

Later a new solution was developed where a example-based method of data mining is used, this method is called case-based reasoning (Kolodner, 1992, [14]), it uses representative examples from the database to approximate a model of similar tracks to which a track being evaluated will be compared using a well-defined distance metric. With a good distance function it's possible to identify anomalous situations but while the distance-metric complexity can improve performance it's also going to deteriorate the ability to identify the reasoning behind the anomaly identification. Using a group of simple metrics can achieve better reasoning results with the price appearing on the hight ratio of false-positives turning the distance-metrics into a complex problem. This solution presented good results with the pre-existence of data to create a model and where the ability to change the distance metric provides a way to improve the model fitness to detect more or less specific anomalies and to more easily access why is the current anomaly being assigned.

2.4 HTM Theory

After previously work developed using the above mentioned data mining methodologies a new one surged as an opportunity to improve current performance. That was the use of HTM theory derived algorithms to model the AIS data available and use it as a outlier detection algorithm to perform anomalous behaviour detection. On this section relevant HTM theory will be described for an easier understanding of later sections and of the reasoning behind the choice of HTM theory to carry the development of this project.

2.4.1 Hierarchical Temporal Memory

HTM or Hierarchical Temporal Memory is a theoretical framework based on how the neocortex functions and which also describes the technology based on neocortical principles. As described in (Hawkins *et al.*, 2016, [12]), HTM is both a theoretical framework for machine intelligence but also for a underlying biological intelligence system.

HTM is a biologically constrained theory on how the cortex works, while this is true it doesn't mean it attempts to include all biological details. HTM theory has it's focus on 'intelligence', to achieve that it bases it's theory on the brain neocortex which believes to be the main factor on the brain. While doing that it tries to extract the structures or algorithms the neocortex makes use to define a set of tools or long term principles to create intelligence excluding for example biological details which pertain to the restrictions imposed by a biological brain. It focus on the features needed on a information-theoretical view while these must rely on compatible principles from the biological theory.

Three features can be pointed on the neocortex (Hawkins *et al.*, 2016, [12]) which can represent the basic approach of this theory:

- Memory - each region on the neocortex is a simple memory system
- Temporal - the things being memorized are mainly temporal patterns
- Hierarchy - all regions perform memory operations, learning simple patterns and building time-based models of inputs which can have increased complexity when regions are hierarchically interconnected.

The work developed has been mainly, until the point of this writing, on the development of a single region with it's ability to memorize temporal-patterns, learn and make predictions based on previous memories of similar patterns which on success should result on the building block for a far more complex and intelligent system. The main components of the theory which translated into practical algorithms are the Spatial Pooler and the Temporal Memory which rely on Sparse Distributed Representations (SDR) as common data representations and to transform the system inputs into an SDR there's the need for encoders.

2.4.2 Sparse Distributed Representations

Empirical evidence demonstrated that neocortex regions represent information using sparse activity patterns in a multitude of areas which include early auditory (Hromádka, DeWeese, and Zador, 2008, [13]) and visual (Weliky *et al.*, 2003, [24]) areas which correspond to sensory features like audio frequencies or visual lines and edges, on later sensory areas more abstract and categorical information is processed like behaviour planning (Graziano, Taylor, and Moore, 2002, [8]). A piece of information is encoded on the inhibition of multiple distributed neurons and which number is a small percentage of the total amount available. That means information is represented by a sparse distribution of inhibited neurons which was translated into an array of bits where the binary value of the bit stands to the inhibition state of the neuron.

SDRs are then a large array of bits where most are zeros and few are ones (Purdy, 2016, [22]), SDRs are different from standard computer representations in that meaning is encoded directly on the representation, e.g. two SDRs with 1 bit in the same location share a semantic property and a bigger number of shared bits implies a closer semantic meaning between the two, in (Hawkins *et al.*, 2016, [12]) a letter is used as an example, to represent a letter of the alphabet using an SDR there may be a bit which represents if the letter is consonant or vowel, a bit related to how it sounds and a bit that represents the general location on the alphabet or on it's draw characteristics. On the SDR vector the bits correspondent to characteristics of a specific letter are the ON bits on a larger list of characteristics, like this if two letters share a lot of characteristics then their meaning is closer where that meaning is restricted to the list of characteristics chosen, on the SDR the ON bits shared between two SDRs determine the distance function of different representations. On the other way it's possible to infer that even if only a subset of this characteristics or if noisy info is included there's still an hight probability of a correct classification of the information on the SDRs as well as the ability to generalize and learn new information with related context.

The use of SDRs is as explained in (Hawkins *et al.*, 2016, [12]) a key component in HTM theory and one of the core principles to achieve truly intelligent systems. Sparse Distributed representations present some important characteristics as evidence to this belief (Hawkins and Ahmad, 2015, [9]) (Ahmad and Hawkins, 2016, [1]) :

- Hight capacity and low mismatch probability
- Reliable classification of SDRs
- Unions of SDRs
- Robustness against noise both on simple classification and after unions

There are also practical characteristics that won't appear on traditional computer data structures like storage (Hawkins *et al.*, 2016, [12]). As a sparse array of bits where generally only 2% or less of the bits are 1's there's no need to store all the bits, it's possible to store only the ON bits, on a 10000 bits SDR we could store only 200, to store this SDR we only need to store the location of these bits. Even better, since the ON bits have semantic meaning we could storage only part of this bits and still have meaningful info in storage, even if we loosed part of the info it stands true that the two SDRs are still semantically similar where this operation could perform a useful generalization function.

Generalization of information stands as a very important component for intelligent systems and SDRs provide the means to achieve it as on the previous simple example. Using the union of SDRs property it's possible to create an SDR capable of containing the semantic meaning of a group o SDRs. While we can't know what were the initial SDRs used to form the union it's possible to compare a new simple SDR and know if it is a member of the SDRs used to create the union. This possibility is due to the SDR sparseness which makes the chance of incorrectly determining the membership very low (Hawkins *et al.*, 2016, [12]). One simple example of the possibilities of this property is the characterization and generalization on objects. A ball is

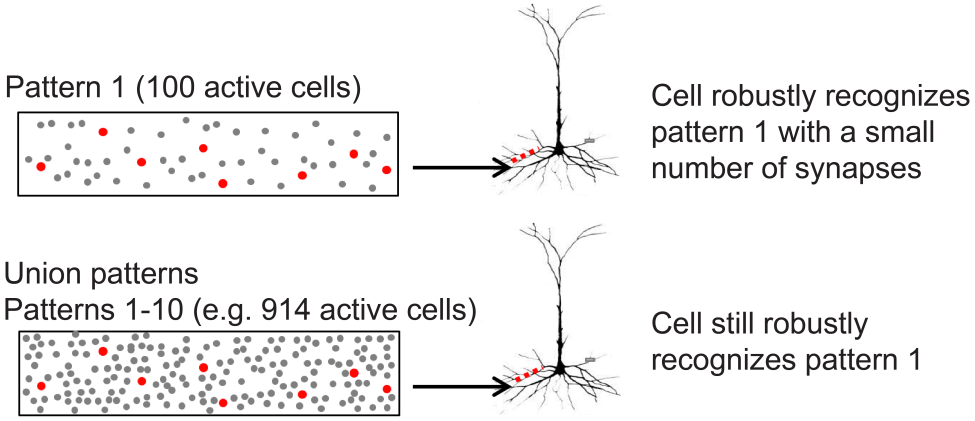


Figure 2.3: Union of Sparse Distributed Representations (Numenta, 2018, [21])

a simple circumference from afar, it's a circle when close, it's a round object if touched with no edges, it has no size limitations or even better it can have a widespread range of sizes, the same for colour range. So to a ball can be defined by the union of the SDRs from different balls, perspectives or even censorial information, with this it's possible to create the generalization SDR which represents a ball. When our brain or intelligent system compares an SDR with input information to be evaluated it's possible to compare it with this union to know how closely related are the two. Lastly, if when comparing there were contradictions between information it's possible to change this union by adding information from the input SDR, this would be the case of child learning what a ball is and defining and generalizing this term, at first a child could think a ball was like always like the football one but later it could find on rugby the ball is pretty different but still be called 'ball'.

2.4.3 Encoder

As discussed above SDRs are the information representations prevalent on HTM systems, to transform general data into SDRs there's the need to use an encoder. So, an encoder is the system component that converts the native data format into an SDR to be fed into an HTM system. Taking into consideration the need of semantic meaning on the converted SDRs it's important for an encoder to be able to capture the characteristics on the input information and following the principle that similar input values should be result in similar sparse representations which implies hight overlapping rate.

Encoders are analogous to the sensory organs on the humans. Different data types imply very different types of encoders since the nature of this data and the amount of meaningful information on this data can be very broad. On (Purdy, 2016, [22]) several aspects are noted as critical to the encoding process and examined individually in detail, respectively:

1. Semantically similar data should result in overlapping active bits.

2. The same input should always result in the same output.

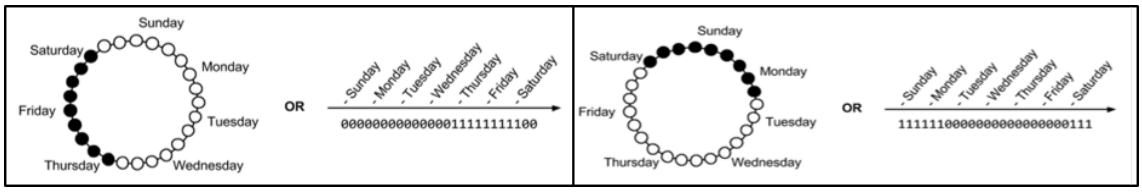


Figure 2.4: The encoding of the day of week on a date encoding (Purdy, 2016, [22])

3. The output dimensionality should be invariant with the input.

4. The output should maintain sparsity across all inputs while having enough active bits to handle sub-sampling and noise.

Examples of well documented encoders are also present on the paper (Purdy, 2016, [22]) which include:

- Encoders for numbers
  - Simple scalar encoder for ranged input
  - More flexible encoder capable of encoding an essentially limitless amount of numbers using an hash function
  - A log encoder that captures similarity between numbers differently based on how large the number is.
  - A delta encoder which is designed to capture the semantics of the change in a value rather than the value itself
- Encoders for categories
- Encoder for time
- Encoder for geospatial data

On 2.4 the encoding of the category 'day of week' which is only part of the meaningful information on a date input is performed as a simple example with a very limited number of bits but which can illustrate the encoding process. On this example the position of the bits ON represents the day of week, this position is meaningful and for example allows the presence of other information like the time of day, if all the one-bits are perfectly distributed around a day like Friday then it could be the middle of the day, on the other hand if most of the active bits are on the right then the current date is closer to the following day, Saturday. When designing the encoder it's possible to decide if the encoder will make use of certain information or not, in this case if the designer of this encoder just wants an emphases on the category with no need for other information then he can choose to discard that information. Still, meaningful information like the closeness to other days of the week can also be included, in this case with this limited amount of bits it's easy to verify that there's one-bits that are shared for three different days as a result that specific bit has

by itself the capability to restrict the range of a date. An example of the usefulness of this would be on the generalization of some action like the trash truck always comes on a Friday, Saturday or Sunday, for this range there's two bits generally more active on this bit array, if this two bits on the date encoder are added to the trash truck general SDR when later something happens and the trash truck comes on a Thursday the expectations based on the current information aren't compatible with the current happenings so it's possible to identify an anomalous behaviour. This happens even if the Thursday shares bits with Friday, while on the other hand this information can improve the quantification of the anomaly because it turns out it didn't came on the expected date but it came on a date which shares some semantic meaning with previously dates.

2.4.4 Spatial Pooler

The HTM spatial pooler, originally described in (Hawkins, Ahmad, and Dubinsky, 2011, [11]), is also a key component of an HTM system, it is responsible of converting arrays of bits previously encoded from sensory inputs into SDRs with a specific low sparsity making the best use of the information on the provided stream of bits from the encoder. This stream can be just one encode of one field input or on the case of multiple fields it's the concatenation of the results of the different encoders. While encoders should be designed to take in account some of the necessary properties of an SDR, they aren't necessarily met, for example, if one of the fields is being encoded into a binary category which should weight the same as all other characteristics which implies that the number of one-bits, W, should be the same as other encoders, then to create a binary category encoder the total amount of bits would be only $2 \times W$ where W bits on leftmost part would be ones or the rightmost W bits would be the ones. This encoder is a very practical example and illustrates an encoder that by itself won't create a proper SDR since the sparsity of the result is 50%, unless bits with no meaning are introduced the result won't be a proper SDR. Here comes part of the responsibilities of the SP which needs to translate the encoded input bit stream into a proper SDR with fixed number of total bits and sparsity, the SP does this and optimizes this operation by learning the recurrent patterns on the input streams and by selectively activating particular one-bits when this input pattern is detected.

The HTM spatial pooler is designed to achieve a set of computational properties that support downstream operations with SDRs, on (Cui, Ahmad, and Hawkins, 2017, [5]) several functional properties of the HTM spatial pooler are systematically analysed, this properties include:

- preserving topology of the input space by mapping similar inputs to similar outputs;
- continuously adapting to changing statistics of the input stream;
- forming fixed sparsity representations;
- being robust to noise;
- being fault tolerant.

The way the spatial pooler algorithm works can be simplified into some simple ideas:

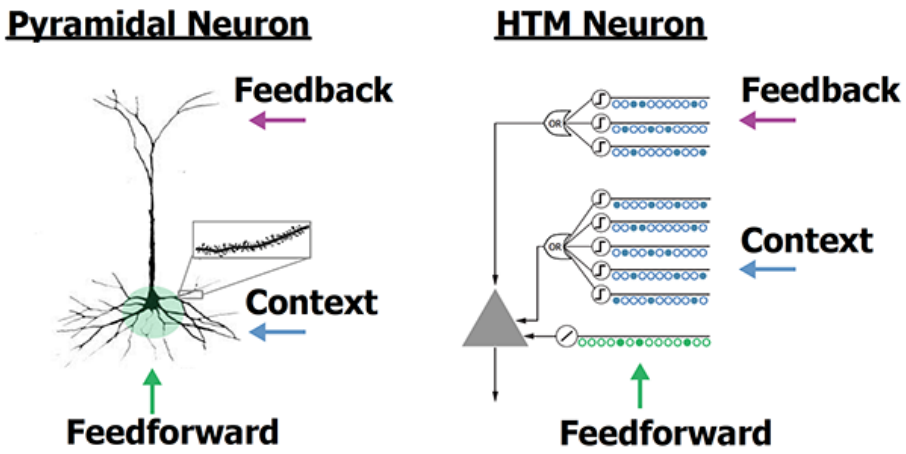


Figure 2.5: Pyramidal Neuron and respective HTM Neuron (Numenta, 2018, [20])

1. The SP number of bits n and the number of active one-bits w are fixed in number and each bit represents a neuron capable of creating, enforcing or enfeeble connections with specific multiple input bits on the input space;

2. Initially all neurons have attributed to them a set of random connections to a subset of the input space bits;

3. Every time a new input happens the w neurons with the most enforced connections to active bits on the input space are activated which results on the SDR with a fixed number of active neurons w. When this happens the connections between the active neurons and input one-bits are enforced while other connections are enfeebled or degrade. Inactive neurons suffer no changes.

Like this neurons on the SP are able to learn specific patterns on the input space, these are simple spatial patterns but are an important step to make use of the semantic information present on the input space to create SDRs which can be easily recognized by downstream neurons and which improve performance on an overall HTM system.

2.4.5 Temporal Memory

The Temporal Memory (TM) algorithm allows two things:

- Learn sequences of SDRs formed by the spatial pooling algorithm

- Make predictions based on current input and in the context of previous inputs

Temporal memory extends the spatial pooler to achieve the goals above, this is done by extending the previous idea where the spatial pooler made connections from neurons into input space

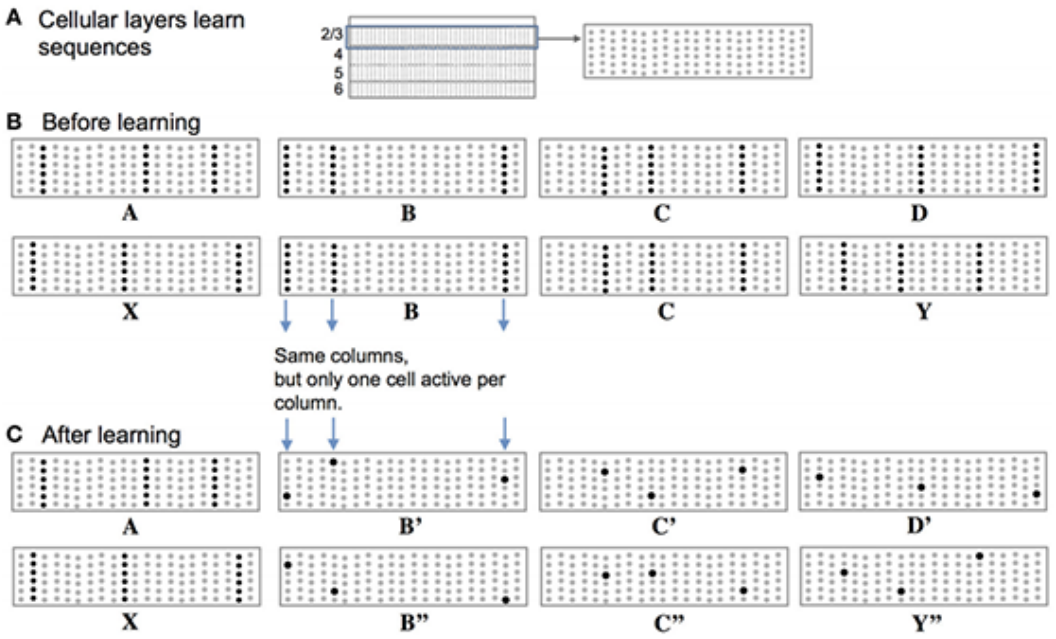


Figure 2.6: Temporal Memory sequence learning (Numenta, 2018, [20])

to make neurons learn and recognize specific patterns on the input space. In the case of the spatial pooler the connections represent proximal dendrite segments which are linearly summed to determine the activation of a cell within the neuron, now the Temporal Memory introduces the idea of a neuron as columns which are a stack of cells which share the same proximal connections (Hawkins and Ahmad, 2016, [10]). With no further changes all the cells in a column would share their activation state so differences between the stacked cells are the new distal dendrite segments which instead of being connected to the input space are connected to cells on other columns. On 2.5 both the basic neuron and respective HTM structure are presented including both proximal and distal connections for feedback and context respectively. This connections share the same principles previously mentioned with possible reinforcement or enfeeblement based on slight different method where instead of choosing the top w neurons a new neuron state is introduced called predictive state that is the core to the decision. A cell in a column enters a predictive state if the number of distal connections is over a threshold. This predictive state implies an expectation which when verified which means the predictive neuron column which the predictive cell was included is active on subsequent input then the connections that were previously responsible to induce the predictive state are reinforced. In the case a predicted cell exists on the column being activated then only the predicted cell is activated, on other cases columns without predictive cells have all their cells activated which would contribute to the activation of more predictive cells and to the ability of learning faster when the column is active on unknown context. On 2.6 a basic Temporal Memory progressive learning process is presented.

