THE VESSEL ROUTE PATTERN EXTRACTION AND ANOMALY DETECTION FROM AIS DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÖZDE BOZTEPE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

AUGUST 2019

Approval of the thesis:

THE VESSEL ROUTE PATTERN EXTRACTION AND ANOMALY DETECTION FROM AIS DATA

submitted by **GÖZDE BOZTEPE** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _______________

Prof. Dr. Pınar Karagöz
Supervisor, **Computer Engineering, METU** _______________

Examining Committee Members:

Prof. Dr. Nihan Kesim Çiçekli
Computer Engineering, METU _______________

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _______________

Assist. Prof. Dr. Hacer Yalım Keleş
Computer Engineering, Ankara University _______________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Gözde Boztepe

Signature :

ABSTRACT

THE VESSEL ROUTE PATTERN EXTRACTION AND ANOMALY DETECTION FROM AIS DATA

Boztepe, Gözde
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Pınar Karagöz

AUGUST 2019, 85 pages

The need for a variety of auxiliary analytical tools to enhance marine safety and marine status awareness has been expressed by various platforms. There are lots of data sources breaking out while the ship is on cruising. Automatic Identification System (AIS) device that is widely used in vessels, is one of them. It broadcasts information such as type of ship, identity number, state, destination, estimated time of arrival (ETA), location, speed, direction, cargo. In this study, to aid operators while sailing, the trajectory extraction and anomaly detection tool have been developed. The AIS messages are used to improve a system for safe navigation. Three different approaches are applied for the prediction of the vessel trajectories. Later, movements that have not matched the route patterns and unusual stop anomalies have been examined.

Keywords: AIS, trajectory, anomaly, LSTM, vessel, maritime

ÖZ

AIS VERILERI ILE ROTA ÖRÜNTÜSÜ ÇIKARIMI VE ANOMALI TESPITI

Boztepe, Gözde
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Pınar Karagöz

Ağustos 2019 , 85 sayfa

Farklı platformlar tarafından deniz güvenliğinin ve durum farkındalığının artırılması için çeşitli yardımcı analiz araçlarının kullanılması gerektiği dile getirilmiştir. AIS (Automatic Identification System), gemilerde yaygın olarak bulunan; geminin tipi, kimlik numarası, durumu, varış noktası, tahmini varış zamanı (ETA), konumu, hızı, yönü ve kargosu gibi bilgileri açık bir şekilde yayınlayan bir cihazdır. Bu bilgiler istasyonlarca dinlenerek veri tabanlarına aktarılmaktadır. Bu çalışmanın amacı AIS verilerini kullanarak rota örüntülerinin çıkarımını yapmak ve ortaya çıkan anomalileri tespit etmektir. Mevcut rota örüntülerinin çıkarılması üzerine üç farklı yaklaşım kullanılmıştır. Daha sonra, rota örüntülerine uymayan hareketler ve olağan dışı durma anomalileri incelenmiştir.

Anahtar Kelimeler: anomali, rota, AIS mesajı, LSTM, gemi hareketleri

To my family and my love

ACKNOWLEDGMENTS

I would like to thank my thesis advisor Prof. Pınar Karagöz. I was always greeted by a smiling face and listen patiently.

I would like to express my gratitude to my committee members, Prof. Nihan Kesim Çiçekli and Asst. Prof. Hacer Yalım Keleş for their precious feedback.

I would like to thank my beloved one, Mustafa Mert Karataş, who is always by my side and make me feel his support until the end.

I thank my industry side advisor Orhan Ayran. He always listens carefully and solves problems quickly. Also, I want to thank Mete Han Kahraman for his help in the study.

I would like to thank my family for raising me so carefully. I always feel their support in every way. I am so glad to have my parents, Şenay and Ercan, and my little sisters Asu and Doğa Berra.

I would like to show my gratitude to my teammates and in particular to Sinan Göker and Naciye Öztürk. Also, I thank my team leaders Elif Gökçe Işık and Örsan Akbaba for their patience during my thesis.

I would like to thank Ziya Ulusoy and Özgür Özuğur for their contributions to the project.

I thank my dear friends Ayşe Merve Seyhan and Özlem Altunbulak for helping me in a rough time and always being with me.

I would like to express my gratitude to Prof. Oğuz Ergin and in particular to my bachelor degree advisor Asst. Prof. Mehmet Tan. Their contributions to my life are huge.

I would like to thank The Presidency of Defence Industries and HAVELSAN for their financial supports in this study.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AIS	Automatic Identification System
IMO	International Maritime Organization
SOLAS	Safety of Life at Sea
STDMA	Self Organizing Time Division Multiple Access
MMSI	Maritime Mobile Service Identity
SOG	Speed Over Ground
COG	Course Over Ground
EMSA	European Maritime Safety Agency
NED	North East Down Coordinate System
WGS-84	World Geodetic System-1984 Coordinate System
TREAD	Traffic Route Extraction and Anomaly Detection
DBSCAN	Density Based Spatial Clustering Application with Noise
LSTM	Long Short Term Memory
LCS	Longest Common Subsequence
LCSR	Longest Common Subsequence Ratio
MongoDB	Mongo Database
DBMS	Database Management System
MAE	Mean Absolute Error
MSE	Mean Squared Error
kNN	k Nearest Neighbors

CHAPTER 1

INTRODUCTION

1.1 Overview

Maritime transportation is highly used by people. Not just for pleasure voyages but also it is used for carrying cargo or transporting fluids, such as petroleum, chemicals, and oils. Although using marine transportation is less costly, its safety is debatable. Recently, both in the civil and military domain, the number of people who concern about maritime safety has increased. The need for auxiliary analytical tools to enhance marine safety and marine status awareness has been expressed by various platforms.

Maritime safety is a crucial topic. Recently, the number of marine accidents has increased. According to the European Maritime Safety Agency (EMSA) [2], 2018 Marine casualties and incidents report [3], nearly 20000 accidents have occurred between 2011 and 2017. Approximately, 15000 of these incidents have been reported in Europe.

There are lots of data sources breaking out while the ship is on cruising. Automatic Identification System (AIS) [4] is one of them. International Maritime Organization (IMO) [5] the Safety of Life at Sea (SOLAS) [6] convention specifies standards for ship construction, equipment, and operations. It makes carrying AIS mandatory.

1.1.1 Automatic Identification System

AIS broadcasts information such as ship location, speed, type of ship, identity number. MMSI number, location, speed, course, type of ship, call sign are categorized as kinematic ship information. Voyage information are type of cargo, ship's draught, destination port, and estimated time of arrival (ETA). There are two different AIS message types. Although the frequency of the first type of message is adjusted according to the speed of the ship, it must be broadcast in at least 3 minutes intervals. The second type of message contains voyage information. It is broadcast every 6 minutes.

The types of AIS devices are Class A, Class B and receivers only. Class A is mandatory for large commercial ships. Class B transmits less information than Class A. It is usually used by small vessels.

AIS uses Self Organizing Time Division Multiple Access (STDMA) [7] to enlarge the range of signals. The signal has a range of 20 miles approximately. AIS messages are collected by shore-based AIS receivers and stored in data centers. When vessels sail away further from the shore, AIS signals are being sent to satellites. Satellites relay the information to the shore.