Basically Temporal Memory will provide information of the current context which is activating

the neuron and is basis to the ability of learning temporal patterns. More information on the TM algorithm including pseudo-code is included on (Hawkins *et al.*, 2016, [12]) as well as some numbers that can further enlighten the capabilities of this algorithm, e.g. if each column has 4 cells and every input is represented by 100 active columns, $(w = 100)$, then there's 4^{100} ways of representing the same input where each context will be represented by a different set of cells within the columns. Now that we represent the same input in so many ways it is important to know how unique each of those representation is, in (Hawkins *et al.*, 2016, [12]) is stated that nearly all random pairs of representations will have a 25% cell overlap, this means that while there's still a sizeable amount of shared meaning between the two the context still makes use of the other 75% to make inputs with different contexts easily distinguishable.

2.5 Next step: Anomaly detection with HTM system

Recalling the objectives of this project which are to create a system able to do the identification of anomalous behaviours on vessels using AIS data we saw earlier that many methods exist to achieve this, some were already used in which some problems like performance on using clustering algorithms or some difficult to solve restrictions like on cased-based reasoning where the current work isn't able to make good use of the temporal sequence of tracks which has the potentially to provide important information to the trajectories model.

Now that HTM theory was presented it's possible to point it's most important characteristic which is the ability to provide temporal sequence context into data, learn and make predictions which provides the necessary stage to implement a anomaly detection system since by having predictions it's possible to consider unpredictable situations as anomalies which makes this HTM based system a very interesting proposal to solve the current challenges.

Chapter 3

Developed System

In this chapter is described the overall system developed using HTM theory to detect anomalous behaviours on vessels movements described by AIS data through Portuguese maritime zone. The developed system is mainly divided on two main components:

1. A component that processes and uses HTM theory to model the AIS data;

2. A visualization application component providing the necessary means to better analyse the data both raw to understand the available data and to grasp the results obtained after processing the data.

3.1 Processing the data with an HTM based system

With the goal of processing the AIS data from maritime vessels and knowing that previously work had been developed to extract the relevant data from AIS log messages and which was stored on a MongoDB database which could easily be accessed from different applications written on about any programming language. Since there was the need to implement a system which would use the HTM algorithms and based on (Numenta, 2014, [19]) which uses HTM theory through NuPIC (Numenta Platform for Intelligent Computing) which is an implementation of the HTM algorithms by Numenta which is the company researching, developing and open sourcing HTM theory, on the paper NuPIC which is available as a python package is used to build a demonstration application using GPS data as input and the HTM learning and predictive abilities to detect anomalies when the described trajectory goes out of the predicted expectations after a learning period. With this and since there was no restriction against the choice of programming language Python was chosen since there was a well documented implementation of the HTM algorithms.

The basic application for processing the AIS data identified on Figure 3.1 as the oversee.py component can be divided from a functional standpoint into several blocks:

- The first block uses *pymongo*, a python driver package to the MongoDB database, to load the right data to process; a standalone mongo parameter file which can be changed to fulfil

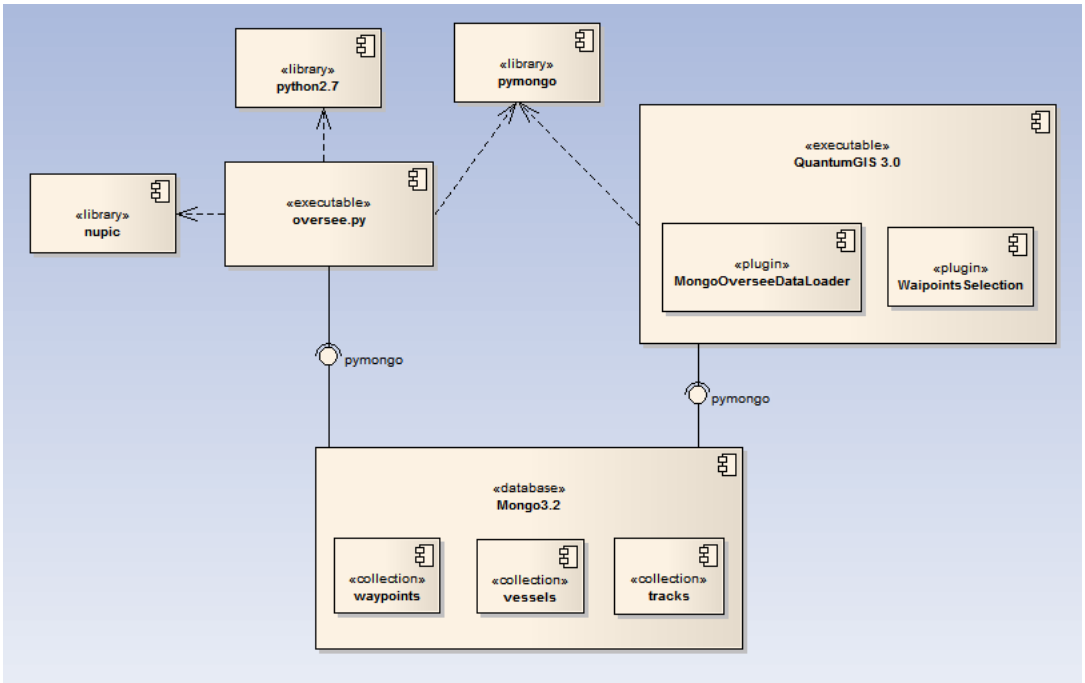


Figure 3.1: System component diagram

the user specifications is available which decides what data is selected and loaded using MongoDB query and aggregation options

- The second block which includes all the steps necessary to process the data using the HTM algorithms modelling it and calculating the anomaly results. It's internal process is described in the Figure 3.2 where CLA stands for Cortical Learning Algorithm which includes both the Spatial Pooler and Temporal Memory algorithms. There's also a standalone HTM parameter file where all options related to HTM algorithms can be tweaked where things like learning rate, memory decaying rate, model capacity and encoder parameters can be found.

- The last block simply saves the results from the HTM model to the database by updating the current information of the processed waypoint adding the current test ID and respective results.

The initial system only has a GPS coordinates encoder so only makes use of the vessels position which was the starting point to the work developed later which will include the test of different data to characterize the vessel trajectories and to the definition of corresponding encoders which will be explored in a later chapter.

Figure 3.2: Anomaly identification process using HTM (Numenta, 2014, [19])

3.1.1 Anomaly Score

The main output of this program is the anomaly score which is given by the direct calculation of the prediction error, it's computation is given by (Ahmad *et al.*, 2017, [2]):

$$S_t = 1 - \frac{\pi(x_{t-1}) \cdot a(x_t)}{|a(x_t)|} \tag{3.1}$$

where:

- x_t is the current input
- $a(x_t)$ is the sparse encoding of the current input and $|a(x_t)|$ is it's scalar norm, i.e. the number of one-bits in $a(x_t)$
- $\pi(x_{t-1})$ is the sparse vector representing the HTM internal prediction of $a(x_t)$

From 3.1 the anomaly score will be 0 if the current $a(x_t)$ one-bits are all matched in the prediction and 1 if none of the bits were predicted. An interesting characteristic of this metric which measures how well the model predicts the current input x_t is that branching sequences are handled well since the $\pi(x_{t-1})$ prediction vector includes multiple predictions formed by the union of all cells on predictive states which should include all possible predictions related to the current context.

3.1.2 Anomaly Likelihood

It was later added to the system a new way to measure anomalies which is described in (Ahmad *et al.*, 2017, [2]) as anomaly likelihood. While prediction error is an instantaneous measure of the predictability of the current input, the new measure instead of directly using the error S_t, models it into an indirect metric as the distribution of error and uses this to calculate the likelihood that the current state is anomalous. Anomaly likelihood is a probabilistic metric on how anomalous the current state is based on the prediction history of the model. The algorithm to compute the anomaly likelihood is described by (Ahmad *et al.*, 2017, [2]):

1. Maintain a window of the last W predictive error values S_t
2. Model the distribution as a rolling normal distribution where sample mean, μ_t, and variance, σ_t^2, are continuously updated from error values in 1 following:

$$\mu_t = \frac{\Sigma_{i=0}^{i=W-1} S_{t-i}}{W} \tag{3.2}$$

$$\sigma_t^2 = \frac{\Sigma_{i=0}^{i=W-1}(S_{t-i} - \mu_t)^2}{W-1} \tag{3.3}$$

3. Compute a recent short term average of prediction errors, $\tilde{\mu}_t$

$$\tilde{\mu}_t = \frac{\Sigma_{i=0}^{i=W'-1} S_{t-i}}{W'} \tag{3.4}$$

where W' is a window for a short term moving average and $W' \ll W$

4. Calculate the Gaussian tail probability, L_t, using a Q-function:

$$L_t = 1 - Q(\frac{\tilde{\mu}_t - \mu_t}{\sigma_t}) \tag{3.5}$$

5. Apply a threshold to L_t based on user-defined parameter ε to report an anomaly. An anomaly is detected if:

$$L_t \geq 1 - \varepsilon \tag{3.6}$$

Currently the last step is not executed by the system while processing the data but the anomaly likelihood, L_t, is saved for later use which permits later research on the best fitting threshold, ε, in different circumstances.

3.2 Visualizing data

This project involves working with big datasets which can include millions of points which include different related information and which can form trajectories and big movement patterns. Since one objective is to process this data and find anomalies the ability to analyse the available data and to interpret the results using an expressive canvas is fundamental. Since all the data can be spatially distributed on a map for better interpretation a Geographical Information System (GIS) was the right choice as visualization and data analysis tool. Instead of developing a new GIS tool which wasn't the focus of this project the open source GIS tool QuantumGIS (QGIS), an official project of the Open Source Geospatial Foundation (OSGeo), was the geographical information system of choice. This is a well documented, multi-platform and open source GIS with various tools to aid on the exploration and analysis of data with a very important characteristic which is an extensible plugin architecture and libraries that can be used to create plugins or to integrate it into a new application, both using C++ or Python.

The QGIS application has a lot of general tools which will help on the design of visualizations, on the selection and inspection of different features which can be found on Figure 3.3 and on data loading from different sources. Unfortunately there was no way to load data from Mongo databases which was solved by making use of the plugin architecture and developing a simple plugin which could use the previously mentioned *pymongo* package to load data from the different mongo databases. On the MongoDB application different databases house collections with information pertaining to the waypoints, tracks and vessels. Each database has these three collections

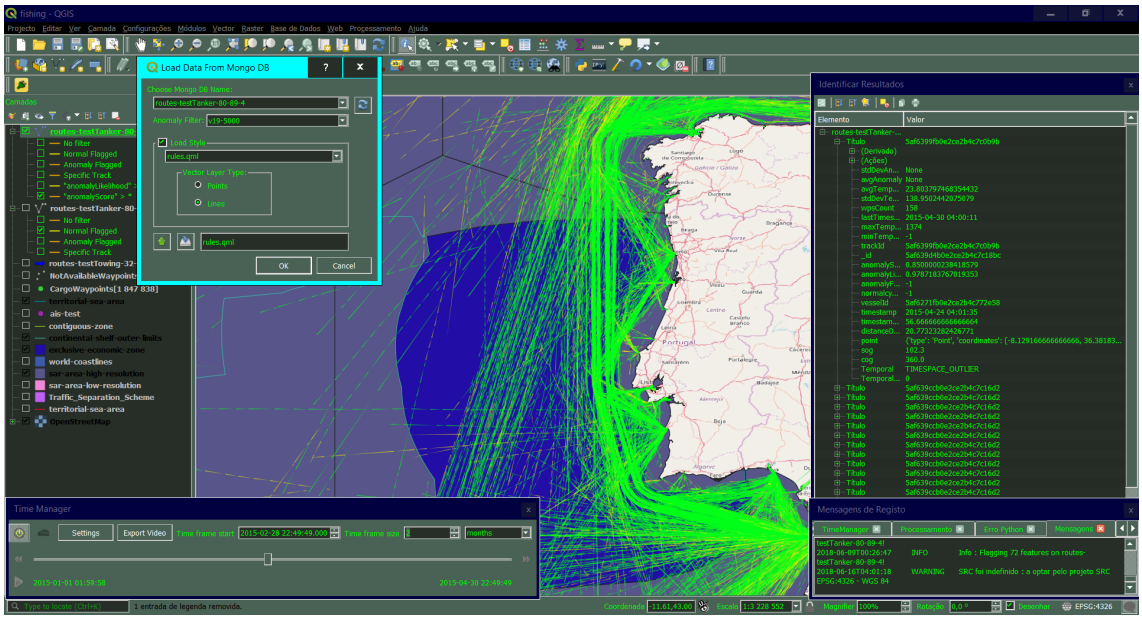


Figure 3.3: QuantumGIS normal behaviour trajectories visualization on Portuguese coast; layer visibility selection (left), feature inspection (right), data time manager plugin (bottom left)

but are correspondent to different vessel types like tanker, fishing or passenger vessels. The initial plugin was a simple script but after the growing need to use other scripts to perform some simple functions which include:

1. The loading of different databases from MongoDB and the choice of versions of anomaly scores resulting from different tests
2. The layer setup and creation of trajectory lines based on the vessel waypoints
3. The ability to save different complex and personalized layer styles and to automatically apply them later on loaded data layers

On Figure 3.4 the developed plugin user interface is shown where these functions are available. The plugin main purpose is to facilitate the data loading and necessary layer setup process since with the right options the process is then carried automatically. When having various big databases and tests which can't all be permanently loaded due to performance issues and being a recurrent action a simple plugin can be very helpful and turn the data analysis process a lot more agile.

Other plugins for feature (geometric figures such as points, lines or polygons) selection were developed without user interface such as:

- Similar features selection plugin which when selecting features on the QGIS interface automatically expands selection to all other features with similar parameter, generally with same track or vessel IDs
- Plugin to manage flags on the database waypoints collection used to aid the creation of sub-datasets such as an anomaly or normalcy flagged dataset.

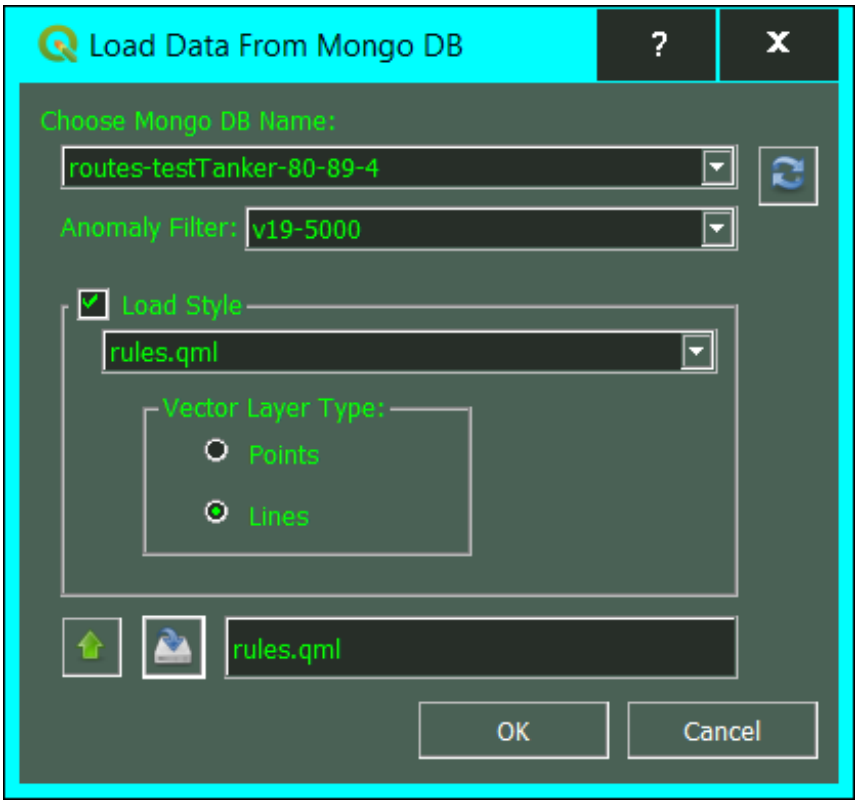


Figure 3.4: Developed plugin user interface for QuatumGIS

These plugins while simple can be easily developed and adapted to perform the needed functionality and were a very useful tool to improve the application usability on specific contexts.

3.3 System Version: 1.0

The developed system was divided on the two respective components, data processing and visualization, then further use of the Python *matplotlib*, *numpy* and *jupyter notebook* packages to aid on the statistic analysis of the results. It was not objective of this project to assimilate all these into a full-fledged application but it is still interesting to note the possibility of joining these components to create a flexible and comprehensive application for Geospatial data modelling, anomaly detection, visualization and analytics's.

On further chapters the research is mainly focused on improving the described basic system for this specific problem by tweaking on the HTM system parameters and by adding new information to the model with new input data and encoders. Those are more specific to the current problem while the previously described system has the capability and expectation of establishing itself as the core to be used under different circumstances on geospatial data modelling and anomaly detection.

Chapter 4

Experimental Work

This chapter introduces part of the experimental work performed along this project. The main goal of the described work is to improve the previously featured basic system by incrementally adapting it to fit the expected purpose. This is achieved mainly by tuning the necessary encoders and by analysing the data to understand the potential information to be added into the system input space to improve normal trajectories modelling.