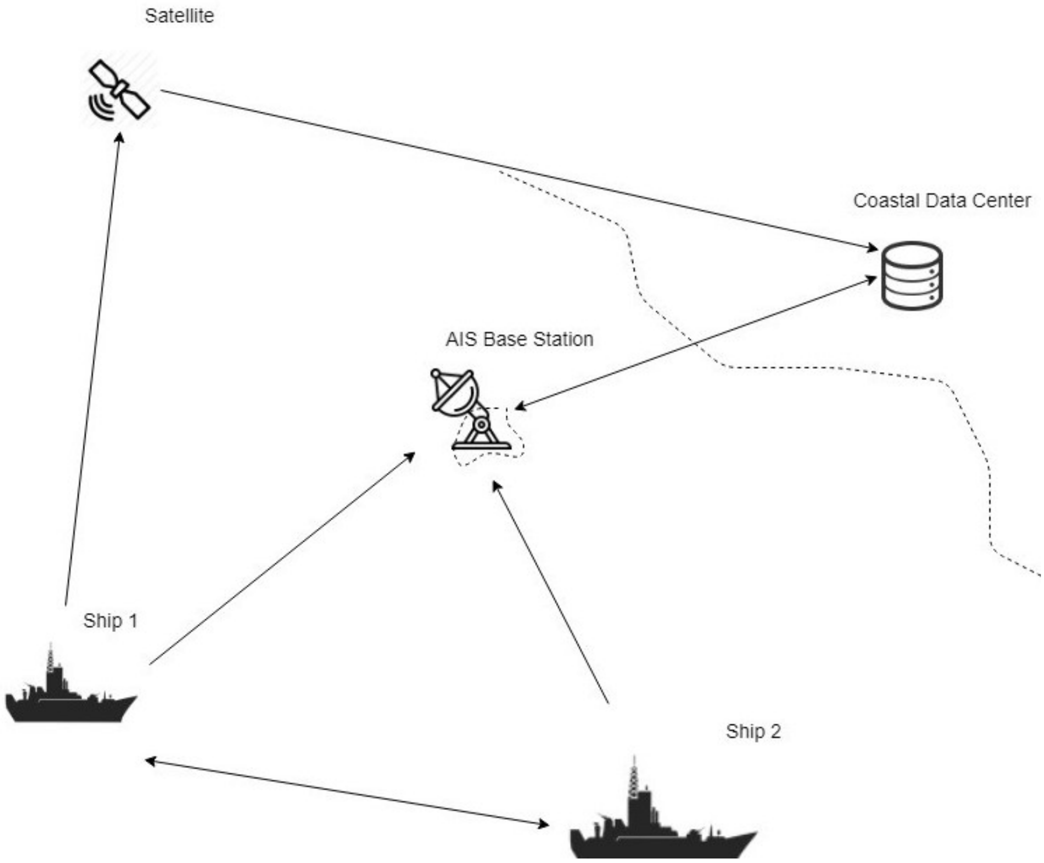


Figure 1.1: The AIS Message Circulation Between The Platforms.

In this study, AIS data set will be used for improving a system for safe navigation.

1.1.2 AIS Message Data

The data set used in this study is obtained from Distributed Event Based System 2018 Grand Challenge [1].

AIS [8] data contains ship id, ship type, speed, latitude, longitude, course, heading, timestamp, departure port, arrival port and arrival time information. There are approximately 540 thousand rows of data. The data contains AIS messages between 10 March and 19 May 2015 [1].

- SHIP ID is a unique IMO number for each ship.

- SHIP TYPE gives information about the type of ship and its cargo.
- SPEED is the ship's current speed in knots.
- LAT is the latitude of ship coordinates.
- LON is the longitude of ship coordinates.
- COURSE is the ship's current direction in degree.
- HEADING is the compass direction.
- TIMESTAMP consists of the current day, hour and minute.
- DEPARTURE PORT is the name of the port in which the ship left.
- ARRIVAL PORT is the name of the port in which the ship will arrive.
- ARRIVAL TIME is predicted the time of arrival.

Ship type contains two digits. The first digit means the category of the vessel. The second digit gives the cargo information of the ship [9].

Digit	Statement
1	Reserved
2	Wing In Ground
3	Special Category
4	High-Speed Craft
5	Special Category
6	Passenger
7	Cargo
8	Tanker
9	Other

Table 1.1: The First Digit in Ship Type

Digit	Statement
1	Major Hazard (Haz A)
2	Hazard (Haz B)
3	Minor Hazard (Haz C)
4	Recognizable Hazard (Haz D)

Table 1.2: The Second Digit in Ship Type

Table 1.3: Sample AIS Data [1].

ID	Type	Speed	Lon	Lat	Course	Heading	Timestamp	Departure Port	Arrival Time	Arrival Port
0xc3	99	8.2	14.56034	35.8109	109	511	10-03-15 12:15	MARSAXLOKK	10-03-15 13:13	VALLETTA
0xc3	99	8.1	14.56537	35.81219	47	511	10-03-15 12:17	MARSAXLOKK	10-03-15 13:13	VALLETTA
0xc3	99	7.8	14.57128	35.81421	47	511	10-03-15 12:19	MARSAXLOKK	10-03-15 13:13	VALLETTA
0x6c	36	5	-5.316057	35.90345	348	511	14-03-15 13:48	CEUTA	14-03-15 15:25	GIBRALTAR
0x6c	36	6.5	-5.316607	35.90769	351	511	14-03-15 13:51	CEUTA	14-03-15 15:25	GIBRALTAR
0x70	30	8.8	14.59415	35.87637	343	511	18-03-15 10:11	MARSAXLOKK	18-03-15 10:32	VALLETTA
0x70	30	6.4	14.59434	35.87656	343	511	18-03-15 10:14	MARSAXLOKK	18-03-15 10:32	VALLETTA
0x70	30	5.3	14.59464	35.87675	343	511	18-03-15 10:17	MARSAXLOKK	18-03-15 10:32	VALLETTA
0x6d	83	0.7	-5.352159	35.92432	214	247	18-03-15 15:01	CEUTA	27-04-15 6:03	GIBRALTAR
0x6d	83	0.2	-5.352372	35.92413	208	263	18-03-15 15:04	CEUTA	27-04-15 6:03	GIBRALTAR
0x6d	83	0.3	-5.352543	35.92402	208	263	18-03-15 15:07	CEUTA	27-04-15 6:03	GIBRALTAR

1.2 Contributions

The aim of this study is developing a tool for helping safe navigation. The tool contains two different abilities. Firstly, ship trajectories are extracted by using historical AIS messages. The second step is anomaly detection by using the obtained trajectories and defined rules.

Our contributions are as follows:

- Prediction of next position and sequence using LSTM algorithm
- Presenting an improved version of an existing algorithm [10] and eliciting of the vessel trajectories
- Determination of differences between two trajectories using LCS metric

1.3 The Outline of the Thesis

In this thesis, there are six chapters beginning with this introductory chapter. The second chapter contains a summary of the reviewed related work about both ship trajectory prediction and anomaly detection. Chapter 3 presents information about technologies that have been used in this study. The first part contains a detailed explanation of the methods which have been used in trajectory and arrival port prediction. Afterward, the techniques for anomaly detection are described. Chapter 4 consists of a complete description of the implementation. In the beginning, clustering methods, long short term memory, and thread algorithm are explained. Later, the anomaly detection method is defined. Chapter 5 contains the analysis and results of the study. Finally, the last chapter includes a discussion of the results and information about future work.

CHAPTER 2

RELATED WORK

In this chapter, the first part mainly focuses on trajectory prediction studies. Afterward, works on anomaly detection are comprehensively reviewed.

2.1 Managing AIS Data

De Vreede, in the master thesis, presents using MongoDB to manage AIS data [11]. In that study, four attributes of AIS data have been decoded and stored. Because MongoDB supports the Spatial-Temporal analyses of AIS data, it is used. These four attributes are latitude, longitude, MMSI, and timestamp. To enhanced performance, four indexes and 4D Morton Code Index [12] have been developed. A 4D Morton Code Index is a decoded version of the four attributes, which are latitude, longitude, MMSI, and timestamp. It has been developed to improve the effectiveness of the database.

The AIS message can be of 27 different types. Each of them has a structure. Document store databases, like MongoDB, are flexible in the data scheme. They can handle complex data, queries and their performance and scalability. In the study, there are two different databases. The first is a spatial-temporal based database. The second is use case-based which are location, trajectory and bounding box. Morton Code Index has been using both in spatial-temporal focused and use case focused databases.

According to the results in De Vreede Study, the use of a case-based database is faster than spatial based in terms of querying trajectories. Using Morton Code indexing enables fast and effective response. However, before using the Morton Code, data should be clustered to improve effectiveness.

2.2 Movement Prediction

Bodunov and Schmidt [13] proposed a solution for DEBS Grand Challenge 2018 that is prediction of arrival port and arrival time [1]. The arrival port prediction is a classification problem whereas the arrival time prediction is a regression problem.

They used Random Forest, Gradient Boosting Decision Trees (GBDT) [14], XGBoost Trees [15] and Extremely Randomized Trees [16] algorithms to detect the arrival port. The Random Forest algorithm has 97% accuracy while the other approaches have 92% accuracy at most. The best prediction was made with the features that are Ship Type, Speed, Lon, Lat, Course, Heading, Departure Port Name, Reported Draught. Long Short-Term Memory [17] approach was discarded because each trip was labeled individually.

For the arrival port prediction, latitude, longitude, speed, course, heading, timestamp, departure and arrival port features were used as input into Feed-Forward Neural Network. Feed-Forward Neural Network was built up with a single input layer and one hidden layer of 200 neurons, mean squared error loss function and RMSProp [18] optimizer. The dropout ratio is 20%. The accuracy of NN is 90%.

Cunbao and Zheping [19] analyzed how AIS data is used effectively. They create AIS data-based Data Mining Platform which clusters AIS messages. Data Mining Object concept reflect maritime traffic instance. After some research, Data Mining Objects have been shaped by some criteria, such as Vessel Quantity Distribution, Quantity Distribution of various ship type, Draft Distribution, LOA Breadth Distribution, Tonnage Distribution. To store this information, database design is necessary. Static and voyage information like course over ground, speed over ground and position, Dynamic AIS information which is MMSI, Breadth, LOA, Draft are stored database tables. There are two database tables which are static voyage information table and dynamic information table for each AIS station. The study contains an algorithm that is used for selecting the best data mining object from the database. This algorithm groups DTO according to their similarity. Ship length, width, and tonnage to determine the similarity.