4.1 Geospatial Coordinates encoder tuning

The first step taken to improve the basic system was tuning the Geospatial encoder. This encoder described in (Purdy, 2016, [22]) makes use of an hash function to encode unlimited positions in a bounded SDR and assimilates speed to improve the encoder performance. The way this works is quite complex but some principles are important to understand it and perform the encoder parameter tuning so next some of them will be summarily described:

- The encoder creates an SDR where the position of a square is decided by using an hash function which creates a correspondence between coordinate positions and positions on the bounded SDR;

- The active W bits to describe the current position are bounded by this square and weighing scheme is used on the bits which will give weights to specific bits making them more likely to be the active ones inside a square bounded area;

- The size of the square is decided using the speed input which makes this encoder more adaptable. This will make the square smaller for low speeds in which case the set of active bits are chosen from a smaller set. This means that only other very close positions and with small square bounds (and speed) will share many active bits necessary to identify positions as the same. On the other hand it will make positions while on hight speed a lot more general so that even relatively far positions can be interpreted as similar.

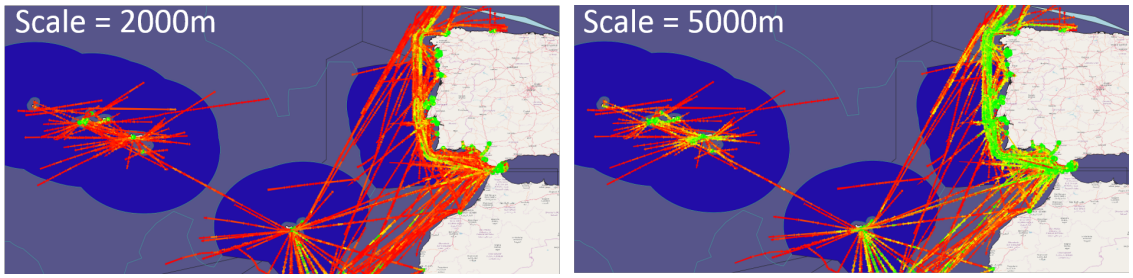


Figure 4.1: Visualization of resulting trajectories by using different scale parameter with green for normal movement and red on courses deemed anomalous (4 months data on passenger vessels)

- The concept of closeness above is decided by a parameter named scale. Scale basically decides the range in which positions are described with very similar SDRs in which case are basically the same positions.

Now that the encoder was described it's possible to realize the importance of the scale parameter. The objective of this work is to model general trajectories of vessels on maritime zone where there are no roads to limit the vessels movements and while there are limited zones for some specific travel courses these are not physical barriers so the trajectories are always defined by lanes several kilometres wide. This fact implies the need to use a big scale to enable the system to model the general trajectories instead of modelling very particular tracks.

There's also a different factor which will contribute to the scale choice namely the positions rate. The data being used in this project have tracks with positions at regular intervals of about 1 hour, this interval is not small at all and on high velocity trajectories vessels will have positions with distances going easily above dozens of kilometres. To model a trajectory all intermediate positions need to be filled so one more time the need for high scale values is the conclusion which in this case could be made up by using more data if available with limitations since the generalization performance on the trajectory models would still be affected if the scale is too small.

With the above factors it is the case that the scale for this encoder and while using this particular dataset needs to be on the several kilometres limit while some level of tuning is needed to find the right number where the generalization performance is neither too good or too bad. At this point where the project just started the simplest way to solve this problem was by experimentation where the resulting models were compared using visualizations which made clear the learned trajectories. Models were defined using scales between 2 and 10 km from where some information could be extracted:

- A scale of 10 km is too big resulting in problems since a great number of positions will be deemed as similar and each position encompasses too big of an area which creates an over generalization of the model in where any movement around within the trajectories neighbourhood area can be considered normal.

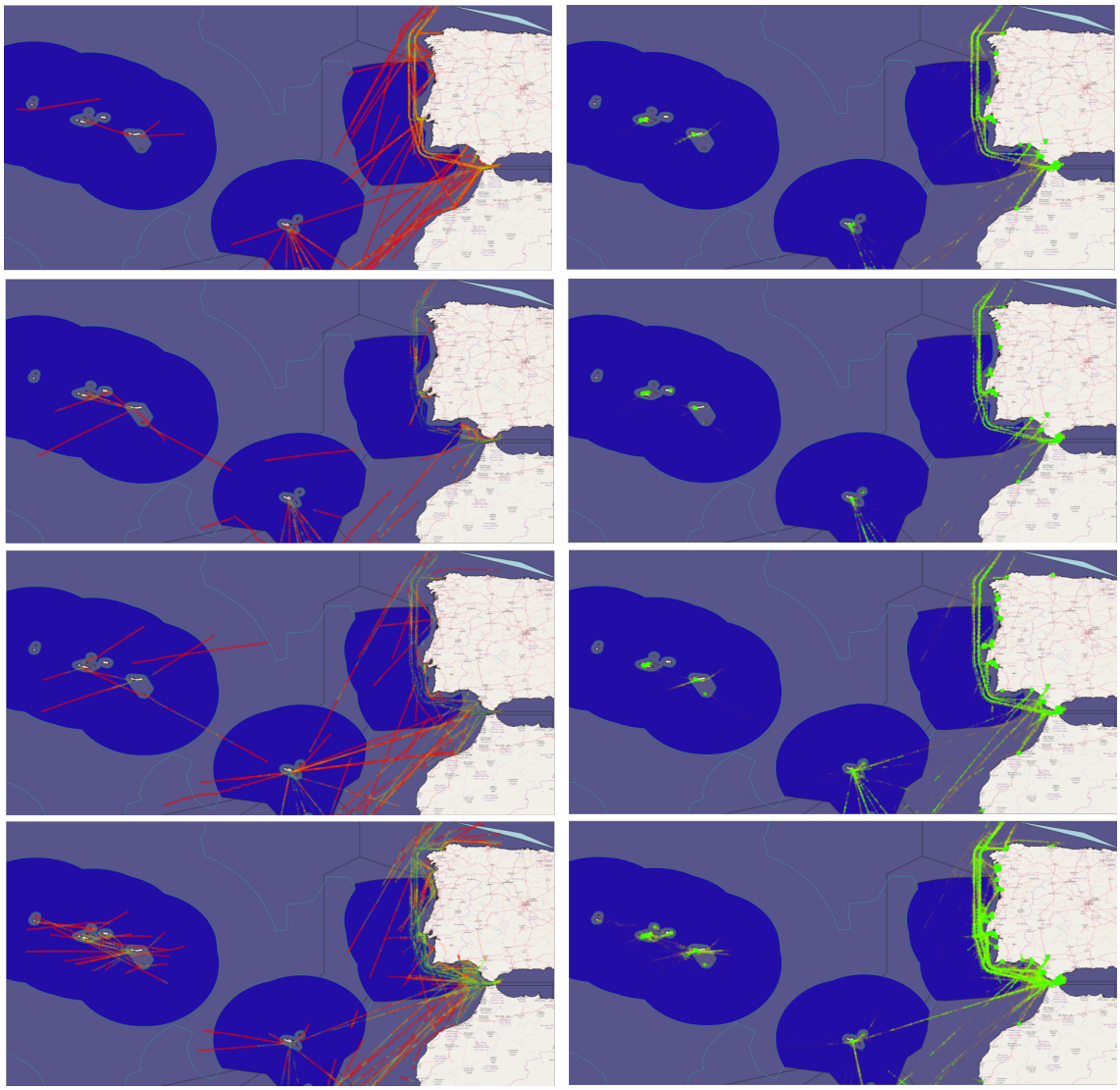


Figure 4.2: Visualization of resulting trajectories divided by month; Highlight of anomalous behaviour on left and normal on right image (1 month data; passenger vessels; scale:5000)

- 2 km is too small and even with large amounts of data the model still has difficulties on the generalization of trajectories.

In the end the scale chosen was of 5 km taking note that if more data is available smaller values could be a better choice. The dataset being used here and in future experiments unless explicitly indicated in contrary is representative of 4 months of data with vessel positions distanced by 1 hour. On Figure 4.1 is presented the resulting visualization of passenger vessels processed data with scales of 2 and 5 km and by using the anomaly scores to distinguish between learned trajectories with scores of 0 or very close and remaining ones ranging from yellow to red marking colour depending on how anomalous were considered by the system.

It's also possible to divide the data into different periods of time to better understand the

progressive learning of trajectories. On Figure 4.2 every pair of images corresponds to one month of data where the left image highlights anomalous behaviours and the right image normal ones. Some information can be summed:

1. On the first month there's anomalous behaviours on all possible trajectories, the model apparently started learning the trajectory between the Mediterranean and the north of Europe;
2. On the months after this trajectory presents almost no anomalous behaviour while on the fourth month when apparently there's more activity some slight accumulation of generally yellow positions can be found which should improve the generalization of this trajectory.
3. On the later couple of months other trajectories start to be modelled including the ones from Mediterranean to Madeira Archipelago.
4. The maritime activity on various trajectories can be very different depending on the time period. On the second month there's very little activity compared to the other periods.

The choice of the scale parameter can be extremely relevant to the performance of the modelling process and that's why it was important to describe the factors which can affect it and what are some of it's consequences in the results. While only this type of vessel, respectively the passenger vessels database, was used along this section other vessel type databases were used during the decision process and the result achieved was similar with the modelling of the major trajectories using 4 months of data.

4.2 Data Analysis

After having the basic system tuned it was time to understand how the data available for each waypoint could be used to better model trajectories. For that reason some data analysis was needed which will, in part, be described on this section. The subsequent descriptions stand as examples of some of the possible results made during the analysis process and has it's focus on the analysis of the fishing vessels database.

4.2.1 Speed Over Ground (SOG)

The speed over ground (SOG) was previously already used on the Geospatial encoder, while this is true it is used as a mean to improve the results when using positions in which case the speed per se has a much lesser weight on the SDR results specially when the objective is to model trajectories where positions have very hight generalization. To better understand if using the speed as an input space variable could improve the system results some data analysis was performed, since speed over ground was previously used on the CBR metrics it made sense to use some data from there to better understand the difference between approaches. The results expected were that anomaly score average should be a lot higher from a SOG threshold onwards since vessels with hight velocities are unexpected specially since the vessels are of the fishing kind. On Figure 4.3 it's possible to make some observations:

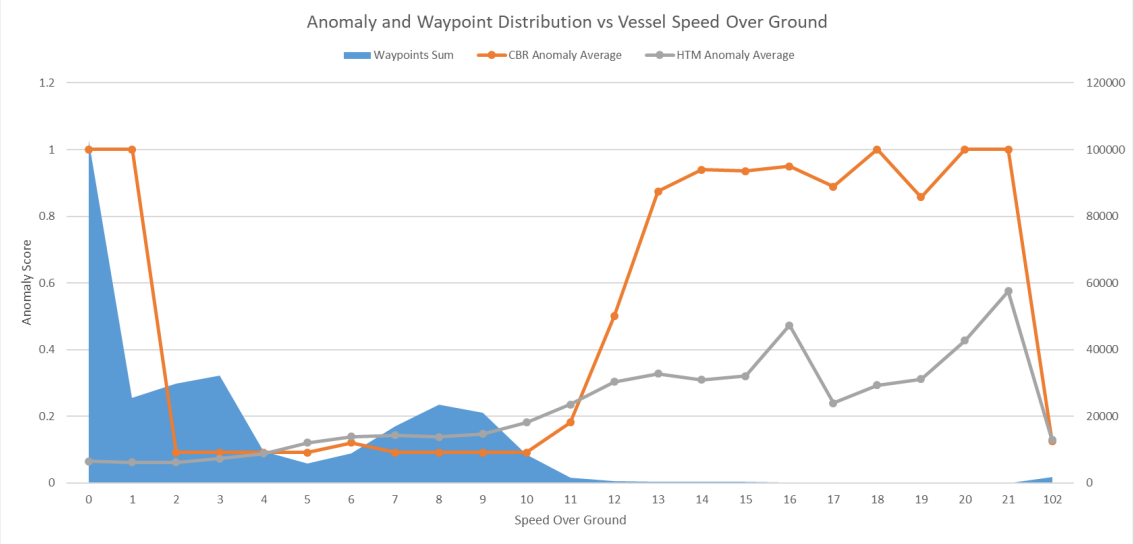


Figure 4.3: CBR and HTM anomaly average and waypoints distribution over SOG

1. There's a huge number of very low speed waypoints that mostly represent places where vessels stop (SOG between 0 and 1). It's also important to note that the CBR discards most of the very low speed waypoints while the HTM is modelling them as any other points which explains the strange discrepancy on this speed segment.

2. Most waypoints are between the speed range of 1 and 12 with two apparent cluster on from 1 to 5 and 5 to 12. The HTM anomaly average on the 5 to 12 slightly increases which doesn't happen on the CBR.

3. On the speed range of 12 to 17 where some very small number waypoints can be counted the anomaly average increases steeply on the CBR and while still evident on the HTM system is not nearly explicit.

4. From a speed of 17 onwards there's even less points where the high anomaly average is clear.

5. There are some points with a SOG value of 102 which should have been identified by both systems with highly anomalous average since that's the actual SOG maximum possible on any AIS messages and very unlikely to be real. In this case most likely than not the value on the message doesn't reflect the actual vessel speed because some problem occurred on this vessel AIS system.

On Figure 4.4 we can see multiple visualizations which correspond with the previously considered interesting ranges. It's now possible to identify that on these ranges different kinds of vessels can be distinguished:

- SOG 0 to 4 - the biggest concentration of vessels is in this range and corresponds with the fishing vessels around the coastline. The trajectories for these are generally well modelled,

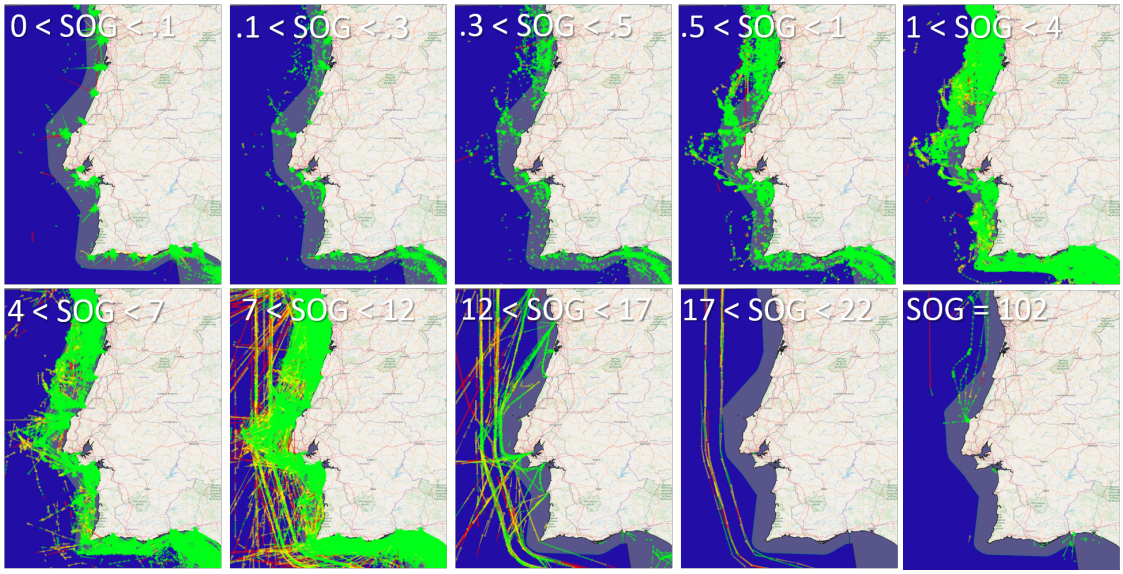


Figure 4.4: Waypoints visualization on different Speed Over Ground ranges (speed in knots)

at least on the current position basis, there's probably the need to add other informations to identify specific anomalies on this range since vessels with low speed and very close consecutive positions will create a very general trajectory all around the coastline.

- SOG 4 to 7 - this is the range that marks the increase of anomaly average, it's possible to identify some points which aren't so close to the coast.

- SOG 7 to 12 - on this range it's possible to find both the faster fishing vessels close to the coastline as well as vessels on long course trajectories which only now can be clearly identified. While it's possible to clearly identify some long course trajectories it's also possible to observe that they aren't well modelled being identified as anomalous which is the most probable cause for the previously noted increase in the average anomaly.

- SOG 12 to 22 - this time only long course trajectories are visible, on this range almost no trajectories close to the coast can be observed. The system wasn't able to model these trajectories quite well when they were out of the coastline zone where the previous trajectories were, on the other hand even with higher speed the trajectories close to the coastline are still being considered normal which indicates that even with enough speed difference they are being predicted based on the model of the low speed generalized positions. This is indicative that using SOG as an explicit input could help to better separate trajectory models, this should be relevant specially in zones where the concentration of waypoints is very hight.

- SOG 102 - The speed affecting the Geospatial encoder isn't enough to be critical, positions close to the coast are still being identified as normal even with unprecedented high speed.

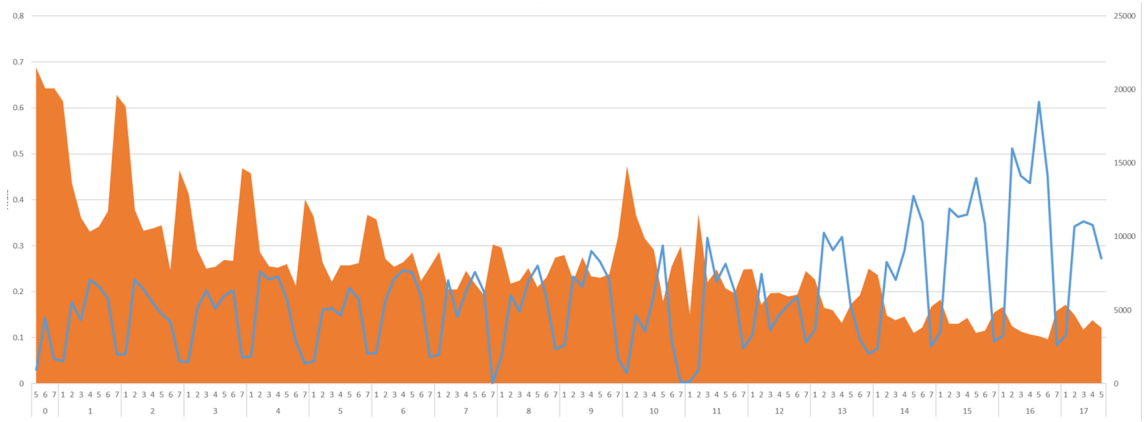


Figure 4.5: Waypoints (blue) and anomaly average (orange) distribution over weeks and days of week

With previous observations it was concluded that SOG could be a very good input to improve the trajectories modelling. The different ranges show that SOG could help to better separate data of very different kinds of trajectories specially on hight waypoints concentration zones which if more data is available would include most areas.