Mao and Tu [20] suggest the Extreme Learning Machine [21] based method which is for path predictions. According to authors some attributes which are latitude, longitude, COG, SOG, MMSI and time in AIS message are more useful for the learning and prediction process. In the study, data were obtained from the raw database file format. Thus, ArcMap [22] has been used as a transformation tool for processing data. In the preprocessing phase, the longest duration of navigation and route complexity have been calculated. The reason why calculate duration that short duration navigation has not enough AIS message to process. The $\cos\theta$ is calculated for each route. The mean value of $\cos\theta$ is the route complexity. The short duration navigation has not enough AIS message to process. Therefore, they are not included in the calculation. In the study, routes that have lower than 0,8 complexity are not suitable for processing.

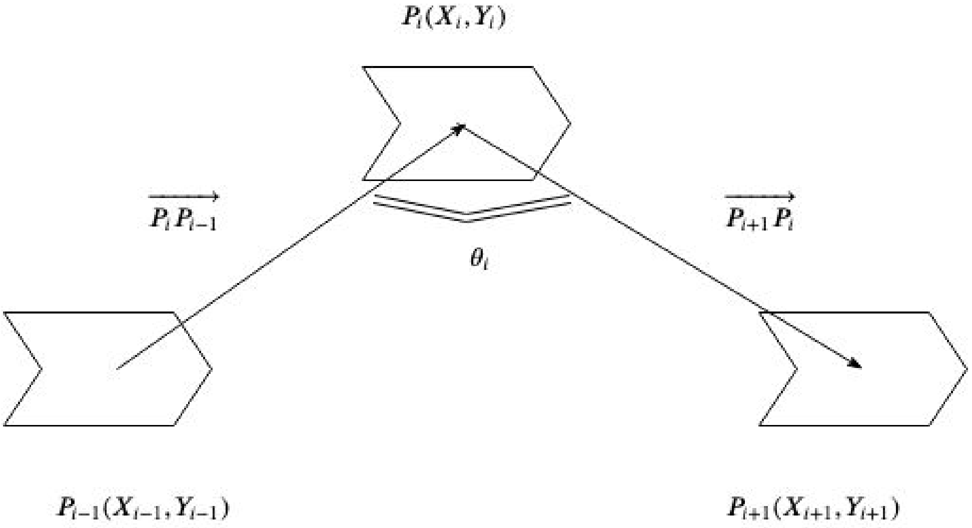

$P_i(X_i,Y_i)$
$\overrightarrow{P_iP_{i-1}}$
$\overrightarrow{P_{i+1}P_i}$
θ_i
$P_{i-1}(X_{i-1},Y_{i-1})$
$P_{i+1}(X_{i+1},Y_{i+1})$

Figure 2.1: The Angle Between The Previous and The Next Location of A Single Vessel.

In the database, there are 403599 records. There are some operations for the preprocessing phase. The first step is determining noisy trajectories. The shape of routes determines whether a route is clean or not. There are three types of noisy trajectory type: tangled, discontinues and loose. All the noisy routes are cleaned from data.

Next, the interpolation method is used for filling missing values. After removing erroneous speed data, missing speed value is calculated by using the Haversine distance in the one-minute interval.

After preprocessing, there are 200 clean trajectories which are stored in 200 CVS file. Each file has latitude, longitude, SOG, COG, ROT, time and MMSI information. To predict trajectories Extreme Learning Machine is used. To evaluate ELM, prediction of the same trajectory in 20 minutes and 40 minutes has been performed. Haversine distance calculation is also used for evaluation. ELM performance in 20 minutes is better. Prediction in 40 minutes is more though because of dynamic conditions. The error of the predicted location is between 0 and 2.5 in 20 minutes. On the other hand, in 40 minutes the error is between 0 and 6.

Shi, Zhiyuan, and Xu[23] present the LSTM-based method for flight trajectory prediction. In the study, trajectories have been predicted in 4D space using latitude, longitude, speed, and heading information [24]. The study contains three major parts such as collection of historical data, transforming coordinates, and building LSTM [17].

In the beginning, Automatic Dependent Surveillance-Broadcast is used to collect location, heading, landing and departure airport and call-sign information. ADS-B messages are broadcast three times per second [25]. Messages have been collected in the surveillance area. The recorded data is from June 2017 to November 2017.

In the WGS-84 [26, 27] coordinate system, the difference between altitude and position values leads to an increase in the error. Therefore, North East Down (NED) [27, 28] coordinate system has been selected for the work. Position information is converted from WGS-84 to NED.

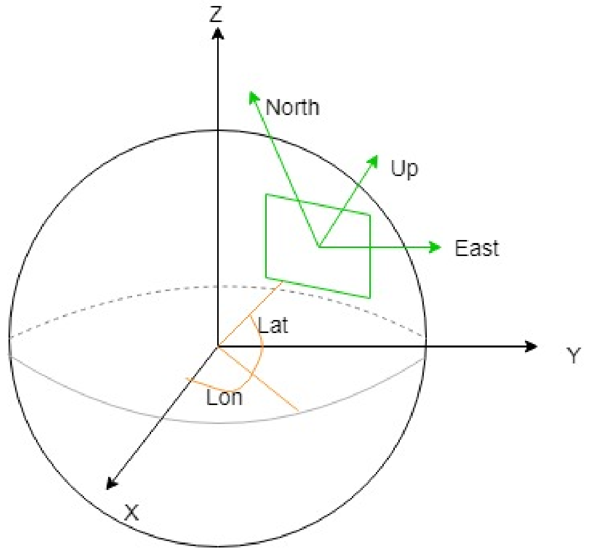


Figure 2.2: North East Down (NED) Coordinate System.

To the LSTM network, the data has been normalized with Min-Max normalization. The dynamic time warping (DTW) is applied to measure the similarity between predicted and real trajectories. Position, speed, and heading information create the input sequence for LSTM. The time interval is 10 minutes for two aircraft. The LSTM network is built with four layers with two hidden layers. LSTM model makes predictions using 10 points. The first hidden layer contains 30 neurons. The second hidden layer is designed with 60 neurons. The first hidden layer has been fed into the second one. The results are outputted one neuron layer with a linear activation function. To avoid over-fitting, the drop out ratio is 0.2, Mean Squared Error (MSE) is selected as the loss function. RMSprop is used as the optimizer.

At the end of the study, the LSTM network has been compared with the Markov Model [29] and the weighted Markov Model [30]. LSTM gives smoother trajectories than Markov Models. The weighted Markov Model has better results than Markov Model.

2.3 Anomaly Detection

This section consists of a detailed description of several trajectory prediction methods. In the beginning, Pallotta, Vespe and Bryan's work [10], Traffic Knowledge Discovery is clarified. Also, in the study TREAD algorithm is described. Later, Mao and Tu's study is explained.

In NATO Science and Technology Organization, Centre for Maritime Research and Experimentation [31] develop a framework called Traffic Route Extraction for Anomaly Detection [10] (TREAD). Maritime traffic route can be extracted using clustering technique in AIS data. The framework has been formed by an object-based model that consists of vessel, waypoints, stationary, entry, exit and route objects. All waypoint objects are shaped by using the DBSCAN algorithm. Route objects have been derived from waypoint objects. Entropy is used for qualification of extracted routes.

The vessel objects are derived and updated from the AIS data stream. The vessel object contains call sign, name, International Maritime Organization(IMO) [5] number, size, position, course over ground (COG) and speed over ground (SOG) of the ship. There is a vessel objects manager to detect entrances in the selected bounding box. The vessel objects manager updates the information and status of vessels. There are two different vessel status, such as stationary and sailing. Waypoint objects are created and updated by these events. There is a stationary objects manager which collects vessel objects having a lower speed than the given threshold. Also, the stationary object manager consists of stationary points, like port and offshore platforms (POs).

DBSCAN [32] algorithm is used in waypoint clustering. DBSCAN created and updated clusters with forming objects based on density in their neighborhood. The points that do not belong to any cluster are noise in this study. Other types of waypoints are an entry (EOs) and exit (EXs) points. There are entry and exit points managers to create and update according to the selected area. After clustering waypoints, route objects can be extracted by connecting two points. Route objects manager creates and updates the route objects. When a ship enters the selected area, the manager checks its features for the routes. If there is a route used by vessels having the same features, the vessel is added to the related cluster. Otherwise, the vessel is used to create a new route. To activate the new route, there should be enough number of detection.

To predict future routes, route objects are classified using derived historical route objects. At the same time, trajectory anomalies have detected. A vessel is converted to a time vector, which contains current coordinates of vessels and next coordinates according to SOG and COG in the current time window. The vector has been associated with the current state. After observed position vectors, these have been combining as a sequence. If the sequence is different from the derived route, the trajectory anomaly has been detected. In the study, entropy is used to check the quality of predictions.