4.2.2 Time

After looking into the speed as a possibility to improve the trajectories model it was possible to note that information to aid on the differentiation of trajectory types could improve the capacity of generalization while limiting the wrong assimilation of data from apparent unrelated ones. This kind of data works as a category which would improve the trajectory differentiation. On HTM systems which rely heavily on temporal/sequence memory where patterns are identified one of the inputs which can be generally found is the time since lots of patterns can be better described if related with time based information. For example if the current system was used to process traffic data at distinct geographic locations the use of time related data would most likely than not be very important since most normal spikes on traffic should be related with specific times of the day like early morning or late afternoon. On the other hand to identify strange spikes on the traffic volume it would be quite important to have time related information since if no info on the time was provided the sequence would be distorted from the anomalous spike on and until it was learned again, if time based info is provided after an anomalous spike at unexpected time other spikes are still related to time and can be considered normal even if sequent to a sequence which took the model into unpredictable results.

On our data every position message is accompanied by a timestamp which can perfectly temporally locate the message. On 2.4.3 one of the encoders mentioned on (Purdy, 2016, [22]) was the time encoder which can be composed of multiple simpler encoders, from a timestamp a lot of data can be extracted and since SDRs make use of semantic information it is a good idea to use a composite SDR which makes the best use of the relevant information present on different

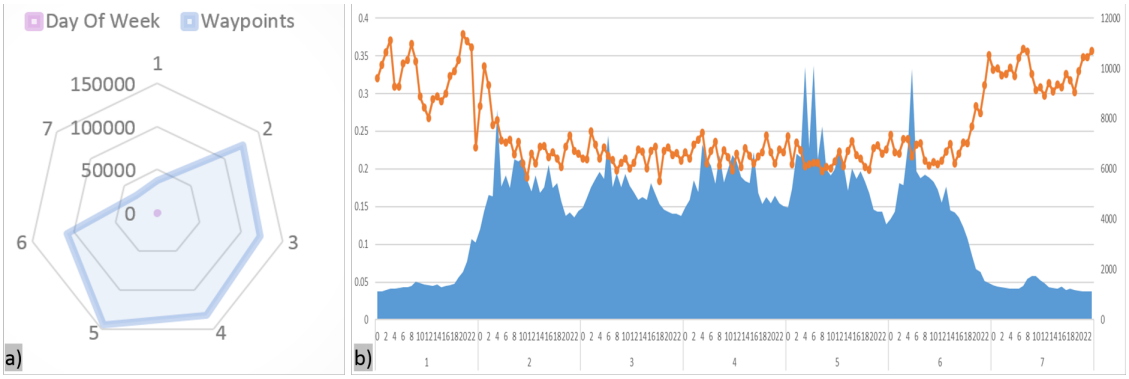


Figure 4.6: a) Waypoints (blue) distribution over days of week b) Waypoints (blue) and anomaly average (orange) distribution over days of week and time of day

situations which means information with little meaning to the goal at hand can be forsaken. This is actually quite relevant since if meaningless information is added on a model used to make anomaly detection it will only distort the results, e.g if the time of day is added but no actual patterns relate to this variable than the only influence would be that even when other inputs aren't correctly predicted and an anomaly would be detected the time of day could still be correctly predicted which would only lower the anomaly score on this situation. Some of the time related information which can be extracted into different SDRs are listed next:

- Weekday vs weekend
- Day vs night
- Month of the year
- Day of the month
- Day of the week
- Time of the day
- Minute of the hour

To find out which kind of information could be relevant to current data some analysis was needed, on 4.5 it is possible to observe the waypoints and anomaly distribution over the weeks, some information can be noted:

1. As expected, along the weeks overall anomaly average decreases since most trajectories are modelled.
2. It's also possible to observe some anomaly spikes on weekends (day 1 and 7) accompanied by the negative spikes on the number of waypoints. Most likely specific types of trajectories on the weekends are not well modelled by the system.

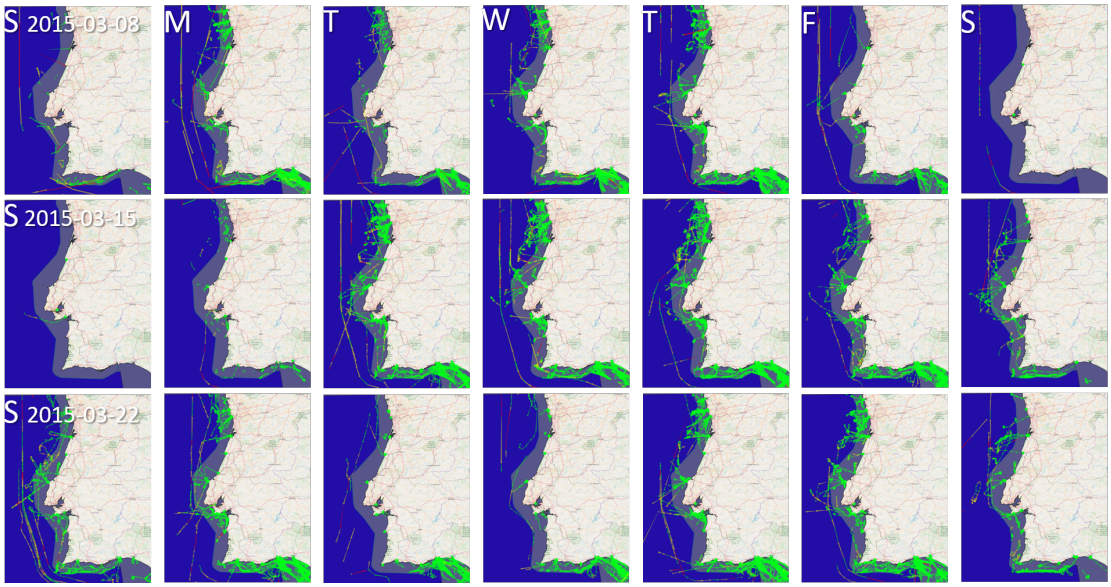


Figure 4.7: Waypoints distribution over days of week (3 weeks: from Sunday to Saturday)

Other analysis were performed on the data by aggregating the data into specific time periods where the results were that no other information seemed to be as relevant as the weekdays since no other related time patterns were identified, the presence of a simple pattern over the time of day could be found but with very little impact, reason why it was noted but will very probably be dismissed later. To better understand the impact of the day of week and time of day patterns the data was aggregated over both time intervals and resulted on the graphics on Figure 4.6 from which we can observe:

- Clearly skewed distribution of the waypoints towards weekdays with hight anomaly average on weekends which confirms a clear pattern where most likely some trajectories are mainly on weekdays. Since the vessels are of the fishing type it's expected that during the weekdays the normal work is performed with rest on the weekends.

- It's also possible to observe patterns over the time of day but not so evident and which is still connected with the day of the week since there's a pattern on weekdays and a different one for weekends.

It was possible to identify the possibility of using week days or the weekend binary category as additional information since it looked like two distinct patterns for both waypoints and anomaly average distribution were identifiable. Still there was the need to understand if this information is related with different trajectories since it was possible that it was simply related with a reduced activity rate on those days without difference in terms of normal trajectories in which case this information won't provide any benefits. To better understand this question the simpler way was to create a visualization representative of the different week days to test these hypothesis. On Figure 4.4 it's possible to see some of the results observed, answers about the usefulness of using

weekdays weren't quite conclusive since while it's possible to clearly identify very different levels of activity that's not the kind of anomalies being identified, since this reduced activity looks to be affecting all kinds of trajectories it could be considered as not very useful to help on the modelling process, it is not particular able to categorize any subset of trajectories.

4.2.3 Last notes

Previously we have seen how to find if certain information could be more or less useful if added to the system input space, this kind of analysis can be very important to better understand not just the variables being analysed but also to better understand the current results. These were just examples on the current project but which can clarify on how the analysis can be performed by using information both from raw data and from the current system results. While doing this analysis the main finding about the current state of the model was that in lots of cases the generalization is too deep since there's not enough information to separate trajectories efficiently which indicates the need of data to help better categorize them specially on zones with lots of activity. This need is possibly emphasized by the restriction created by the use of a big scale value on the geospatial encoder which results in the easy creation of models with a big generalization level while contributing negatively to the anomaly detection process. The way chosen to solve this problem was then to add more information to the model. The simplest data to add to better categorize trajectories are the speed over ground previously seen since it helps distinguish different types of trajectories by speed, on the other hand there's actually other info which is implicit on previous visualizations which is the course over ground which implies the directionality of the trajectory movement and could help on diminishing the effects of over generalization.

4.3 Adding data to input space

As discussed before there is the need of an encoder to add any data to the HTM system input space. Of course, most data can make use of open source developed encoders previously mentioned in which case the real work is not in designing a new encoder but in choosing, adapting and tuning the encoders into the project accordingly with pretended goals and with the data characteristics. While this is true it's important to note that encoders described further in this section were the ones found to achieve the best results but maybe there were other possibilities not considered with better performance.

4.3.1 SOG Encoder

The first new input added was the speed over ground, this value comes directly from the AIS position messages and has the following characteristics:

1. It represents a vessel speed over ground in knots at the position in the message
2. It is limited between 0 and 102

These characteristics are very simple and there's even a value range thus it was expectable this value could simply be encoded by a scalar encoder while in practice this wasn't so simple as that. The first test used a Scalar Encoder with:

- Width, $w = 5$ —- Number of active bits on the SDR
- Bucket, $n = 35$ —- Total number of bits on the SDR

Most of the simple encoders can be tuned by these values plus other specific information, in this case the range given by minimum value and maximum which were identified previously, 0 and 102 respectively. The w and n parameters also represent the encoder resolution and radius, this values could be given directly in which case other parameters shouldn't be provided since they are dependent. In this case:

- $range = max - min = 102 - 0 = 102$
- $radius = \frac{range}{n/w} = \frac{102}{35/5} = 17$, which means that semantically, only representations for numbers with a difference of at least 17 units are totally different, numbers within the 17 range will share bits which means they are semantically closer.
- $resolution = \frac{range}{n} = \frac{102}{35} = 2.9$, this means that numbers need a difference of at least 2.9 units to have different representations

These parameters were based on the analysis of the SOG distribution, it was possible to identify some limits like 17 which isolated a very specific type of trajectories coupled with a 2.9 resolution which would enable the difference between the various ranges which related to trajectories on 4.4.

After applying this Scalar Encoder with these parameters adding SOG as an input the first results observed, based mostly on the histograms of anomaly scores and probability for both an anomalous and a normal subset, were summarily:

- An improvement on the performance of the model on the identification of expected trajectories with the average anomaly decreasing.
- An increase on the average anomaly probability when detecting anomalous behaviours.
- A general improvement of the anomaly detection since the average anomaly probability decreased in general and increased on the anomalous population.

It was later identified the possibility that the value of $w = 5$ should be too low since the standard value for Scalar encoders was 21 and taking in account that the value standard for the geospatial encoder was 50 which made the difference between the weights of the inputs apparently too large. A new model was obtained with:

- Width, $w = 21$ —- Number of active bits on the SDR
- Bucket, $n = 147$ —- Total number of bits on the SDR

which is basically a scaled version of the previous encoder with the changes being the weight and an improvement on the resolution of the encoder which could actually reduce the generalization since a bigger number of different values were now possible. The results were that the average anomaly was further reduced but the anomaly detection suffered since the average anomaly on the anomalous population decreased to levels worst than both the previous version with SOG encoder and the initial version.

After further analysis of the encoders options, results and of the goals of using the encoder a potential problem was identified with different resolutions and it's relations with the ranges of speeds. A lower resolution should improve the generalization capability while adding a new categorization on the trajectories which would improve both general average anomaly and anomaly detection. On the other hand with higher resolution the generalization shouldn't be so good which shouldn't have improved the results which was also true to the anomaly detection since only the average anomaly decreased which doesn't really improve detection, i.e. if both the average anomaly on the anomalous subset and on the normal subset decreases there's no increase on the difference so no improvement on the detection. At this point the question was why would the average anomaly decrease if the generalization was worst, to this question no definite answer was found.

On the hypothesis that the previous possibility was right and the problems were related with the resolution a new encode was designed. Using the properties of a Log Encoder to have a variable resolution. According to previous data the encoder for SOG should probably have a lower resolution on higher values and higher resolution on lower ones. The LogEncoder used had the same $w = 21$ but $radius = 0.3$ and minimum value changed to 1 (values under 1 would be encoded as 1), which meant complete different representations of:
$10^0, 10^{0.3}, 10^{0.6}, 10^{0.9}, 10^{1.2}, 10^{1.5}, 10^{1.8}, 10^{2.1} \approx 1, 2, 4, 8, 16, 32, 63, 126$
which should allow a good range generalization since trajectories with SOG around this values should have a similar representation with the adequate resolution, i.e the meaning difference between 1 and 2 is about the same as 16 and 32, this results in a better description of the idea of speed ranges. The results for this encoder were better than all previous ones, there was an even sharper decrease of the anomaly average in general while the anomalous subset suffered an increase in the anomaly average, together these results mean the general improvement on anomaly detection.

The final encoder parameters are:

- Log Encoder
- $w = 21$
- $radius = 0.3$
- $min = 1$ & $max = 102$

4.3.2 COG Encoder

The course over ground is also directly obtained on the AIS position message, it's characteristics are:

1. It represents the vessel direction at the position on the message.
2. It's value is expressed in degrees ranging from 0 to 360

One more time the encoder used was a Scalar encoder, since the goal of adding the COG to the input is to improve the categorization of trajectories it is important for the encoder to have a relatively big radius, this means the ranges of values which should share a lot of meaning need to be reasonable to allow the generalization of certain directions. The radius used on this encoder was of 15 which meant there would be in general 12, $(360/15 = 12)$, directions. This number was simply chosen considering that the general difference between tracks directions couldn't be that big to model a particular trajectory while there shouldn't be that many directions, this represents the fact that if a track is 15 degrees off from all the expectations that it should be considered anomalous. One difference in this encoder was that it was a periodic encoder since the meaning between 360 and 0 is the same, the encoder needs to encode that meaning in the SDR. The final encoder parameters are:

- periodic scalar encoder
- $w = 21$
- $radius = 15$
- $min = 0$ & $max = 360$

The results of adding this encoder to the initial system were the expected improvement on the general average anomaly which went down, this stands true since the majority of the data is expected to be detected as normal. On the other hand there was no actual improvement on the anomaly detection capabilities, while this is true this evaluation is based on a specific dataset of anomalous tracks which could not be very susceptible to this input. If possible further tests should be done with other datasets and which of course is true for other results.

4.3.3 Time Encoder

As discussed before sometimes having some measure of time as an input in the system may aid on the modelling of time related patterns. On the majority of the vessel databases evaluated that wasn't the case since what generally could be considered a time related pattern wasn't trajectories but the level of activities in particular maritime zones. Still, on the case of fishing vessels there was the possibility of some time related patterns, with this possibility in mind an encoder for time was added to the system to try and see if the results could present any particularly interesting anomalies.

In 4.2.2 was identified the possibility of making use of information related to the day of week to improve the model, more objectively the necessary information can be reduced to a simple category encoder with information pertaining to the current day classification between weekday or weekend. The encoder used was exactly a weekend encoder, a subclass of date encoders with

the previous characteristics, for this encoder the only needed parameter was the width, w, in which case $w = 21$ was used again since for previous encoders it worked well.

After application of this encoder the results as expected didn't improve, no actual anomalous behaviours correlated with this information could be regarded as real. While it is disappointing these were the expected results since in the previous analysis it was already noted that most likely the related patterns were more on activity levels then on trajectories, of course, it isn't meaningless since now we can be more clear on the real impact of this input. The results observed were a decrease in the average anomaly in general without any improvement on the detection of anomalous behaviour. This is expected since there's no real pattern which relate to this input and after enough learning the only result should be that there are similar trajectories for both weekdays and weekends, like this in most cases the prediction will be right on the first or second waypoint of a trajectory and from then on the next prediction should be generally right for this input since a step of context should be enough to know what the current day was. Basically in most cases the predictions related with this input and which will be right result in the decreased average anomaly.

4.3.4 Distance Encoders

After discussing time related patterns and while looking into information to improve the model the idea of using the space surged. Of course the system already uses time and space information since by using the HTM sequence memory the model being made describes successive positions time related. To make use of different information to characterize the trajectories the idea is to use distance. Three different distances were used in experiences:

1. distanceDelta meaning distance to last known position - this information is expected to help on the identification of vessels which stopped the AIS system during some time or even on the verification of the position and speed informations since distance depends on both and if both are normal for some trajectory then the distance also should be normal.

2. distanceFromSog0 meaning distance from track start - the distance between the start point coordinates and the current position - generally the track start was given by entry on the maritime space or by a stop point when the vessel SOG was 0. With this input it should be easier to detect anomalous transitions between trajectories, e.g. if a vessel goes on one trajectory and changes into a different one only during the transition could this behaviour be detected, after enough context which could be as little as a single waypoint the change in trajectory happened and the new one could be predicted without further anomalies.

3. distanceFromSequenceStartDelta meaning the difference between:
   - distance from track start calculated using distance between current and start position
   - distance from track start by accumulating distance between consecutive positions
   the goal of this information is to help characterize trajectories where the course isn't well defined, i.e. the objective isn't in getting to some specific place. This idea had as focus the fishing ships which have a very random travel course but on specific zones, for example,

lots of fishing vessels trajectories start by travelling some distance afar of the pier, then just go around fishing and later get back to the same pier. This example can be described by the continuous increase of the distance from start point which is then almost maintained followed by the decreasing distance from track start.

All these distance encoders share some characteristics that come from the features of distances data types. Distance is a counter of space accumulated since something, in this case since other position, this is important since because it's an accumulation the error between tracks distances will accumulate differently and for bigger distances the expected error will be bigger. Without taking this into account it's very difficult to use distance metrics to characterize trajectories. To solve this the encoder used on distances is a Log Encoder which as seen before is able to provide various ranges with varying resolutions which is what we need here, i.e. for bigger distance values the differences between tracks can be wider which means resolution should decrease with the distance.