The AIS dataset contains the messages between January 1st to February 20th 2013 for La Spezia in the Northern Tyrrhenian Sea. The detected objects are displayed in Table 2.1.

Object Type	Count
Stationary Areas	27
Entry Points	14
Exit Points	16
Routes	60

Table 2.1: TREAD Results for La Spezia

Roy [33] expresses anomaly definitions in maritime domain. In the study, the types of anomalies and explanations are based on Canadian Forces. There are huge data while navigation in maritime that operators cannot handle it. The need is defining and reporting anomalies to operators in navigation automatically. The purpose of the study is knowledge representation for the rule based expert systems. The study contains description of knowledge acquiring and interpretation aspects and developing abnormal behavior detection prototypes.

Threats and anomalies are two different conceptions. Threats can be defined as activities that endanger the own-ship. However, anomalies are explained as activities that are not common behaviours. Anomalies can be categorized as dynamic kinematic and dynamic non-kinematic anomalies. The dynamic kinematic and non-kinematic categories of the anomalies are presented in tables 2.2 and 2.3.

Course	Not Towards Expected Ports Not Towards a port
Speed	Too low speed for the class of the ship Unattainable speed Trawling speed in closed zones
Reporting	Missing Reports No report Report Quality Track appears out of anywhere
Location	Ship Position Route Zone Depth Legal Limit Navigability Proximity
Manouver	Transiting Vessel Loitering

Table 2.2: Dynamic Kinematic Anomalies

Passengers	Too many people on deck Undesirable Undeclared More than a percent of people are sick with similar disease
Cargo List	Doesn't fit the type of ship Dangerous Cargo
Crew List	Crew type Crew size More than a percent of people are sick with similar disease
Last Port of Call	Failed CBRN (CBSA checks ships when they leave ports) Did not cleared 24 hours pre-loading report
Ship signature	
Next Port of Call	

Table 2.3: Dynamic Non-Kinematic Anomalies

In Khan, Rahim and Ahmad's [34] study, a longest common subsequence [35] based algorithm is proposed for the time series. The algorithm calculates the similarity between two time series efficiently. The LCS [36] is used to find the maximum length of the common subsequence. Dynamic Programming based LCS was proposed by Hirschberg [36]. In the paper, the proposed algorithm, LCSS, works half of the time of DP based LCS. LCSS algorithm is presented in Algorithm 1. There are two time series, S and T, and their length m and n respectively. Tp stores matched positions. Md is the length of smaller series divided by 2. l is the smaller time series length. k is the array of matched items.

Algorithm 1 Longest Common Subsequence based Algorithm [34]

```
while $l \leq Md\ and\ Count_1 < len(m)$ do
  for $i = l\ to\ m$ do
    for $j = Tp\ to\ n$ do
      if $(S\ or\ T\ is\ empty)$ then
        return LCSS is zero
      else if $(S[i] = T[j]\ and\ Tp = 0)$ then
        $add\ S[i]\ to\ k$
        $Tp = j + 1$
        $Count + +$
        $Continue$
      else if $(S[i]! = T[j])$ then
        if $(i = m\ and\ j = n\ andLCSSiszero)$ then
          return
        end if
      end if
    end for
  end for
  if $(Count < Count_t)$ then
    $LCSS_{Old} = LCSS_{New}$
    $Count_1 = Count$
    $Count = 0$
  end if
  $remove\ l^{th}\ from\ S$
  $l + +$
  $Tp = 0$
end while
```

Moreover, there is an extensive version of the algorithm. To work with LCSS, the elements of two time-series have to be at consecutive positions. The special longest common subsequence algorithm calculates the similarity between nonconsecutive time series. For instance, there is a sensor that measures the temperature. The extracted temperature degrees are 25, 26, 22, 27, 25, 28, 25 at $t_1, t_2, t_3, t_4, t_5, t_6, t_7$. The sensors are reading new temperature values as 25, 22, 27, 25 at t_1, t_2, t_3, t_4. LCSS based decision support system cannot find the similarity between two time-series. However, they are similar. To solve this problem SLCSS is proposed. The algorithm of SLCSS is presented in algorithm 2. Sp indicates the position of the matched elements in S. Tp stores the positions of T. Count keeps the number of sequential matches between two series. The maximum number of consecutive matches is recorded as C_1. k_1 is the temporal storage for common subsequence. k_2 gives the maximum length of the subsequence. The proposed algorithms work more effective than DP based LCS [36]. LCSS makes half of the comparison of DP-based LCS.

Algorithm 2 Special Longest Common Subsequence Algorithm [34]

```
while $l \leq Md\ and\ Count_t < len(m)$ do
  for $i = l\ to\ m$ do
    for $j = Tp\ to\ n$ do
      if $(S[i] = T[j]\ and\ Tp = 0\ and\ Sp = 0)$ then
        $add\ S[i]\ to\ k_1$
        $Sp = i$
        $Tp = j + 1$
        $Count + +$
        $Continue$
      else if $(S[i]! = T[j]\ andSp = i + 1\ and\ Tp = j)$ then
        $add\ S[i]\ to\ k$
        $Sp = i$
        $Tp = j + 1$
        $Count + +$
        $Continue$
      else if $S[i]?T[j]\ and\ Count > C_1$ then
        $C_1 = Count$
        $replace\ SLCSS\ ink_2byk_1$
        $Count = 0$
        $k1 = empty$
      end if
    end for
  end for
  $removel^{th}fromS$
  $l + +$
  $Tp = 0$
end while
```

Handayani, Sediono, and Shah [37] propose using Support Vector Machine method to detect anomalies. In the work, trajectories are assumed that they have been already extracted. Abnormal behavior can occur while navigation, such as random movement, unexpected stops, close track, direction violence, etc. The purpose of the study is creating a model that represents anomalies. To build the model, SVM is used which is a supervised method. SVM measures the similarity between input data and stored data. They have split AIS data into two part as interpolated and non-interpolated. SVM is applied to both data. The tracks are already categorized as normal and anomaly. Then, both groups have been randomized. After that, first group of normal and anomaly tracks is combined for training. SVM classification has applied. Then, the second group of both data is combined for testing and SVM has been applied.

The work has focused on two anomalies. The first is unusual stop routes. There are trajectories that ships follow and stop in the middle of the ocean. The second one is u-turn trajectories. The vessels start following the normal route. Then, the ships make a abnormal turn and start to make u-turn.

The dataset that is used in the study contains AIS message in Port Klang from July to September 2013. It has 9845 rows. Behavior model focus on seven attributes that are status, speed, location, course, heading, and timestamp. They have tried three different types of separations for training and test data. The best result is obtained using 0.7 for training and 0.3 factor for test data. Classification with interpolated data has a better result.

CHAPTER 3

BACKGROUND

3.1 Storing AIS Data

Storing data is a crucial issue. Working with the complex and large volume of data needs efficient databases. Firstly, historical AIS messages are a large volume of data. There are twelve attributes in a single AIS message. Some of them are mainly accessed by queries, such as SHIPID, LAT, LON, and TIMESTAMP. Thus, the need for indexing these fields has taken place. Traditional Relational Databases do not confront the needs because of scaling problems. Most NoSQL databases are horizontal scaling, on the contrary to relational databases. Another vital issue is the geospatial indexing. Although ship id is used for uniqueness, the location of a vessel is the most accessed fields by queries. Hence, creating an index on latitude and longitude becomes inevitable.

De Irine's study [11] explains the suitability of MongoDB [38] for use. MongoDB is a document-oriented NoSQL [39] database. MongoDB has effectiveness, flexibility and horizontal scaling that means connecting more machines into the existing resources. Also, it has geospatial indexing. Because of those reasons, AIS messages have been stored in MongoDB. With geospatial indexes, geospatial queries are easily performed. The details of these queries will be explained in Chapter 4.

3.2 AIS Message Clustering

The clustering of AIS messages is the first approach to trajectory prediction. In this method, the first step is to cluster the existing AIS messages. Then, classification methods are used to predict the route for new AIS messages. For clustering DBSCAN [32], KMeans [40] algorithms have been applied. Random Forest Algorithm [41] is used for the classification of the new data.

3.2.1 Random Forests

Random Forests Algorithm[41] is one of the supervised classification methods. It is a special form of the Decision Tree Algorithm. The Decision Tree Algorithm uses all data feature to build a rule-based tree. Yet, the Random Forest Algorithm selects features randomly to make more than one decision tree. The more trees in the forest, the stronger the model becomes. Random Forest Algorithm handles any missing values. Also, it can work with categorical features. Random Forest Algorithm architecture has given in Figure 3.1.