The resulting encoders were respectively:

1. Log Encoder, $w = 21$, $min = 1$, $max = 100$, $radius = 0.3$
2. Log Encoder, $w = 21$, $min = 10$, $max = 1000$, $radius = 0.2$
3. Log Encoder, $w = 21$, $min = 10$, $max = 6000$, $radius = 0.1$

The results obtained with distanceDelta and distanceFromSog0 weren't very conclusive but at least to the subset of anomalous behaviours used there was no improvement on the detection of anomalies while the average anomaly distribution increased a little further decreasing the anomaly detection. On the other hand distanceFromSequenceStartDelta actually didn't have any major change on the results which was accepted as a good result since at least it meant this information was able to characterize the trajectories correctly while there was the possibility of using it to aid on the identification of the related anomalous behaviour previously identified noting that the performance of the detection is dependent on the subset of anomalous data but the ability to represent the trajectories affects the overall model.

4.3.5 Delta Encoders

Other then the previously mentioned encoders to add previous data to input space some experiments with Delta Encoders were also performed. Delta encoders are specific encoders which instead of using values provided as the base to the encoding of meaningful information into an SDR make use of the difference between consecutive values. Delta encoders were used in trials with the SOG, COG and distanceDelta and none of the cases the results were relevant to the improvement of the system which in general is normal since none of these types of data change rate should show better results in characterizing a trajectory. Still it was possible to observe some hints about it's usability in the results, for example, on the model using a Delta encoder with the SOG input it was very easy to identify zones of abrupt speed changes which were in general classified as abnormal since in these points spikes are detected.

4.4 Test List

From the initial simple system v1.0 with only a geospatial encoder and until the best fitting system settings are found lots of work was needed to better understand which information could improve the system, how to provide the system with that information and so on. In the next table a brief summary of the different tests from which data was stored along this project is presented with a brief description.

Table 4.1: Experimental system tests with settings summary description

Test ID	Changes Description	Based on Test
v10-10000	Geospatial Encoder scale = 10000	
v10-2000	Geospatial Encoder scale = 2000	
v10-5000	Geospatial Encoder scale = 5000	
v11-5000	+ SOG Scalar Encoder w=5 n=35 min=0 max=102	v10-5000
v12-5000	+ Weekend Date Encoder w=21	v11-5000
v13-5000	+ SOG Scalar Encoder w=21 n=147 min=0 max=102	v10-5000
v14-5000	+ SOG Log Encoder w=21 radius=0.3 min=1 max=102	v10-5000
v14d-5000	SOG (Log -> Delta) Encoder	v14-5000
v15-5000	+ COG Scalar Encoder w=21 radius=15 min=0 max=360	v10-5000
v15d-5000	COG (Scalar -> Delta) Encoder	v15-5000
v16-5000	+ distanceDelta Log Encoder w=21 radius=0.3 min=0 max=102	v10-5000
v16d-5000	+ distanceDelta (Log -> Delta) Encoder	v16-5000
v17-5000	SOG Log Encoder && COG Scalar Encoder	v14-5000 + v15-5000
v17-2000	Geospatial Encoder scale=(5000 -> 2000)	v17-5000
v17-3000	Geospatial Encoder scale=(5000 -> 3000)	v17-5000
v17-4000	Geospatial Encoder scale=(5000 -> 4000)	v17-5000
v17a-5000	COG Log Encoder w=(21 -> 51)	v17-5000
v17b-5000	COG Log Encoder radius=(15 -> 5)	v17-5000
v17c-5000	COG Log Encoder radius=(15 -> 10)	v17-5000
v18-5000	- Geospatial Encoder	v17-5000
v19-5000	+ distanceFromSequenceStartDelta Log Encoder w=21 radius=0.1 min=10 max=6000	v17-5000
v20-5000	+ distanceFromSog0 Log Encoder w=21 radius=0.2 min=10 max=1000	v17-5000
v20b-5000	+ distanceFromSog0 Log Encoder w=21 radius=0.2 min=10 max=1000	v10-5000

Chapter 5

Conclusion

On this chapter we will go over the overall results obtained during all the tests performed to better understand the real improvements achieved as well as the real system results as the solution to the challenge of detecting anomalies in vessels behaviours.

5.1 Work Results Summary

Along the development of this dissertation and with the objective of solving the problem of detecting anomalous behaviours by making use of AIS data various decisions were made, the first one was to try to use an HTM theory based system to solve the problem. All along the final objective was the detection of anomalies which was taken as the string to guide all following decisions. Detecting anomalous behaviour implied the need to have a good model to describe our data so that it was possible to discern which behaviours are normal and which are anomalous. With that in mind all the work progressed as was explained in previous chapters until it was time to take this dissertation and project phase as finished and so it is now time to evaluate the progress achieved.

During all this project development decisions were made based on the analysis of data wether it was the raw data to create the model or results achieved by the different tests to try and keep improving, to better understand what was being modelled or how it was modelled and how that contributed to the final goals. Various tests using the previous developed system and encoders were made and identified on 4.1 and various results obtained which we will briefly go over next.

In Figure 5.1 it's possible to see the overall change on the anomaly score distribution for both an anomalous and a normal subset of the population. These kinds of charts when seen as a single one can be very interesting to observe overall results, for these charts the overall objectives while performing the tests were:

1. On the right group for the distribution of anomalies on the normal dataset the goal was to concentrate the majority of the population on the minimum of bars possible and of course on the lower side which would mean that most points should have very low anomalies scores and that the normal population was being well identified.

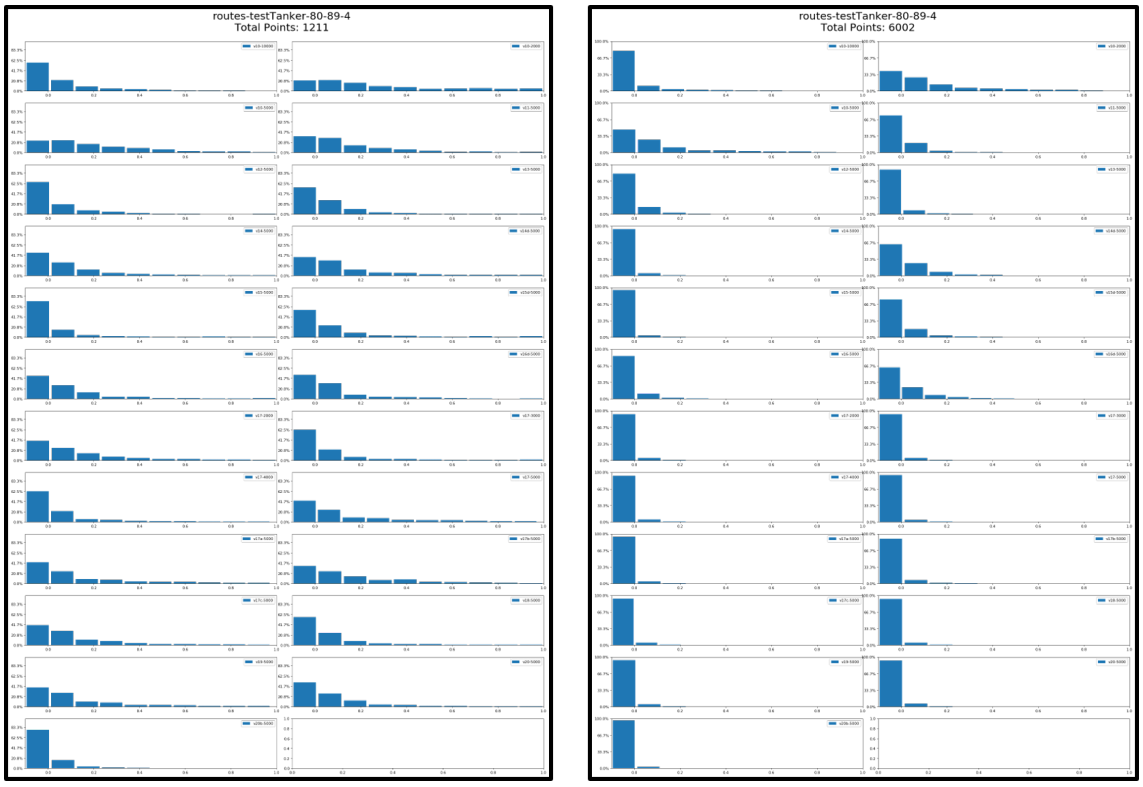


Figure 5.1: Histogram depicting anomaly distribution for all tests on subsets of anomalous (left) and normal data (right)

2. For the left group the goals were the other way around meaning that most distribution should go to the higher side with anomaly scores being the higher the better.

3. Of course good results can only be achieved if a good compromise between 1 and 2 are achieved, in practice if the results on the right show that almost 100% of the population was on the first column than it means that all points within other columns can be considered anomalous. On the other hand if the distribution is all over then it's very difficult to infer which values correspond to anomalies.

In reality the previously described anomaly likelihood makes use of this way of thinking to improve the anomaly detection performance by comparing the latest subset of results to the overall results history and computing a probability that the current results are anomalous. On Figure 5.2 we can see the corresponding anomaly likelihood distribution, for the likelihood and derivative to the way it was computed the values are ranged between 0.5 and 1 instead of 0 and 1 like on the anomaly score case, it's also important to note that in general it was expected the need of using a very high value as a threshold as described in Ahmad *et al.* (2017, [2]) in which $\varepsilon = 10^{-5}$ was considered a good general threshold value to separate the occurrence of anomalies in a wide range of domains. On this project that value which would mean having anomalies only detected when the anomaly likelihood was over 0.99999 ($L_t \geq 1 - \varepsilon \Leftrightarrow L_t \geq 0.99999$) didn't really fit and so a domain-specific one was found and which we will briefly go over later. On this Figure it is quite

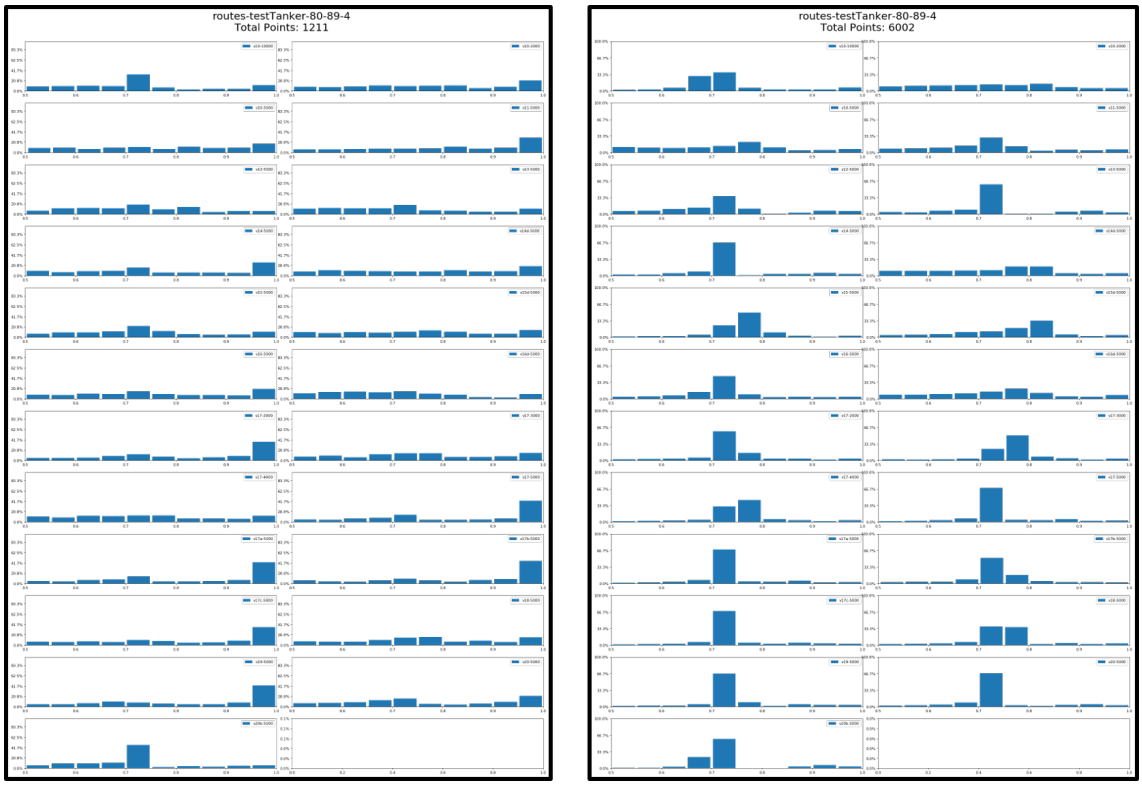


Figure 5.2: Histogram depicting anomaly likelihood distribution for all tests on subsets of anomalous (left) and normal data (right)

easy to understand the overall improvement since we can observe that in later tests the anomaly likelihood distribution for both subsets is generally quite well defined:

- For the anomalous subset distribution it's possible to observe various tests where the values are mostly distributed close to 1;
- On the normal subset distribution it's even more clear to see that from the early tests (top distributions) on there's a general improvement with most values being on the central column between 0.7 and 0.75;
- With both populations well defined the tests should show large improvements on the ability to detect anomalies. On the other hand there's also later tests (close to the bottom) where it's possible to identify the decrease on the anomalous population characterization which are generally indicative of mostly failed tests where some change was introduced and proved armful to the overall results.

On all these charts it was possible to observe the general qualitative improvement on the different tests and to understand even better the results, to better compare them and to really quantify it a measure of the tests accuracy was needed. The measure of choice was the F-measure (Van Rijsbergen, 1979, [23]) generally used on the statistical analysis of binary classification problems which in this case is the classification between normal and anomalous.

testsID	true+Rate	true-Rate	false+Rate	false-Rate	+Likelihoo	-Likelihoo	diagnostic	precision	F1Score	FbScore
v10-10000	7.1	97.6	2.4	92.9	3.0	1.0	3.1	37.6	11.9	20.2
v10-2000	10.6	99.0	1.0	89.4	10.1	0.9	11.1	67.0	18.3	32.4
v10-5000	6.6	98.0	2.0	93.4	3.4	1.0	3.5	40.4	11.4	20.0
v11-5000	12.4	97.2	2.8	87.6	4.5	0.9	4.9	47.3	19.6	30.3
v12-5000	3.3	97.1	2.9	96.7	1.2	1.0	1.2	18.9	5.6	9.7
v13-5000	4.1	97.9	2.1	95.9	1.9	1.0	2.0	28.1	7.2	13.0
v14-5000	14.3	98.0	2.0	85.7	7.0	0.9	8.0	58.4	23.0	36.1
v14d-5000	6.7	97.7	2.3	93.3	2.9	1.0	3.0	36.7	11.3	19.3
v15-5000	3.5	98.2	1.8	96.5	1.9	1.0	1.9	27.5	6.2	11.5
v15d-5000	5.8	98.8	1.2	94.2	4.8	1.0	5.0	49.0	10.3	19.6
v16-5000	7.9	97.6	2.4	92.1	3.2	0.9	3.4	39.5	13.2	22.0
v16d-5000	5.0	96.7	3.3	95.0	1.5	1.0	1.5	23.5	8.3	13.5
v17-2000	17.6	97.8	2.2	82.4	7.9	0.8	9.4	61.6	27.4	41.0
v17-3000	5.5	97.9	2.1	94.5	2.6	1.0	2.7	34.5	9.5	16.9
v17-4000	3.9	97.7	2.3	96.1	1.7	1.0	1.7	25.5	6.7	12.1
v17-5000	27.2	98.5	1.5	72.8	18.1	0.7	24.5	78.5	40.4	57.0
v17a-5000	27.2	98.5	1.5	72.8	18.1	0.7	24.5	78.5	40.4	57.0
v17b-5000	26.8	98.4	1.6	73.2	16.6	0.7	22.3	77.0	39.8	56.1
v17c-5000	20.4	98.2	1.8	79.6	11.2	0.8	13.9	69.4	31.5	46.9
v18-5000	9.2	97.7	2.3	90.8	4.0	0.9	4.3	44.8	15.2	25.2
v19-5000	28.7	98.0	2.0	71.3	14.4	0.7	19.8	74.4	41.5	56.4
v20-5000	12.7	98.2	1.8	87.3	7.1	0.9	8.0	58.8	20.9	34.1
v20b-5000	2.1	98.1	1.9	97.9	1.1	1.0	1.1	17.7	3.7	7.0

Figure 5.3: Table with quantitative results for all tests; F_β score with $\beta = 0.5$ and threshold $L_t > 0.99$

The F-measure is the harmonic mean between precision and recall and basically it takes into consideration:

- Recall: The number of relevant items which are selected;
- Precision: The number of selected items which are relevant;

where for this project the relevant items are all anomalous waypoints. The formula correspondent to the harmonic mean or $F_1 score$ is:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.1}$$

where:

$$Precision = \frac{truepositive}{truepositive + falsepositive}; \quad Recall = \frac{truepositive}{truepositive + falsenegative} \tag{5.2}$$

The general formula can be expressed as:

$$F_\beta = \frac{(1+\beta^2) \cdot truepositive}{(1+\beta^2) \cdot truepositive + \beta^2 \cdot falsenegative + falsepositive} \tag{5.3}$$

and where F_β measures the classification effectiveness and attaches β times as much importance to recall as precision (Van Rijsbergen, 1979, [23]).

On Figure 5.3 we can see various quantitative results for all tests. These results were obtained using the subsets of anomalous and normal behaviour and in which case all points of each subset was considered to have as true classification the overall set classification. On this table it's possible to identify various results related with every test including:

- True Positive Rate, True Negative Rate, False Positive Rate and False Negative Rate
- Positive Likelihood and Negative Likelihood

Parameter	Max	Threshold	testsID	true+Rate	true-Rate	false+Rate	false-Rate	+Likelihoo	-Likelihoo	diagnostic	precision	F1Score	FbScore
true+Rate	87.2	0.79_aS+aL	v10-2000	87.2	23.6	76.4	12.8	1.1	0.5	2.1	18.7	30.8	22.2
true-Rate	99.9	0.98_aSxaL	v17-5000	24.9	99.9	0.1	75.1	165.8	0.8	220.3	97.1	39.6	61.4
false+Rate	76.4	0.79_aS+aL	v10-2000	87.2	23.6	76.4	12.8	1.1	0.5	2.1	18.7	30.8	22.2
false-Rate	99.3	0.99_aSxaL	v20b-5000	0.7	99.7	0.3	99.3	2.5	1.0	2.5	33.3	1.5	3.4
+Likelihoo	259.7	0.99_aSxaL	v17-5000	21.6	99.9	0.1	78.4	259.7	0.8	331.1	98.1	35.5	57.5
-Likelihoo	1.2	0.79_aL	v15d-5000	41.3	50.7	49.3	58.7	0.8	1.2	0.7	14.5	21.4	16.6
diagnostic	331.1	0.99_aSxaL	v17-5000	21.6	99.9	0.1	78.4	259.7	0.8	331.1	98.1	35.5	57.5
precision	98.1	0.99_aSxaL	v17-5000	21.6	99.9	0.1	78.4	259.7	0.8	331.1	98.1	35.5	57.5
F1Score	65.9	0.93_1S+aL	v17b-5000	63.5	94.1	5.9	36.5	10.8	0.4	27.8	68.5	65.9	67.4
FbScore	73.6	0.86_aSxaL	v17b-5000	46.6	98.5	1.5	53.4	30.7	0.5	56.6	86.1	60.5	73.6

Figure 5.4: Table with quantitative results for all parameters maximum value, meaning best or worst value depending on the parameter type; the tie-breaker was the F_β Score so that it is the maximum with the better overall results

- Diagnostic
- Precision
- F_1 Score and F_β Score

The measure which was taken as the deciding factor on choosing the best results was the F_β Score, this was the one measure which could best reflect the test performance. The use of the F_β Score instead of the F_1 Score is due to the fact that in our data the expected ratio of negative classification is much higher than the rate of positive, with this by using a $\beta \leq 1$ the effect of false negatives was attenuated which resulted on choices for better results with higher true negative rates with some cost on the true positive rate. The value for β was of 0.5 and was the value used on all specifications of F_β Score.

To find the better options to optimize results various settings were used:

- Thresholding using:
  1. Anomaly Score
  2. Anomaly Likelihood
  3. Anomaly Score AND Anomaly Likelihood
  4. Anomaly Score OR Anomaly Likelihood
- Thresholds for:
  1. Anomaly Score: 0 or 0.1
  2. Anomaly Likelihood: ranging between 0.79 and 0.99

In Figure 5.4 are presented the overall maximum results obtained after running with all different parameters, the results represent the maximum parameter across all settings and with the best

overall performance decided by using the F_β Score as tie-breaker. On the threshold cells are all the information necessary to know which threshold configurations were used for that result, the information is composed by:

- <Threshold>_<AnomalyType> - with anomaly type being aL,aS, respectively anomaly Likelihood and Score
- <AnomalyLikelihoodThreshold>_<AnomalyType><Logic><AnomalyType> - with anomaly type being aL,aS,1S, respectively anomaly Likelihood, anomaly Score (with threshold 0) and anomaly Score (with threshold 0.1); with logic being x,+, respectively Logical AND and OR

We can observe some very good results like:

- True negative rate of 99.9% while still being able to detect almost 25% of the anomalous waypoints. This test has the ID v17-5000 and uses as threshold for anomalies:
  $L_t > 0.98 \ \vee \ S_t > 0$
- The best test given by the maximum F_1 Score has a true negative rate of 94.1% and is able to detect almost 65% of the anomalous waypoints. This test has the ID v17b-5000 and uses as threshold for anomalies:
  $L_t > 0.93 \ \wedge \ S_t > 0.1$
- The best overall test is given by the maximum F_β Score and has a true negative rate of 98.5% while still being able to detect almost 50% of the anomalous waypoints. This test has the ID v17b-5000 and uses as threshold for anomalies:
  $L_t > 0.86 \ \vee \ S_t > 0$

From these it's possible to conclude that the best system settings were the variations on the tests v17 with the general best balanced results which means that for modelling the trajectories correspondent with these results which were from the database of vessels of type tanker the best settings were the ones from the test 17b-5000 which means:

- Geospatial Encoder scale = 5000
- SOG Log Encoder w=21 radius=0.3 min=1 max=102
- COG Periodic Scalar Encoder w=21 radius=5 min=0 max=360

Finally in Figure 5.5 where the best result obtained by each test is presented the previous statement is one more time confirmed with the top results being achieved by the various v17 variations. It's also possible to conclude that the various tests improved the overall anomaly detection performance as well as better understand and compare the impact of the addition of new inputs or encoders changes on the system performance mentioned on the previous chapter. Lastly a note on the importance of the threshold method used to decide the anomalous behaviour classification

testsID	threshold	true+Rate	true-Rate	false+Rate	false-Rate	+Likelihoo	-Likelihoo	diagnostic	precision	F1Score	FbScore
v10-10000	0.0_aS	43.2	80.6	19.4	56.8	2.2	0.7	3.2	31.0	36.1	32.9
v10-2000	0.97_aSxaL	16.7	97.2	2.8	83.3	5.9	0.9	6.9	54.3	25.5	37.4
v10-5000	0.83_aSxaL	36.0	85.6	14.4	64.0	2.5	0.7	3.4	33.6	34.7	34.0
v11-5000	0.79_aSxaL	44.5	93.1	6.9	55.5	6.4	0.6	10.8	56.5	49.8	53.6
v12-5000	0.08_aS	18.2	94.6	5.4	81.8	3.4	0.9	3.9	40.5	25.1	32.5
v13-5000	0.03_aS	33.6	94.5	5.5	66.4	6.1	0.7	8.8	55.4	41.8	49.0
v14-5000	0.03_aS	42.1	96.3	3.7	57.9	11.4	0.6	19.0	69.8	52.5	61.7
v14d-5000	0.82_aSxaL	30.6	91.9	8.1	69.4	3.8	0.8	5.0	43.1	35.7	39.8
v15-5000	0.91_aS+aL	35.9	92.3	7.7	64.1	4.7	0.7	6.7	48.4	41.3	45.3
v15d-5000	0.89_1S+aL	38.0	85.6	14.4	62.0	2.6	0.7	3.6	34.7	36.3	35.3
v16-5000	0.06_aS	36.6	94.1	5.9	63.4	6.2	0.7	9.2	55.6	44.1	50.4
v16d-5000	0.08_aS	24.8	86.2	13.8	75.2	1.8	0.9	2.0	26.5	25.6	26.2
v17-2000	0.8_aSxaL	43.0	98.0	2.0	57.0	21.2	0.6	36.4	81.0	56.2	68.9
v17-3000	0.0_aS	38.4	93.3	6.7	61.6	5.7	0.7	8.6	53.4	44.7	49.6
v17-4000	0.0_aS	38.6	93.1	6.9	61.4	5.6	0.7	8.4	52.9	44.6	49.2
v17-5000	0.87_aSxaL	39.7	98.9	1.1	60.3	35.1	0.6	57.5	87.6	54.7	70.6
v17a-5000	0.87_aSxaL	39.7	98.9	1.1	60.3	35.1	0.6	57.5	87.6	54.7	70.6
v17b-5000	0.86_aSxaL	46.6	98.5	1.5	53.4	30.7	0.5	56.6	86.1	60.5	73.6
v17c-5000	0.79_aSxaL	40.1	98.0	2.0	59.9	19.9	0.6	32.6	80.1	53.5	66.8
v18-5000	0.0_aS	43.6	93.1	6.9	56.4	6.3	0.6	10.5	56.1	49.1	53.1
v19-5000	0.9_aSxaL	38.2	98.6	1.4	61.8	26.4	0.6	42.1	84.2	52.6	67.9
v20-5000	0.06_aS	32.9	97.8	2.2	67.1	14.9	0.7	21.7	75.0	45.8	59.7
v20b-5000	0.0_aS	22.9	96.3	3.7	77.1	6.1	0.8	7.7	55.3	32.4	43.1

Figure 5.5: Table with results for all tests overall best result given by the best F_β Score

which as can be seen on the Figure there wasn't a particular threshold method which performed better on all tests but the majority of the best performing tests used as threshold a combination of both the anomaly score and the anomaly likelihood.

5.2 Contributions

All along goals of this dissertation were to verify the feasibility of using an HTM based system to perform anomaly detection on maritime vessels and better understand how and how well could an HTM based system model maritime vessel trajectories using AIS data. With the end of the project there are various contributions to make and which can be helpful on the choice and design of an HTM based system, respectively:

- HTM based system can achieve very good results on the modelling of data which has a temporal sequence to it and even if the rate of sequent data is limited it can perform reasonably with the right parameters. On this project the initial system makes use of positions temporally spaced by one hour and with the right scale the system models the main trajectories pretty accurately.

- Trajectories can be modelled using simply GPS coordinates where generalization is obtained out of the box by the fact that HTM makes use of SDRs to hold information which perform very well on the task.

- To improve trajectories characterization the addition of other inputs data can be a very simple solution. By using speed and movement directionality which could also be directly extracted by using pairs of positions, even if not provided by measuring devices, which was the case in this project where the data was provided on the AIS position messages by using vessel measuring devices, it was possible to improve the overall trajectory model and with that the anomaly detection effectiveness. On the other hand it is important to note that

adding data into a model can very easily negatively affect the model predictive ability which will strongly affect the anomaly detection performance and so the choice of data to add into a HTM model for anomaly detection has to be made carefully and take into account that it should contain meaningful information which could characterize the model by it's sequence instead of it's instantaneous values, i.e. data with no natural sequence patterns won't be predictable.

- The anomaly detection on HTM based systems is by design very natural since the model is predictive. For specific domains the exploitation of different thresholds limits and modes of operation proved by the results of different tests can improve system performance on the anomaly detection vector.

5.3 Limitations and Future Improvements

The final system while having a much better performance than earlier versions is still quite limited. It's ability to model trajectories is dependent on the trajectories courses and if these courses are very difficult to generalize and predict the anomaly detection performance will decrease sharply. The tests using distance to the start point for example try to make use of a different trajectory characteristic to model it, the results weren't particularly interesting but it's a possible way to improve on this limitation.

Other limitation of the system is parameter tuning which since the production of a model with relevant results implies the need to process large amounts of data which with limited processing power and time constraints puts a difficult barrier on the possibility of optimizing the parameters. On the other hand this limitation isn't as important on a deployed system since there the model while doing it in real time will only need to process data on a much lower rate that the current tests which use a batch of months of data instead of stream of AIS data. There's also an important limitation which would create the need to redesign the way the inputs are fed to the system in a real time environment since to model trajectories the data is divided in tracks which have a sequence rationale to them, this is an indispensable characteristic since if the data is fed without this rationale it's impossible for the system to learn sequences, context or make predictions based on it. On a real time environment the model may have the sequences learned by using batches of tracks but it will still need context to understand the new input, this will not happen naturally with the data stream from an AIS system which will receive messages from all vessels in it's disordered way and would need to be solved by providing a number of historic data for each vessel point previous positions so that the system computes the anomaly likelihood of the current point based on the historic provided.

The current system uses a single HTM model for all trajectories being only separated by vessel type which can have high performance and efficiency costs, it would be interesting to use multiple models instead. The same way different models were used for different vessel types the same logic could be applied for other characteristics, for example:

- Model different geographical zones separately by dividing the maritime zone into a grid of zones each with a particular model which could be more accurate on it's trajectories characterization.
- Create different models for different tracks average velocity ranges.

These were just some possible limitations and improvements identified which could be reserved for future work and of course there's also the possibility to keep doing the same work of analysing other data for input in the model or optimizing the current encoders taking into account specific vessel types for example.

Appendix A

Statistics Sample Results of Tests Data Model

In this appendix are presented further statistic examples about the test results which were used during the result analysis of each test. Some of which were presented before with lower resolution and are present here again for reference with higher resolution to provide a better understanding of the interpretations and decisions made along the project. The database of tanker vessels with about 461000 waypoints was the one chosen for these examples since this was the one in which the later part of the project had it's decisions based.

These include:

- Anomaly Distribution
- Zero Anomaly Score Rate
- Anomalous Rate
- Anomaly detection results over different thresholds

A.1 Graphics and Tests Data

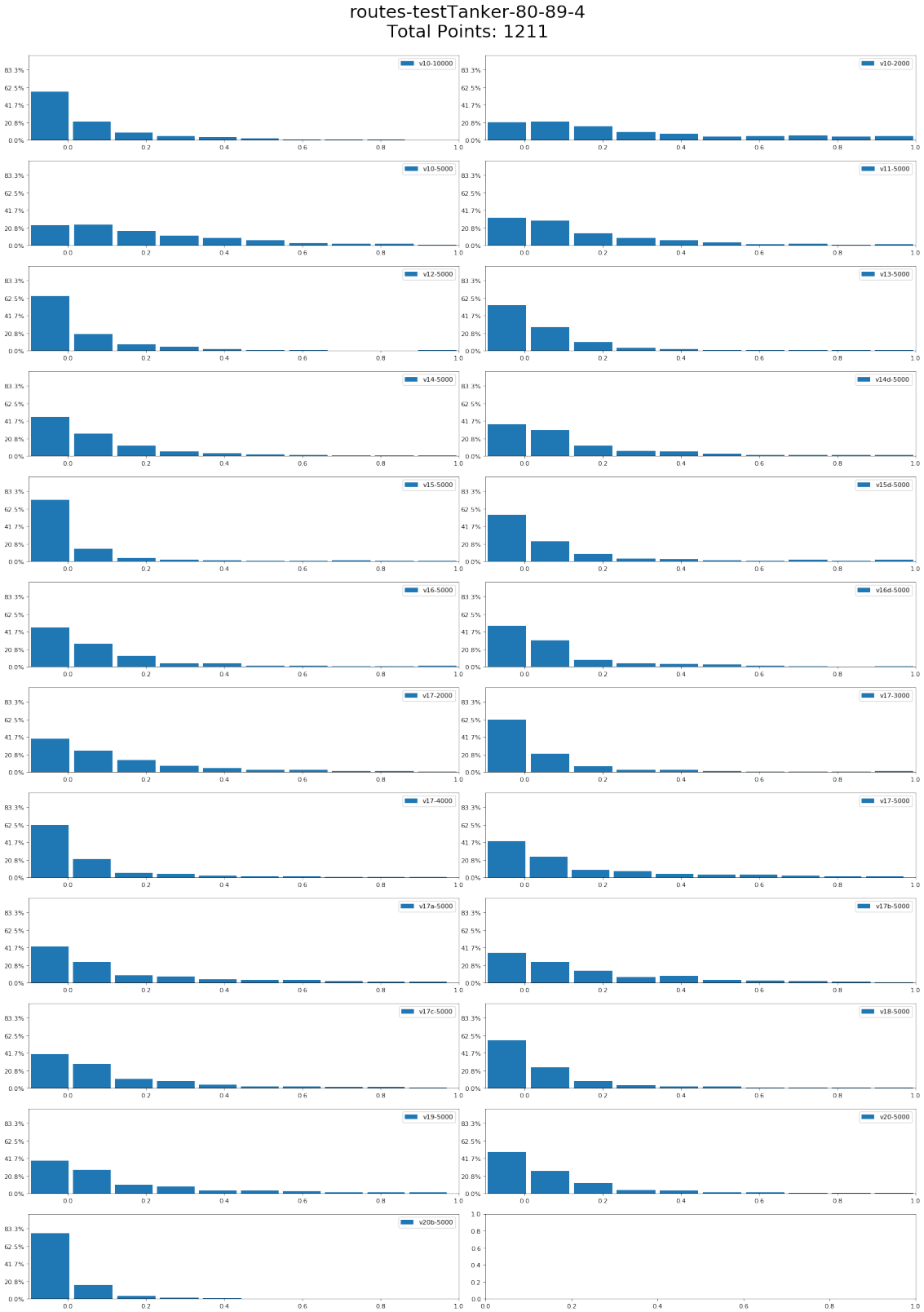


Figure A.1: Anomaly Score Distribution for anomalous subset.
Note: Scores of 0 were changed to negative so that it's possible to easily separate them on the distribution

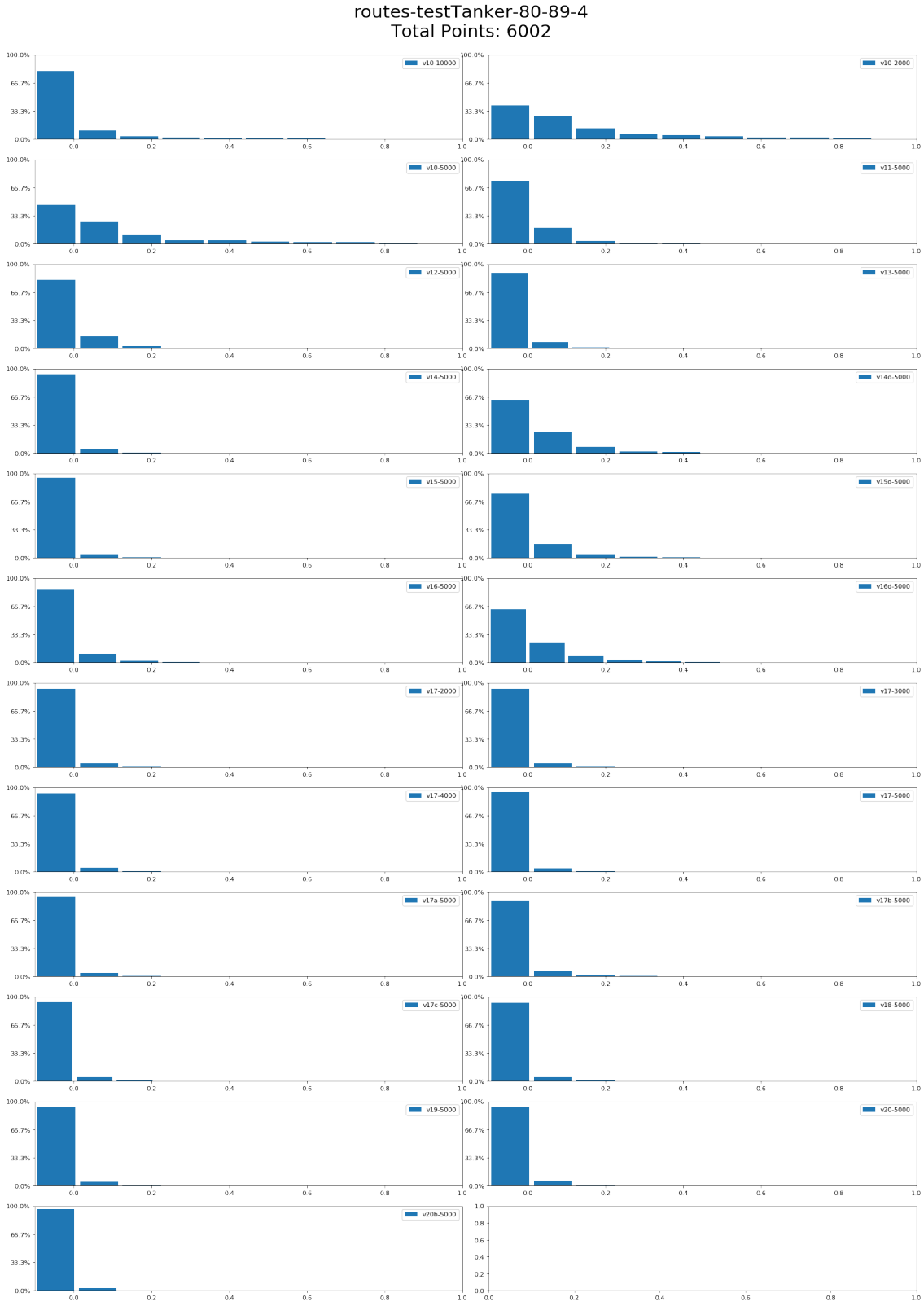


Figure A.2: Anomaly Score Distribution for normal subset.
Note: Scores of 0 were changed to negative so that it's possible to easily separate them on the distribution