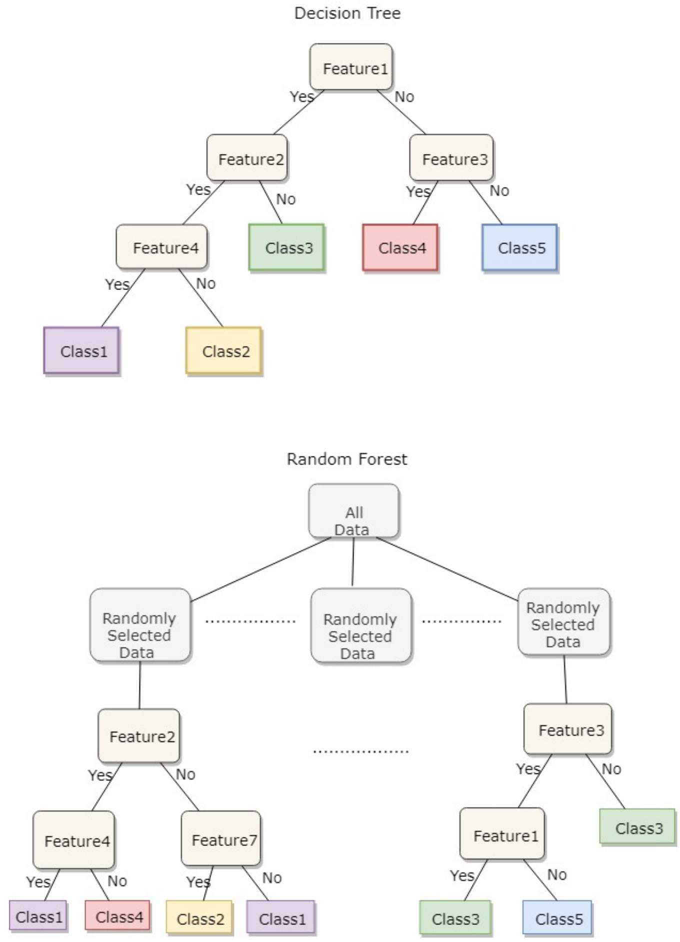


Figure 3.1: Decision Tree and Random Forest Architecture

3.2.2 Density Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise[32] (DBSCAN) is an unsupervised learning algorithm. It works based on a radius of clusters and the least number of points in clusters. DBSCAN can handle outliers in the data. In order to work with DBSCAN, the distance value must be given. Theoretically, the algorithm visits every point and determines its neighbors and clusters within distance. All points have been classified as core, density reachable or outlier by DBSCAN. If a point is inside a cluster that is a core point. Density reachable points are reachable points by core points within distance. The other points are outliers. The basic form of DBSCAN is given in Figure 3.2

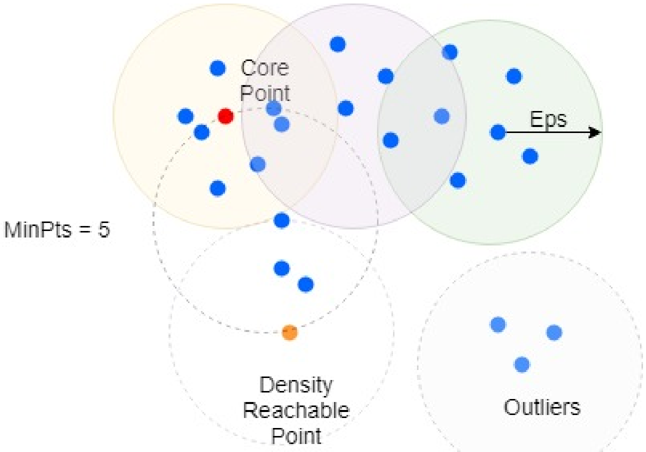


Figure 3.2: DBSCAN Algorithm Architecture

3.2.3 K Means

K Means [40] is an unsupervised learning algorithm. The aim of the algorithm is finding the groups in data. K refers to the number of clusters. K Means works with similarity. It uses Euclidean Distance to measure the similarity. The K Means algorithm has two main steps. The first one is finding the nearest centroid for each point. The second step is the centroid update. The basic example of K Means is given in

Figure 3.3. The aim of K Means is formulated in equation 31.

$$\mathrm{J} = \sum_{i=1}^{k} \sum_{j=1}^{n} \|\mathbf{x_j}^{i} - c_i\|^2 \tag{31}$$

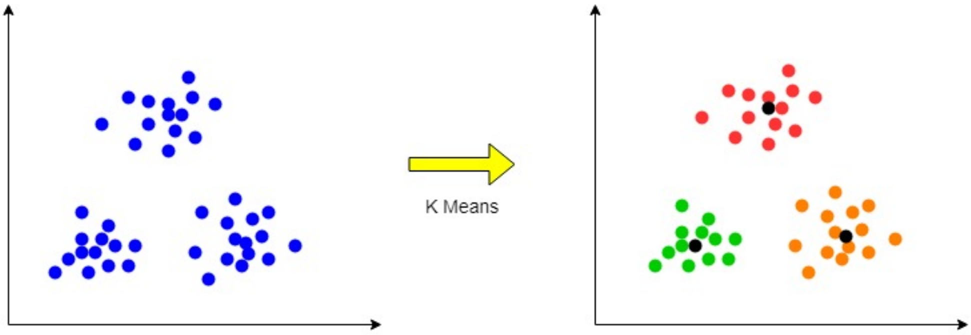


Figure 3.3: K Means Algorithm Visualisation

3.2.4 Euclidean Distance

Euclidean distance is a distance metric that measures the ordinary distance between two given points. Euclidean Distance formula is shown in Equation 32

$$d = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \tag{32}$$

3.2.5 Cosine Distance

Cosine distance is a metric for the similarity of two vectors. It measures similarity with calculating the cosine of the angle between them. It has positive values only. The cosine distance method is used to determine the cluster of AIS messages.

$$1 - cos(u, v) = 1 - \frac{\sum_{i=1}^{D} u_i \times v_i}{\sqrt{\sum_{i=1}^{D} {u_i}^2} \sqrt{\sum_{i=1}^{D} {v_i}^2}} \tag{33}$$

3.2.6 Silhouette Score

The Silhouette Coefficient is calculated with the mean intra-cluster distance and the mean of the distances to all the points in a cluster. The silhouette score [42] is the mean of the Silhouette Coefficient of all samples. The greatest value of Silhouette Coefficient is 1 and the least value is -1. This means the higher the score, the better the results. The Silhouette Coefficient is calculated for each sample in the cluster C_i in Equation 34

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, if|C_i| > 1 \tag{34}$$

Where a is average distance of i to all points in the same cluster and b is average distance of i to all points in the nearest cluster.

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j) \tag{35}$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \tag{36}$$

3.3 Neighborhood Search Method

To implement the Neighborhood Search Method algorithm Haversine Distance, Geodesics Inverse and Direct Solutions and DBSCAN clustering algorithm are used.

3.3.1 Bearing Calculation

Bearing is the horizontal angle between the true north and the direction of an object. The bearing between a and b, is calculated with equation 37.

$$\begin{aligned} x &= cos(lat_b) * sin(lon_b - lon_a) \\ y &= cos(lat_a) * sin(lat_b) sin(lat_a) * cos(lat_b) * cos(lon_b - lon_a) \\ \beta_{a,b} &= atan2(x, y) \end{aligned} \tag{37}$$

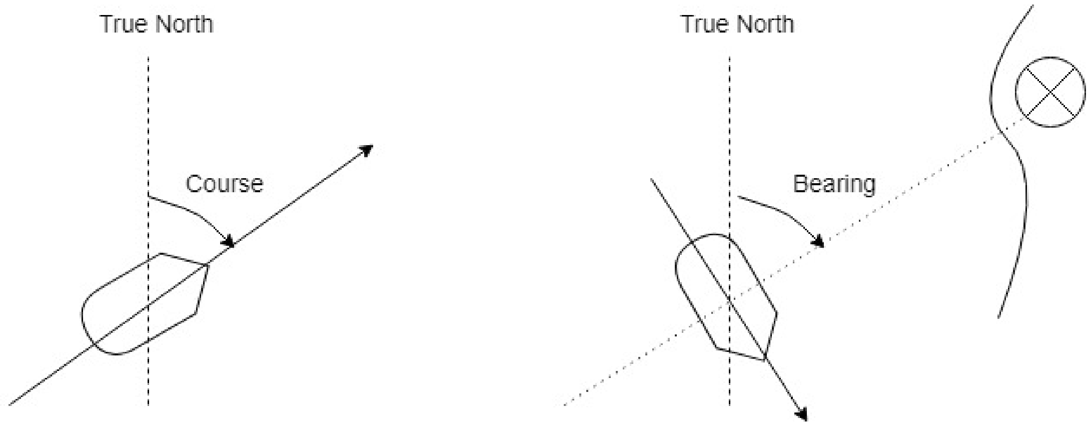


Figure 3.4: The Difference Between Course and Bearing

3.3.2 Haversine Distance

Haversine Distance is used for measuring the distance between two points on the surface of Earth which is called Great Circle Distance.

$$d = 2rsin^{-1}(\sqrt{sin^2(\frac{\phi_2 - \phi_1}{2}) + cos(\phi_1)cos(\phi_2)sin^2(\frac{\psi_2 - \psi_1}{2})}) \tag{38}$$

3.3.3 Direct and Inverse Solutions of Geodesics on The Ellipsoid

Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations[43] presents the direct and inverse solution of geodesic length formula. The direct formula calculates the latitude and longitude of a new position with the starting point, distance, and course. The inverse solution presents a calculation of the angle between two locations. The direct solution formula is shown in Equation 39 and the inverse solution formula is in Equation 310.