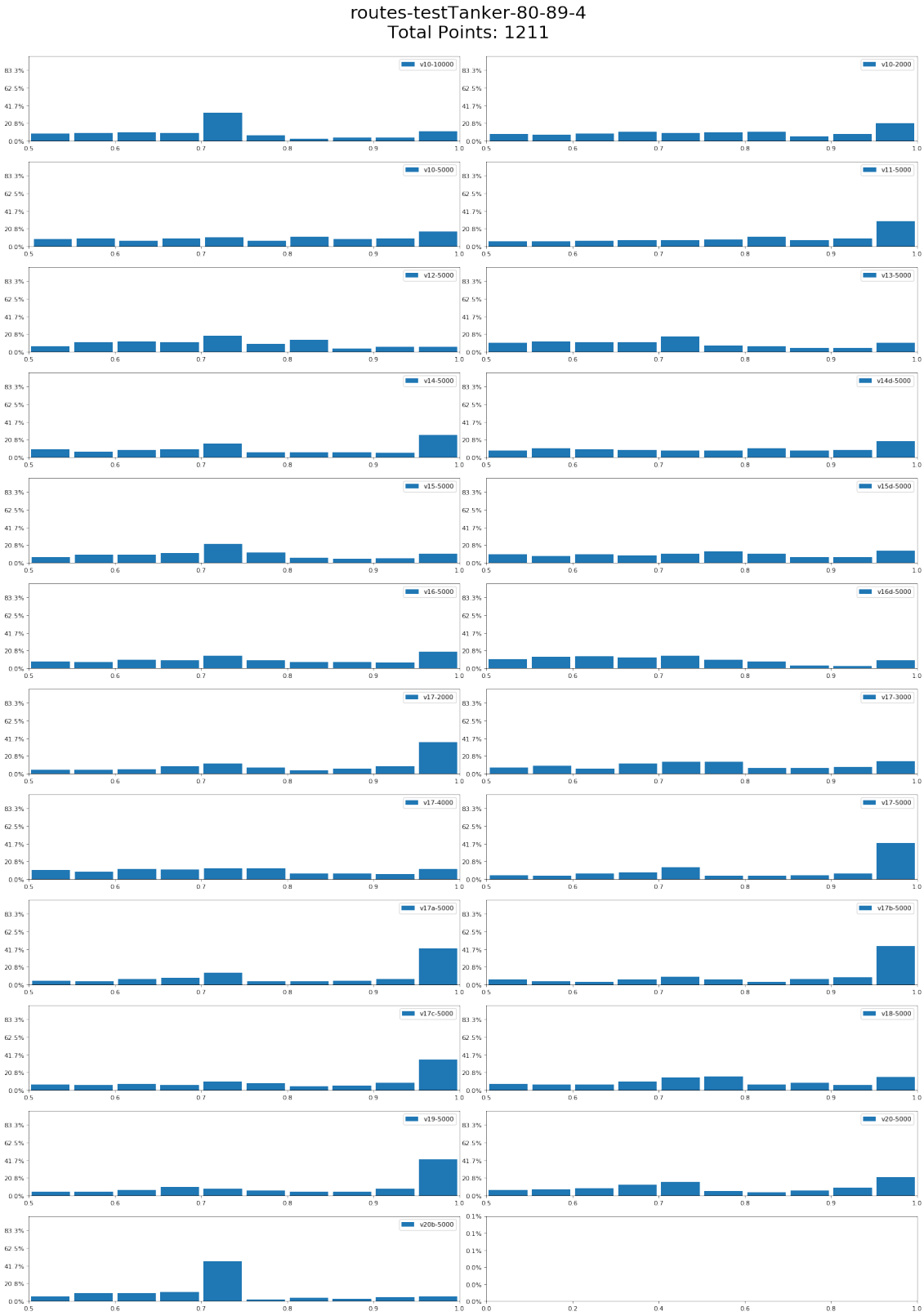


Figure A.3: Anomaly Likelihood Distribution for anomalous subset

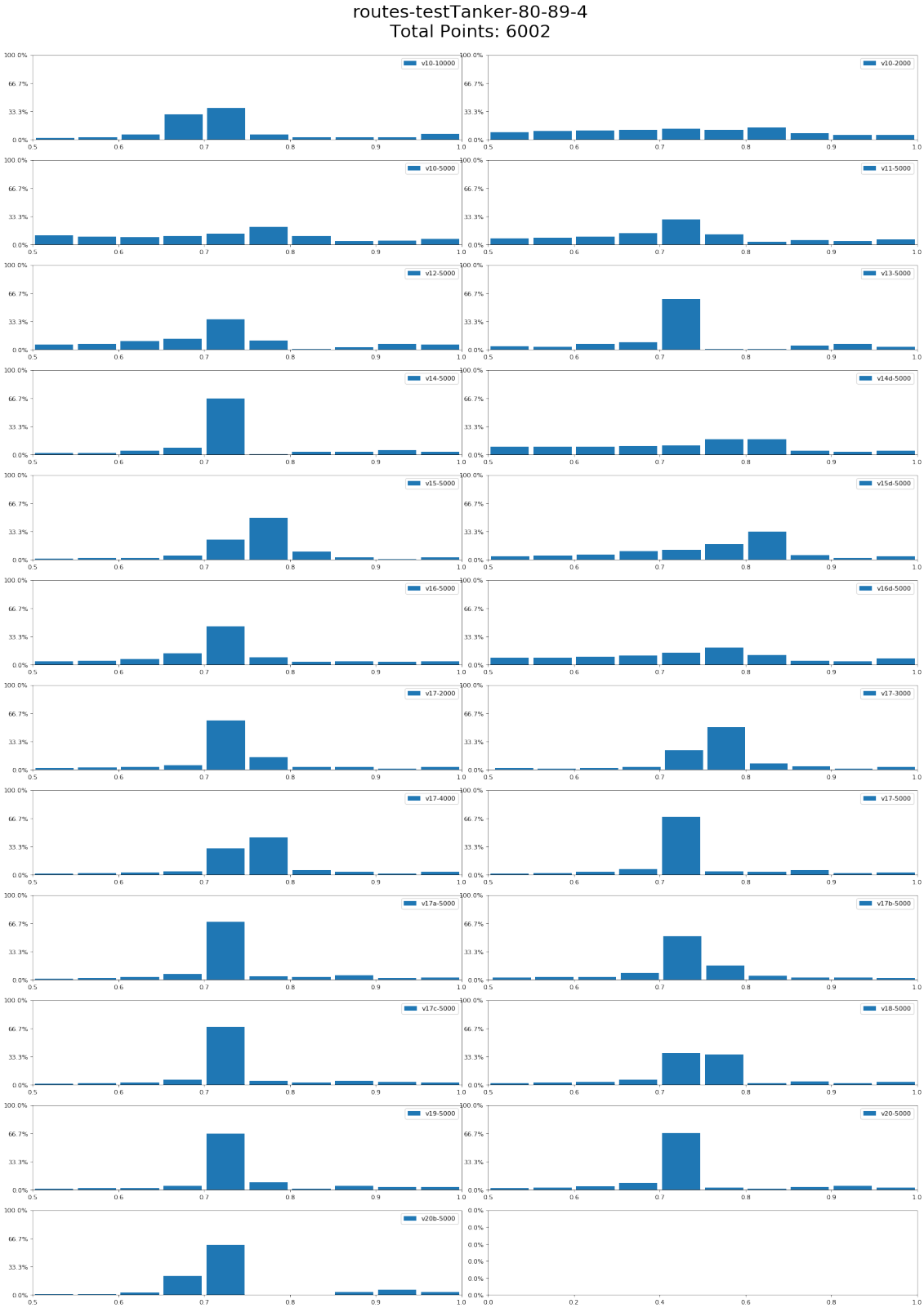
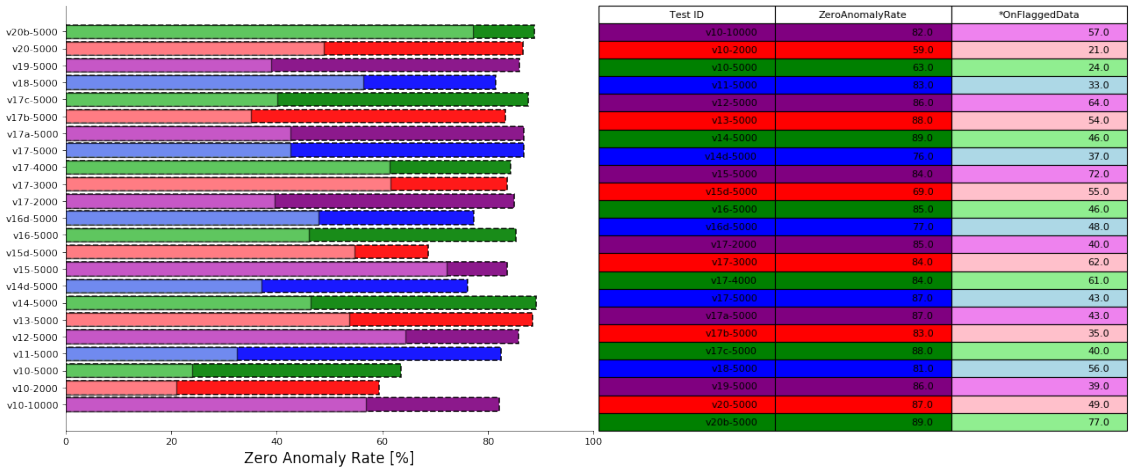
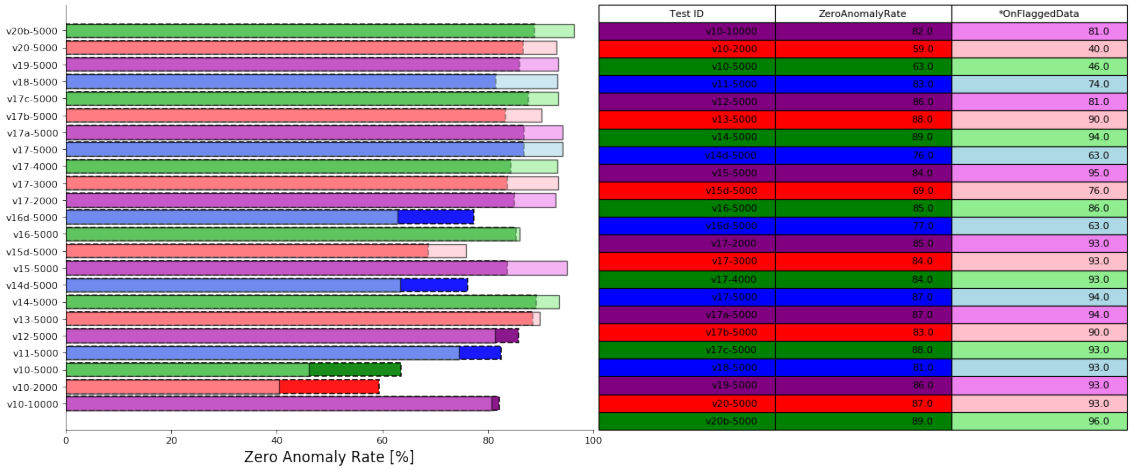


Figure A.4: Anomaly Likelihood Distribution for normal subset



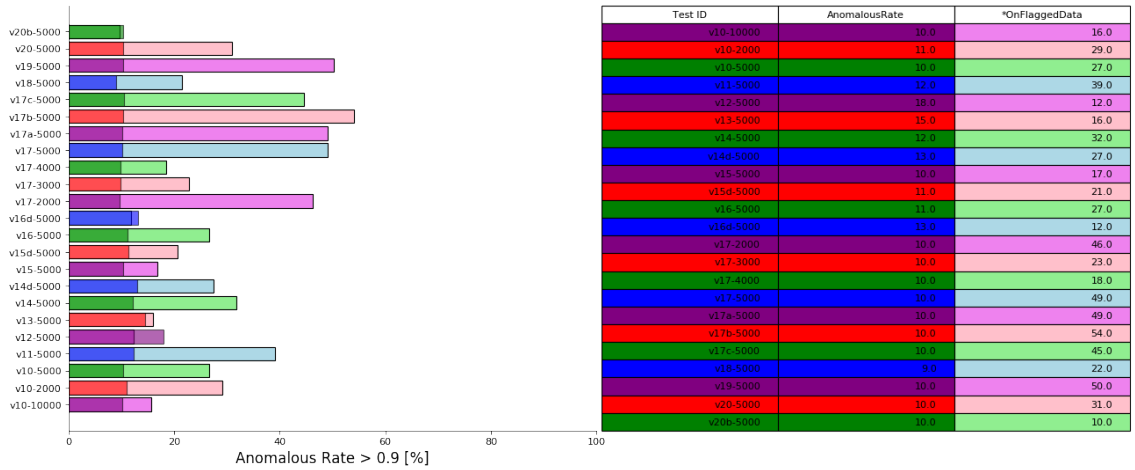
Test ID	ZeroAnomalyRate	*OnFlaggedData
v10-10000	82.0	57.0
v10-2000	59.0	21.0
v10-5000	63.0	24.0
v11-5000	83.0	33.0
v12-5000	86.0	64.0
v13-5000	88.0	54.0
v14-5000	89.0	46.0
v14d-5000	76.0	37.0
v15-5000	84.0	72.0
v15d-5000	69.0	55.0
v16-5000	85.0	46.0
v16d-5000	77.0	48.0
v17-2000	85.0	40.0
v17-3000	84.0	62.0
v17-4000	84.0	61.0
v17-5000	87.0	43.0
v17a-5000	87.0	43.0
v17b-5000	83.0	35.0
v17c-5000	88.0	40.0
v18-5000	81.0	56.0
v19-5000	86.0	39.0
v20-5000	87.0	49.0
v20b-5000	89.0	77.0

Figure A.5: Rate of anomaly scores equal to 0 for each test with both the rate for the last month of data (where the model is already reliable) and for the subset of anomalous flagged data



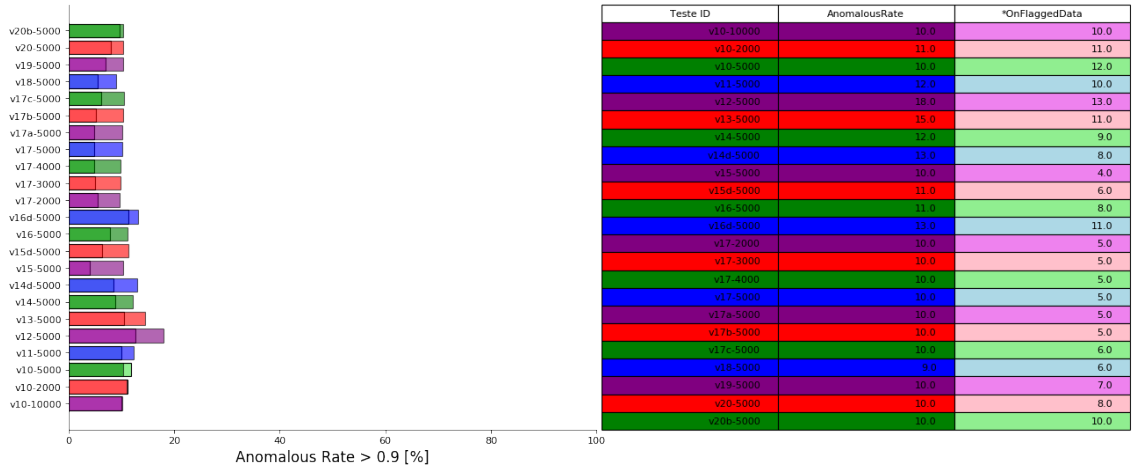
Test ID	ZeroAnomalyRate	*OnFlaggedData
v10-10000	82.0	81.0
v10-2000	59.0	40.0
v10-5000	63.0	46.0
v11-5000	83.0	74.0
v12-5000	86.0	81.0
v13-5000	88.0	90.0
v14-5000	89.0	94.0
v14d-5000	76.0	63.0
v15-5000	84.0	95.0
v15d-5000	69.0	76.0
v16-5000	85.0	86.0
v16d-5000	77.0	63.0
v17-2000	85.0	93.0
v17-3000	84.0	93.0
v17-4000	84.0	93.0
v17-5000	87.0	94.0
v17a-5000	87.0	94.0
v17b-5000	83.0	90.0
v17c-5000	88.0	93.0
v18-5000	81.0	93.0
v19-5000	86.0	93.0
v20-5000	87.0	93.0
v20b-5000	89.0	96.0

Figure A.6: Rate of anomaly scores equal to 0 for each test with both the rate for the last month of data (where the model is already reliable) and for the subset of normalcy flagged data



Test ID	AnomalousRate	*OnFlaggedData
v10-10000	10.0	16.0
v10-2000	11.0	29.0
v10-5000	10.0	27.0
v11-5000	12.0	39.0
v12-5000	18.0	12.0
v13-5000	15.0	16.0
v14-5000	12.0	32.0
v14d-5000	13.0	27.0
v15-5000	10.0	17.0
v15d-5000	11.0	21.0
v16-5000	11.0	27.0
v16d-5000	13.0	12.0
v17-2000	10.0	46.0
v17-3000	10.0	23.0
v17-4000	10.0	18.0
v17-5000	10.0	49.0
v17a-5000	10.0	49.0
v17b-5000	10.0	54.0
v17c-5000	10.0	45.0
v18-5000	9.0	22.0
v19-5000	10.0	50.0
v20-5000	10.0	31.0
v20b-5000	10.0	10.0