Variables	Statement
a	length of major axis of the ellipsoid
f	flattening of the ellipsoid
b	minor axis of the ellipsoid; For Earth
ϕ_1, ϕ_2	latitude of the points
$U1$	$arctan((1 - \ f)tan\phi_1)$
$U2$	$arctan((1\ f)tan\phi_2)$
$L1, L2$	longitude of the points
$L = L2 - \ L1$	difference in longitude of two points
λ	Difference in longitude of the points
$\alpha1, \alpha2$	forward azimuths at the points
α	forward azimuth of the geodesic at the equator
s	distance between the two points
σ	angular separation between points
$\sigma1$	angular separation between the point and the equator
σm	angular separation between the midpoint of the line and the equator

Table 3.1: The Variables in Geodesic Formulas

$$\begin{aligned}
&U_1 = \arctan[(1-f)\tan\phi_1] \\
&\sigma_1 = \text{arctan2}(\tan U_1, \cos\alpha_1) \\
&\sin\alpha = \cos U_1 \sin\alpha_1 \\
&u^2 = \cos^2\alpha(\frac{a^2-b^2}{b^2}) = (1-\sin^2\alpha)(\frac{a^2-b^2}{b^2}) \\
&A = 1 + \frac{u^2}{16384}(4096 + u^2[-768 + u^2(320 - 175u^2)]) \\
&B = \frac{u^2}{1024}(256 + u^2[-128 + u^2(74 - 47u^2)])
\end{aligned} \tag{39}$$

$$
\begin{aligned}
&\sin\sigma = \sqrt{\left(\cos U_2 \sin\lambda\right)^2 + \left(\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos\lambda\right)^2} \\
&\cos\sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos\lambda \\
&\sigma = \operatorname{arctan2}\left(\sin\sigma, \cos\sigma\right) \\
&\sin\alpha = \frac{\cos U_1 \cos U_2 \sin\lambda}{\sin\sigma} \\
&\cos\left(2\sigma_{\mathrm{m}}\right) = \cos\sigma - \frac{2\sin U_1 \sin U_2}{\cos^2\alpha} = \cos\sigma - \frac{2\sin U_1 \sin U_2}{1-\sin^2\alpha} \\
&C = \frac{f}{16}\cos^2\alpha\left[4 + f\left(4 - 3\cos^2\alpha\right)\right] \\
&\lambda = L + (1-C) f \sin\alpha\left\{\sigma + C\sin\sigma\left[\cos\left(2\sigma_{\mathrm{m}}\right) + C\cos\sigma\left(-1 + 2\cos^2\left(2\sigma_{\mathrm{m}}\right)\right)\right]\right\}
\end{aligned}
\tag{310}
$$

After λ reach to designed accuracy, the equations in 311 have evaluated.

$$
\begin{aligned}
&u^2 = \cos^2\alpha\frac{a^2-b^2}{b^2} \\
&A = 1 + \frac{u^2}{16384}(4096 + u^2 - [768 + u^2(320 - 175u^2)]) \\
&B = \frac{u^2}{1024}256 + u^2[-128 + u^2(74 - 47u^2)]) \\
&\Delta\sigma = B\sin\sigma\{\cos(2\sigma_{\mathrm{m}}) + \frac{1}{4}B(\cos\sigma[-1 + 2\cos^2(2\sigma_{\mathrm{m}})] - \\
&\quad \frac{B}{6}\cos[2\sigma_{\mathrm{m}}][-3 + 4\sin^2\sigma][-3 + 4\cos^2(2\sigma_{\mathrm{m}})])\} \qquad tips \\
&s = bA(\sigma - \Delta\sigma) \\
&\alpha_1 = \operatorname{arctan2}(\cos U_2 \sin\lambda, \cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos\lambda) \\
&\alpha_2 = \operatorname{arctan2}(\cos U_1 \sin\lambda, -\sin U_1 \cos U_2 + \cos U_1 \sin U_2 \cos\lambda)
\end{aligned}
\tag{311}
$$

3.4 Long Short Term Memory

Long Short Term Memory[17] (LSTM) is proposed by Hochreiter and Schmidhuber in 1997. LSTM networks remember long period information. LSTM units are built up with cells, input gates, forget gates and output gates. The cells are the memory units. The input cells are responsible for which values come into the cell. The forget gates decide which values remain. The output gate layer controls which values used for calculation of the output. The information flows into these three connections. The weights of the connections have been calculated during the training.

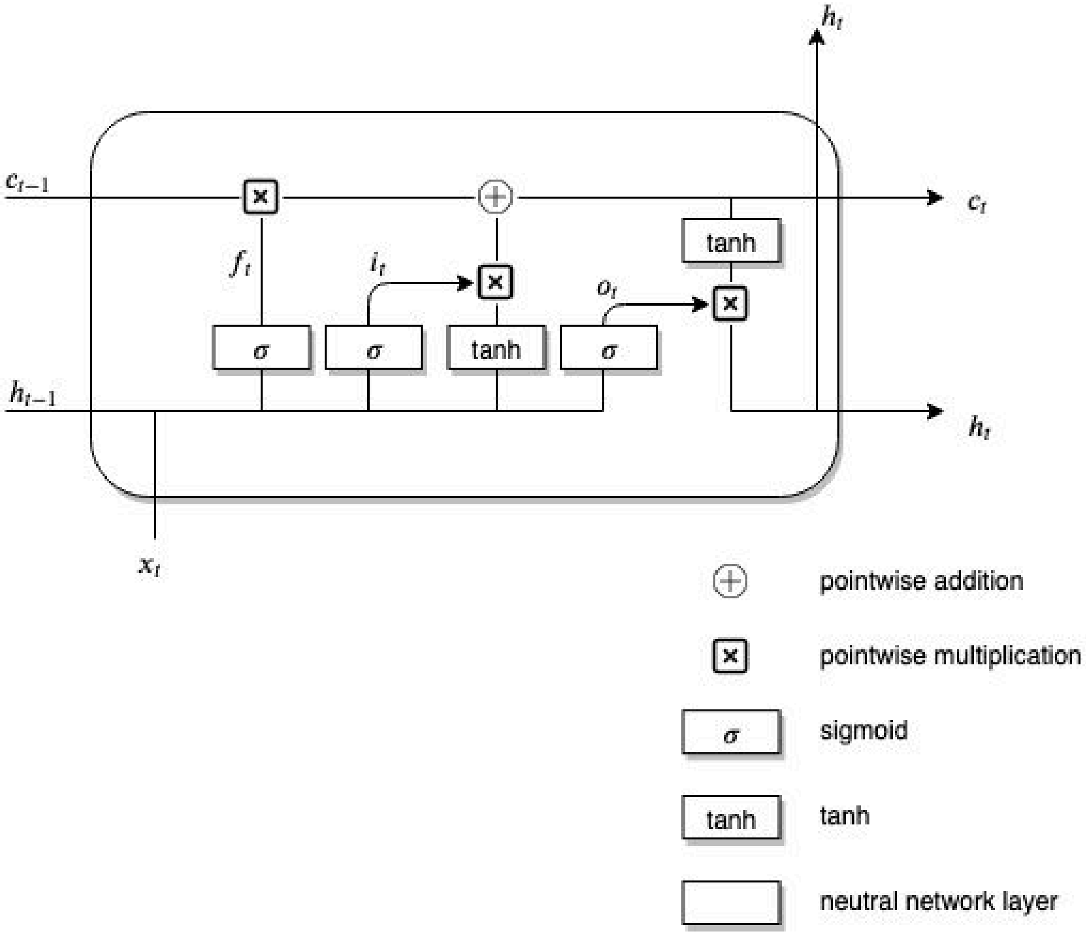


Figure 3.5: Long Short Term Memory Architecture

- $x_t \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in \mathbb{R}^h$: forget gate's activation vector
- $i_t \in \mathbb{R}^h$: input/update gate's activation vector
- $o_t \in \mathbb{R}^h$: output gate's activation vector
- $h_t \in \mathbb{R}^h$: hidden state vector also known as output vector of the LSTM unit
- $c_t \in \mathbb{R}^h$: cell state vector
- $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h} andb \in \mathbb{R}^h$: weight matrices and bias vector parameters which are learned during training

In the first step of an LSTM unit, The forget gate decides which input values are forgotten from the cell. x_t and h_{t-1} are used for the decision. The forget gate outputs 0 means forgetting or 1 means keeping the information. Next, the input gate decides which new information is updating in the cell. The tanh layer builds a new candidate vector. Afterwards, the previous cell state c_{t-1} has been updated into c_t. Lastly, in the output layer, the output has been determined based on the cell state.

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
c_t &= f_t * c_{t-1} + i_t * \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
h_t &= o_t \sigma_h * (c_t)
\end{aligned} \tag{312}$$

3.5 Anomaly Detection

In this section, the Longest Common Subsequence [35] algorithms are used for abnormal trajectories detection. The extracted trajectories are saved in the database. A new position sequence is gathering. The operators want to know that the new sequence contains abnormal movements. To solve this problem, LCS and LCSR algorithms are used.

3.5.1 Longest Common Subsequence

Longest Common Subsequence [35] algorithm detects common subsequences in the given two sequences.

There are $x[1...m], y[1...n]$ and $x = [abcdcab]$ and $y = [bcad]$. Both $[[a]bcdcab]$ and $[[b]cad]$ are compared. They are different. Then move on next item. In comparison of $[a[b]cdcab]$ and $[[b]cad]$, the items are the same. Therefore, b is the first element of the subsequence. In $[ab[c]dcab]$ and $[b[c]ad]$, they are equal. So items are added to subsequence and its final form is become bc. $[abc[d]cab$ and $[bc[a]d]$ are not equal. The index is moving on to next item. $[abcd[c]ab]$ and $[bc[a]d]$ are compared and the elements are different. In $[abcdc[a]b]$ and $[bc[a]d]$ comparison, the items are the same. The item is added to the subsequence list. The comparison of final elements $[abcdca[b]]$ and $[bca[d]]$, the items are different. The final subsequence is $[bca]$. The longest common subsequence algorithm is given in Algorithm 3 .

Algorithm 3 Longest Common Subsequence Algorithm [35]

```
Require: $X, Y$
  $m = len(X)$
  $n = len(Y)$
  for $i\ in\ range(m + 1)$ do
    for $j\ in\ range(n + 1)$ do
      if $i == 0\ or\ j == 0$ then
        $L[i][j] = 0$
      else if $X[i - 1] == Y[j - 1]$ then
        $L[i][j] = L[i - 1][j - 1] + 1$
      else
        $L[i][j] = max(L[i - 1][j], L[i][j - 1])$
      end if
    end for
  end for
  return $L[m][n]$
```

3.5.2 Longest Common Subsequence Ratio

Longest Common Subsequence Ratio [44] method is derived from LCS algorithm. LCSR calculates LCS with a ratio. To achieve this, the result of LCS is divided into the maximum length that sequences have. LCSR is calculated with equation 313. But this algorithm has been modified. As the problem is detecting anomaly routes, minimum of extracted routes or querying sequence is used for calculation of ratio.

$$LCSR(X, Y) = \frac{LCS(X, Y)}{minLen(X, Y)} \tag{313}$$

3.6 Evaluation

There are different evaluation metrics that we use in this study, such as Mean Absolute Error, Accuracy, Precision, Recall and F1 score. First, the metrics need definition of true positive, true negative, false positive and false negative.

True Positive means that the model predicts the right value and it is the actual value as well.

False Positive is that the model predicts the right value, but it is not the real value.

True Negative means that the prediction is wrong and it is actually wrong.

False Negative is that the prediction is wrong. However the actual value is right.

3.6.1 Mean Absolute Error

Mean Absolute Error shows that how prediction results are close to actual ones. The negative value of the difference becomes positive due to absolute value. The smaller MAE means better prediction.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(|Y_i - \overline{Y_i}|) \tag{314}$$

3.6.2 Accuracy

Accuracy demonstrates the percentage of the right prediction in overall data. Its formulation is presented in Equation 315.

$$Acc = \frac{TruePositive + TrueNegative}{TotalData} \tag{315}$$

3.6.3 Precision

Precision shows how accurate the model predicts. Precision is a ratio of true positives overall positive predictions.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{316}$$

3.6.4 Recall

The recall is an evaluation of how sensitive the model predictions.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{317}$$

3.6.5 F1 Score

F measure gives the balance of the precision and recall results. It is in the range of 0 and 1. F1 score equation is represented in Equation 318.

$$F1Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{318}$$

CHAPTER 4

PROPOSED METHODS

In this chapter, the implementations for route prediction and anomaly detection have been presented. The implementation language is determined as Python, because of the easy to use and the libraries it contains. All data are stored in MongoDB. Indexes service for rapid interactions. MongoDB [38] has geospatial features. Geospatial data has been stored as GeoJSON [45, 46] objects in WGS-84 [26, 27] format. Due to the need for the geospatial queries, the 2dsphere indexes that provide querying on an earth-like sphere have been applied in this work.

4.1 Movement Prediction

In this section, trajectory prediction methods are explained. First, AIS message clustering is proposed to determine the vessel routes. Following that, LSTM methods are explained.

4.1.1 AIS Message Clustering

The clustering methods have been used to extract vessel trajectories. In the beginning, latitude and longitude from AIS messages are clustered with different algorithms. Due to that, the density is crucial to extract routes, a density-based algorithm, DBSCAN [32] is used. After the clustering process, a classification algorithm has been applied. The AIS messages in each cluster are used as an input for the classification algorithm. To determine the supervised learning algorithm, several classification algorithms have been tested with AIS data. The Random Forest Algorithm gives the best results. The results are presented in Chapter 5. After classifying data, routes have been predicted for new locations.

Algorithm 4 AIS Message Clustering Algorithm

```
$data \leftarrow readAISMessagesFromDB$
$train \leftarrow data[Lat, Lon]$
$dbscan.fitPredict(train)$
for $cluster$ in $clusters$ do
    $trainX \leftarrow Lat, Lon$
    $trainY \leftarrow ArrivalPort$
    $randomForest.fit(trainX, trainY)$
end for
```

4.1.2 Long Short Term Memory Method

Long Short Term Memory [17] method has been applied in several recent problems as a solution. For instance, in time-based series, LSTM gives great results. In Shi, Zhiyuan, and Xu's study [23], flight trajectories are predicted using the LSTM network in keras [47]. In the study, the sliding windows method has been applied. As an input, speed, heading and location information is been used together. The length of the window is set experimentally. The length of the window has been defined as 10. The LSTM network has four layers. Two layers are the hidden network layers with 30 and 60 neurons respectively. The output layer has a single neuron. The activation function is linear. The dropout ratio is 0.2. As the optimizer, RMSprop [18] is applied.

The LSTM network model in flight trajectory prediction [24] is used for prediction of vessel trajectory in this study. The AIS messages data is separated by 50 messages to create sliding windows. While building sliding windows, the same ship's AIS messages are used. The layers of the LSTM network are the same as the flight trajectory prediction LSTM model. The results of the first LSTM attempt are in Chapter 5. Although the results of predictions for flight trajectories are good, the LSTM model in the study is not suited well for vessel trajectories. Therefore, a set of experiments have been made to enhance the LSTM model.

To begin with, the different activation function has been tested, such as Sigmoid and Relu [48]. Moreover, the length of the sliding window has been increased to 50 instances. In this version of LSTM, the input is normalized with min-max normalization. Each 50 instance has latitude, longitude, speed, course and time information. At the next position, latitude, longitude, speed, course and time are predicted with the model. Even though the actual values are given into the sliding window model, the predicted values are used to predict the next values. The model has eight layers that contain two hidden layers. The first hidden layer has 30 neurons and the second one has 60 neurons. After each hidden layer, there are activation functions and dropout layers that have the 0.2 ratio. The activation functions are Relu. There is a dense layer that has one neuron. The last activation function is linear. As an optimizer, RMSprop has been used.

Furthermore, the arrival port is predicted using the LSTM method. Since the arrival port is a categorical value, the arrival port column is normalized with one hot encoder method. Other values are normalized with the min-max normalization method. To predict arrival port values, latitude, longitude, speed, course and time information are used as the input data. Like other LSTM implementations, the input data is split into sequences. The sequences have 50 instances. With each sequence, the arrival port is forecast. The model is built up with 2 hidden layers that have 30 and 60 neurons, respectively. Additionally, there are two dropout layers between LSTM layers. The one neuron dense layer is the final layer. The activation function is linear. RMSprop is the optimizer. The results are presented in Chapter 5.

Finally, the next position of latitude and longitude values are predicted separately. The input data is normalized with min-max normalization. The data is divided into 50 each part. Each part of the data contains latitude or longitude information along with speed, course and time. With 50 instances, the model predicts the next latitude or longitude. The model for latitude and longitude prediction has 6 layers that contain 2 hidden layers. After each hidden layer, there are dropout layers that have 0.2 ratios. There is a dense layer that has one neuron. Lastly, the activation function is linear. As an optimizer, RMSprop has been used.