Figure A.7: Rate of anomaly scores over 0.9 for each test with both the rate for the last month of data (where the model is already reliable) and for the subset of anomalous flagged data



Teste ID	AnomalousRate	*OnFlaggedData
v10-10000	10.0	10.0
v10-2000	11.0	11.0
v10-5000	10.0	12.0
v11-5000	12.0	10.0
v12-5000	18.0	13.0
v13-5000	15.0	11.0
v14-5000	12.0	9.0
v14d-5000	13.0	8.0
v15-5000	10.0	4.0
v15d-5000	11.0	6.0
v16-5000	11.0	8.0
v16d-5000	13.0	11.0
v17-2000	10.0	5.0
v17-3000	10.0	5.0
v17-4000	10.0	5.0
v17-5000	10.0	5.0
v17a-5000	10.0	5.0
v17b-5000	10.0	5.0
v17c-5000	10.0	6.0
v18-5000	9.0	6.0
v19-5000	10.0	7.0
v20-5000	10.0	8.0
v20b-5000	10.0	10.0

Figure A.8: Rate of anomaly scores over 0.9 for each test with both the rate for the last month of data (where the model is already reliable) and for the subset of normalcy flagged data

testsID	true+Rate	true-Rate	false+Rate	false-Rate	+Likelihoo	-Likelihoo	diagnostic	precision	F1Score	FbScore
v10-10000	43.2	80.6	19.4	56.8	2.2	0.7	3.2	31.0	36.1	32.9
v10-2000	79.0	40.4	59.6	21.0	1.3	0.5	2.6	21.1	33.3	24.7
v10-5000	76.1	46.1	53.9	23.9	1.4	0.5	2.7	22.2	34.3	25.8
v11-5000	67.5	74.5	25.5	32.5	2.6	0.4	6.1	34.8	45.9	38.5
v12-5000	35.7	81.3	18.7	64.3	1.9	0.8	2.4	27.8	31.2	29.1
v13-5000	46.2	89.9	10.1	53.8	4.6	0.6	7.6	47.9	47.1	47.6
v14-5000	53.5	93.5	6.5	46.5	8.3	0.5	16.6	62.5	57.7	60.5
v14d-5000	62.8	63.4	36.6	37.2	1.7	0.6	2.9	25.7	36.5	29.2
v15-5000	27.7	95.0	5.0	72.3	5.6	0.8	7.3	52.9	36.4	44.8
v15d-5000	45.3	75.9	24.1	54.7	1.9	0.7	2.6	27.5	34.2	29.8
v16-5000	53.9	86.0	14.0	46.1	3.9	0.5	7.2	43.8	48.3	45.5
v16d-5000	52.0	62.9	37.1	48.0	1.4	0.8	1.8	22.0	31.0	24.9
v17-2000	60.4	92.8	7.2	39.6	8.4	0.4	19.7	62.8	61.6	62.3
v17-3000	38.4	93.3	6.7	61.6	5.7	0.7	8.6	53.4	44.7	49.6
v17-4000	38.6	93.1	6.9	61.4	5.6	0.7	8.4	52.9	44.6	49.2
v17-5000	57.4	94.1	5.9	42.6	9.6	0.5	21.3	66.1	61.4	64.1
v17a-5000	57.4	94.1	5.9	42.6	9.6	0.5	21.3	66.1	61.4	64.1
v17b-5000	64.9	90.1	9.9	35.1	6.6	0.4	16.9	57.0	60.7	58.4
v17c-5000	59.9	93.4	6.6	40.1	9.0	0.4	20.9	64.5	62.1	63.5
v18-5000	43.6	93.1	6.9	56.4	6.3	0.6	10.5	56.1	49.1	53.1
v19-5000	61.1	93.3	6.7	38.9	9.1	0.4	21.9	64.8	62.9	64.0
v20-5000	51.1	92.9	7.1	48.9	7.2	0.5	13.7	59.2	54.9	57.4
v20b-5000	22.9	96.3	3.7	77.1	6.1	0.8	7.7	55.3	32.4	43.1

Figure A.9: Results of anomaly detection for all tests with $S_t > 0$ as threshold

Parameter	Max	Threshold	testsID	true+Rate	true-Rate	false+Rate	false-Rate	+Likelihoo	-Likelihoo	diagnostic	precision	F1Score	FbScore
true+Rate	79.0	0.0_aS	v10-2000	79.0	40.4	59.6	21.0	1.3	0.5	2.6	21.1	33.3	24.7
true-Rate	99.0	0.08_aS	v20b-5000	8.5	99.0	1.0	91.5	8.7	0.9	9.4	63.6	15.0	27.7
false+Rate	59.6	0.0_aS	v10-2000	79.0	40.4	59.6	21.0	1.3	0.5	2.6	21.1	33.3	24.7
false-Rate	91.5	0.08_aS	v20b-5000	8.5	99.0	1.0	91.5	8.7	0.9	9.4	63.6	15.0	27.7
+Likelihoo	19.0	0.08_aS	v17-5000	36.5	98.1	1.9	63.5	19.0	0.6	29.4	79.4	50.0	64.3
-Likelihoo	0.9	0.03_aS	v20b-5000	15.2	98.2	1.8	84.8	8.6	0.9	10.0	63.4	24.5	38.8
diagnostic	29.4	0.08_aS	v17-5000	36.5	98.1	1.9	63.5	19.0	0.6	29.4	79.4	50.0	64.3
precision	79.4	0.08_aS	v17-5000	36.5	98.1	1.9	63.5	19.0	0.6	29.4	79.4	50.0	64.3
F1Score	62.9	0.0_aS	v19-5000	61.1	93.3	6.7	38.9	9.1	0.4	21.9	64.8	62.9	64.0
FbScore	65.7	0.03_aS	v17-5000	47.2	96.5	3.5	52.8	13.3	0.5	24.2	72.8	57.2	65.7

Figure A.10: Results of anomaly detection for best result parameters using only anomaly score ($S_t > \delta$) as threshold

testsID	threshold	true+Rate	true-Rate	false+Rate	false-Rate	+Likelihoo	-Likelihoo	diagnostic	precision	F1Score	FbScore
v10-10000	0.0_aS	43.2	80.6	19.4	56.8	2.2	0.7	3.2	31.0	36.1	32.9
v10-2000	0.08_aS	60.8	63.2	36.8	39.2	1.7	0.6	2.7	25.0	35.4	28.3
v10-5000	0.08_aS	55.3	68.8	31.2	44.7	1.8	0.6	2.7	26.3	35.7	29.4
v11-5000	0.08_aS	42.2	92.2	7.8	57.8	5.4	0.6	8.6	52.2	46.7	49.8
v12-5000	0.08_aS	18.2	94.6	5.4	81.8	3.4	0.9	3.9	40.5	25.1	32.5
v13-5000	0.03_aS	33.6	94.5	5.5	66.4	6.1	0.7	8.8	55.4	41.8	49.0
v14-5000	0.03_aS	42.1	96.3	3.7	57.9	11.4	0.6	19.0	69.8	52.5	61.7
v14d-5000	0.08_aS	34.8	85.4	14.6	65.2	2.4	0.8	3.1	32.5	33.6	32.9
v15-5000	0.0_aS	27.7	95.0	5.0	72.3	5.6	0.8	7.3	52.9	36.4	44.8
v15d-5000	0.06_aS	29.6	88.2	11.8	70.4	2.5	0.8	3.1	33.5	31.4	32.6
v16-5000	0.06_aS	36.6	94.1	5.9	63.4	6.2	0.7	9.2	55.6	44.1	50.4
v16d-5000	0.08_aS	24.8	86.2	13.8	75.2	1.8	0.9	2.0	26.5	25.6	26.2
v17-2000	0.03_aS	50.9	95.7	4.3	49.1	11.7	0.5	22.8	70.2	59.0	65.3
v17-3000	0.0_aS	38.4	93.3	6.7	61.6	5.7	0.7	8.6	53.4	44.7	49.6
v17-4000	0.0_aS	38.6	93.1	6.9	61.4	5.6	0.7	8.4	52.9	44.6	49.2
v17-5000	0.03_aS	47.2	96.5	3.5	52.8	13.3	0.5	24.2	72.8	57.2	65.7
v17a-5000	0.03_aS	47.2	96.5	3.5	52.8	13.3	0.5	24.2	72.8	57.2	65.7
v17b-5000	0.08_aS	44.2	96.6	3.4	55.8	13.0	0.6	22.5	72.4	54.9	64.2
v17c-5000	0.06_aS	40.5	97.6	2.4	59.5	16.5	0.6	27.1	76.9	53.0	65.2
v18-5000	0.0_aS	43.6	93.1	6.9	56.4	6.3	0.6	10.5	56.1	49.1	53.1
v19-5000	0.03_aS	49.5	96.0	4.0	50.5	12.3	0.5	23.3	71.2	58.4	65.5
v20-5000	0.06_aS	32.9	97.8	2.2	67.1	14.9	0.7	21.7	75.0	45.8	59.7
v20b-5000	0.0_aS	22.9	96.3	3.7	77.1	6.1	0.8	7.7	55.3	32.4	43.1

Figure A.11: Best results of anomaly detection for all tests using only anomaly score ($S_t > \delta$) as threshold

Parameter	Max	Threshold	testsID	true+Rate	true-Rate	false+Rate	false-Rate	+Likelihoo	-Likelihoo	diagnostic	precision	F1Score	FbScore
true+Rate	87.2	0.79_aS+aL	v10-2000	87.2	23.6	76.4	12.8	1.1	0.5	2.1	18.7	30.8	22.2
true-Rate	99.9	0.98_aSxaL	v17-5000	24.9	99.9	0.1	75.1	165.8	0.8	220.3	97.1	39.6	61.4
false+Rate	76.4	0.79_aS+aL	v10-2000	87.2	23.6	76.4	12.8	1.1	0.5	2.1	18.7	30.8	22.2
false-Rate	99.3	0.99_aSxaL	v20b-5000	0.7	99.7	0.3	99.3	2.5	1.0	2.5	33.3	1.5	3.4
+Likelihoo	259.7	0.99_aSxaL	v17-5000	21.6	99.9	0.1	78.4	259.7	0.8	331.1	98.1	35.5	57.5
-Likelihoo	1.2	0.79_aL	v15d-5000	41.3	50.7	49.3	58.7	0.8	1.2	0.7	14.5	21.4	16.6
diagnostic	331.1	0.99_aSxaL	v17-5000	21.6	99.9	0.1	78.4	259.7	0.8	331.1	98.1	35.5	57.5
precision	98.1	0.99_aSxaL	v17-5000	21.6	99.9	0.1	78.4	259.7	0.8	331.1	98.1	35.5	57.5
F1Score	65.9	0.93_1S+aL	v17b-5000	63.5	94.1	5.9	36.5	10.8	0.4	27.8	68.5	65.9	67.4
FbScore	73.6	0.86_aSxaL	v17b-5000	46.6	98.5	1.5	53.4	30.7	0.5	56.6	86.1	60.5	73.6

Figure A.12: Results of anomaly detection for overall best result parameters

testsID	threshold	true+Rate	true-Rate	false+Rate	false-Rate	+Likelihoo	-Likelihoo	diagnostic	precision	F1Score	FbScore
v10-10000	0.0_aS	43.2	80.6	19.4	56.8	2.2	0.7	3.2	31.0	36.1	32.9
v10-2000	0.97_aSxaL	16.7	97.2	2.8	83.3	5.9	0.9	6.9	54.3	25.5	37.4
v10-5000	0.83_aSxaL	36.0	85.6	14.4	64.0	2.5	0.7	3.4	33.6	34.7	34.0
v11-5000	0.79_aSxaL	44.5	93.1	6.9	55.5	6.4	0.6	10.8	56.5	49.8	53.6
v12-5000	0.08_aS	18.2	94.6	5.4	81.8	3.4	0.9	3.9	40.5	25.1	32.5
v13-5000	0.03_aS	33.6	94.5	5.5	66.4	6.1	0.7	8.8	55.4	41.8	49.0
v14-5000	0.03_aS	42.1	96.3	3.7	57.9	11.4	0.6	19.0	69.8	52.5	61.7
v14d-5000	0.82_aSxaL	30.6	91.9	8.1	69.4	3.8	0.8	5.0	43.1	35.7	39.8
v15-5000	0.91_aS+aL	35.9	92.3	7.7	64.1	4.7	0.7	6.7	48.4	41.3	45.3
v15d-5000	0.89_1S+aL	38.0	85.6	14.4	62.0	2.6	0.7	3.6	34.7	36.3	35.3
v16-5000	0.06_aS	36.6	94.1	5.9	63.4	6.2	0.7	9.2	55.6	44.1	50.4
v16d-5000	0.08_aS	24.8	86.2	13.8	75.2	1.8	0.9	2.0	26.5	25.6	26.2
v17-2000	0.8_aSxaL	43.0	98.0	2.0	57.0	21.2	0.6	36.4	81.0	56.2	68.9
v17-3000	0.0_aS	38.4	93.3	6.7	61.6	5.7	0.7	8.6	53.4	44.7	49.6
v17-4000	0.0_aS	38.6	93.1	6.9	61.4	5.6	0.7	8.4	52.9	44.6	49.2
v17-5000	0.87_aSxaL	39.7	98.9	1.1	60.3	35.1	0.6	57.5	87.6	54.7	70.6
v17a-5000	0.87_aSxaL	39.7	98.9	1.1	60.3	35.1	0.6	57.5	87.6	54.7	70.6
v17b-5000	0.86_aSxaL	46.6	98.5	1.5	53.4	30.7	0.5	56.6	86.1	60.5	73.6
v17c-5000	0.79_aSxaL	40.1	98.0	2.0	59.9	19.9	0.6	32.6	80.1	53.5	66.8
v18-5000	0.0_aS	43.6	93.1	6.9	56.4	6.3	0.6	10.5	56.1	49.1	53.1
v19-5000	0.9_aSxaL	38.2	98.6	1.4	61.8	26.4	0.6	42.1	84.2	52.6	67.9
v20-5000	0.06_aS	32.9	97.8	2.2	67.1	14.9	0.7	21.7	75.0	45.8	59.7
v20b-5000	0.0_aS	22.9	96.3	3.7	77.1	6.1	0.8	7.7	55.3	32.4	43.1

Figure A.13: Overall best results of anomaly detection for all tests

References

[1] S. Ahmad and J. Hawkins. *How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites*. 2016. URL: https://arxiv.org/abs/1601.00720.

[2] Subutai Ahmad *et al.* "Unsupervised real-time anomaly detection for streaming data". In: *Neurocomputing* 262 (2017), pp. 134–147.

[3] R. Asaritoris *et al.* "Review of maritime transport". English. In: UNITED NATIONS PUBLICATION, 2013.

[4] D. Birant and A. Kut. "St-dbscan: An algorithm for clustering spatial–temporal data". English. In: *Data Knowledge Engineering* 60(1) (2007), 208–221.

[5] Y. Cui, S. Ahmad, and J. Hawkins. "The HTM Spatial Pooler: a neocortical algorithm for online sparse distributed coding". In: *Frontiers in Neuroscience* 11 (2017). DOI: http://dx.doi.org/10.3389/fncom.2017.00111.

[6] Martin Ester *et al.* "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *Kdd* 96 (1996), pp. 226–231.

[7] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. "From data mining to knowledge discovery in databases". English. In: *AI Magazine* 17.3 (1996), pp. 37–53. ISSN: 07384602.

[8] M. S. A. Graziano, C. S. R. Taylor, and T. Moore. "Complex movements evoked by microstimulation of precentral cortex". In: *Neuron* 34 (2002), 841–851.

[9] J. Hawkins and S. Ahmad. *Properties of sparse distributed representations and their application to Hierarchical Temporal Memory*. 2015. URL: https://arxiv.org/abs/1503.07469.

[10] J. Hawkins and S. Ahmad. "Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex". In: *Front. Neural Circuits* 10 (2016), pp. 1–13.

[11] J. Hawkins, S. Ahmad, and D. Dubinsky. *Cortical learning algorithm and hierarchical temporal memory*. 2011. URL: http://numenta.org/resources/HTM_CorticalLearningAlgorith
pdf.

[12] J. Hawkins *et al. Biological and Machine Intelligence (BAMI)*. Initial online release 0.4. 2016. URL: http://numenta.com/biological-and-machine-intelligence/.

[13] T. Hromádka, M. R. DeWeese, and A. M. Zador. “Sparse representation of sounds in the unanesthetized auditory cortex”. In: *PLoS Biol.* 6 (2008), pp. 124–137.

[14] J.L. Kolodner. “An introduction to case-based reasoning”. English. In: *Artificial Intelligence Review* 6.1 (1992), pp. 3–34. ISSN: 02692821. DOI: 10.1007/BF00155578.

[15] R. O. Lane *et al.* “Maritime anomaly detection and threat assessment”. English. In: *13th Conference on Information Fusion, Fusion 2010*. 2010.

[16] Lee *et al.* “Traclass: trajectory classication using hierarchical region-based and trajectory-based clustering”. English. In: *Proceedings of the VLDB Endowment* 1(1) (2008b), pp. 1081–1094.

[17] E. Lutins. *DBSCAN: What is it? When to Use it? How to use it?* Accessed on 16-02-2018. 2017. URL: https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818.

[18] J. D. Mazimpaka and S. Timpf. “Trajectory data mining: A review of methods and applications”. English. In: *Journal of Spatial Information Science* 13 (2016), pp. 61–99.

[19] Numenta. *Geospatial Tracking - Learning the Patterns in Movement and Detecting Anomalies*. 2014. URL: https://numenta.com/assets/pdf/whitepapers/Geospatial%20Tracking%20White%20Paper.pdf.

[20] Numenta. *Sequence learning and prediction in the neocortex*. Accessed on 14-06-2018. 2018. URL: https://numenta.com/neuroscience-research/sequence-learning/.

[21] Numenta. *Sparse distributed representations*. Accessed on 14-06-2018. 2018. URL: https://numenta.com/neuroscience-research/sparse-distributed-representations/.

[22] Scott Purdy. *Encoding Data for HTM Systems*. 2016. URL: https://arxiv.org/abs/1602.05925.

[23] C. J Van Rijsbergen. *Information Retrieval (2nd ed.)* Butterworth-Heinemann, 1979.

[24] M. Weliky *et al.* “Coding of natural scenes in primary visual cortex”. In: *Neuron* 37 (2003), 703–718.