4.2 Anomaly Detection

First of all, the Neighborhood Search Method is explained. The algorithm is used for routes extraction. Later, there are two anomaly types examined. The first one is dissimilarity with extracted routes. The second anomaly is the unusual stop trajectories in the open sea.

4.2.1 Neighborhood Search Method

Neighborhood Search Method is a method for the prediction of vessel routes. The algorithm is based on TREAD [10] method. There are two phases of the method. Initially, waypoints have been detected. After that, the centers of the waypoints need to be determined. The centers state ports or stationary points. There is a route between two waypoints. The route is calculated by the synthetic route generation algorithm [10].

Algorithm 5 Route Generator Algorithm [10]

```
**Require:** $R_c^k, c, WP_1, WP_2, step_t, \epsilon$
  $[xp_1, yp_1] \leftarrow centroid(WP_1)$
  **while** $not(inpolygon[xp_{end}, yp_{end}], WP_2)$ **do**
    find $l$ in s.t $\forall l \in l : ||\ R_c^k(l) * [x, y] - [xp_{end}, yp_{end}]||\ \leq \epsilon$
    $[x\dot{p}_{end}, y\dot{p}_{end}] \leftarrow median(Sp)$
    $[xp_{end+1}, yp_{end+1}] \leftarrow [xp_{end}, yp_{end}] + [x\dot{p}_{end}, y\dot{p}_{end}] * step_t$
  **end while**
  **return** $[xp, yp]$
```

Finding waypoints is the first step. To detect waypoints, the speed feature has been selected. If speed value is lower than 0.1 knots, the point has marked as a waypoint. Moreover, in this study, not only AIS speed value is used but also it is calculated. For speed calculation, the distance and time have been extracted between the current and last position. The distance has been calculated by the Haversine distance. After determining the length, the elapse time has been extracted by timestamp fields. Finally, the speed value is obtained by the division of distance and time. If the calculated speed value or AIS based speed value lower than 0.1 knots, then the point becomes a waypoint.

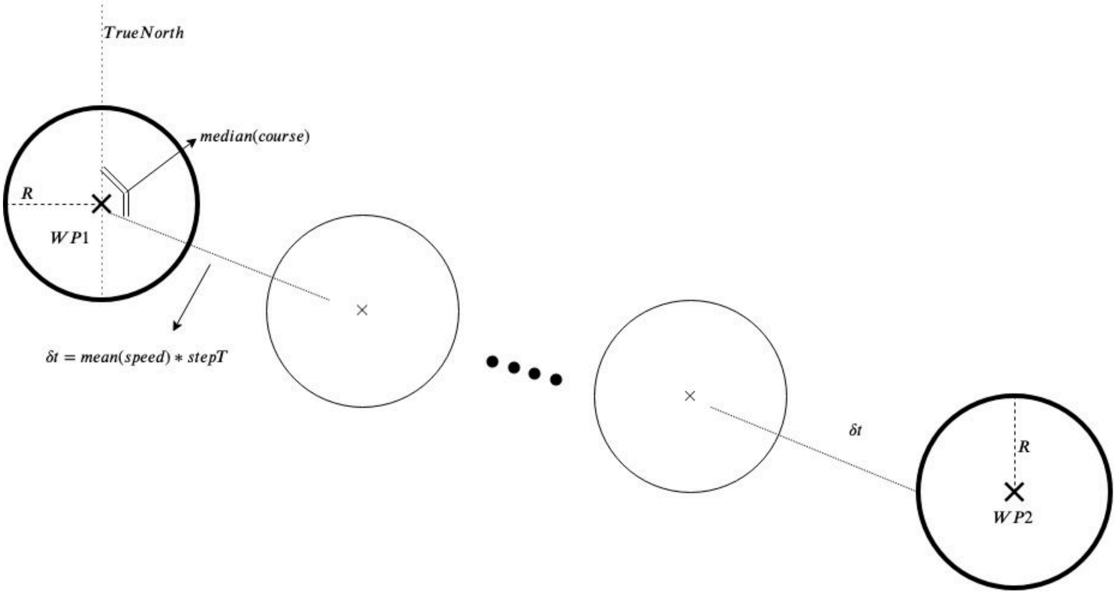


Figure 4.1: Neighborhood Search Method

After detecting the waypoints, they need to cluster to find stationary points. In the clustering process, the DBSCAN [32] algorithm has been used. The routes have been generated between each cluster center.

The vessel routes occur between two centers of waypoints. At the beginning of the algorithm, two points are needed. Around the starting waypoint, the points have been searched in a determined radius. The next speed and course values have been extracted from these points. After obtaining the mean of speeds and the median of course values, the waypoint has been taken forward to the next location. This iterates until reaching the last waypoint.

To improve this algorithm, in this study, two different approaches have been applied. First of all, the bearing between starting and arriving waypoints has been calculated. Thus the ships which are in the opposite direction have not included in the calculation. The second approach is moving the ships using time. Each AIS message has timestamp information. In the calculation, besides next position, the time is also checked. If the ships are not locating the calculated position and time, then they have not considered.

Roughly, the bearing is the angle of direction from one ship to another. In the open sea, there could be several moving ships between two waypoints. To find the route from waypoint 1 to waypoint 2, all AIS messages are being used for calculation. In this approach, the missing AIS messages are interpolated with the former message values. However, we have realized that there are arriving ships from wp1 and returning ship to wp1. The returning ships have caused the algorithm to work incorrectly. Therefore, the bearing from wp1 to wp2 is calculated initially. We have achieved the limit values by adding the bearing value +90 and -90. In Figure 4.2, the first state contains wrong way messages. In the second state, we use only the messages that are coming in the range of calculated bearing. However, the routes that begin with the opposite bearing and continue within the limits could not be extracted. For the solution, we move the ship in the timeline.

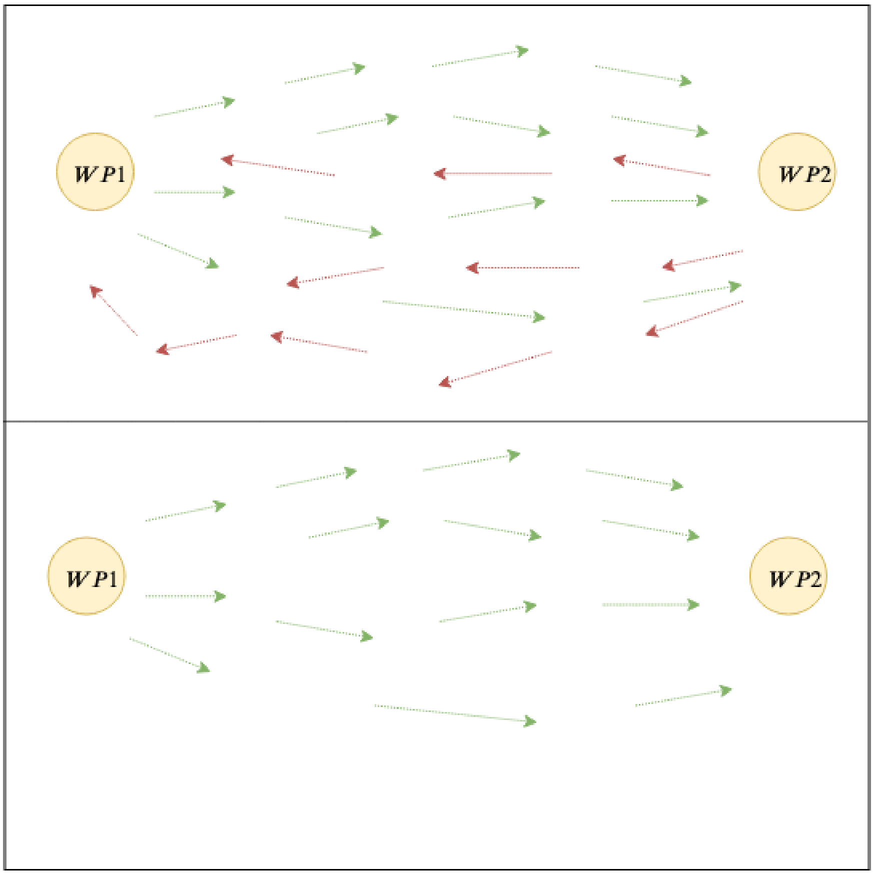


Figure 4.2: Bearing In Neighborhood Search Method

In the time-based approach, the AIS messages are controlled whether they are coming at the right time interval. To achieve this, in each leg of the route, the timestamps of the messages are examined. Initial timestamp values are derived from AIS messages for each ship. Later, using time interval that is given as a parameter to the algorithm, next timestamps are calculated. If the timestamps are in line, then the locations are added as a leg of the route. The projection of method is in Figure 4.3.

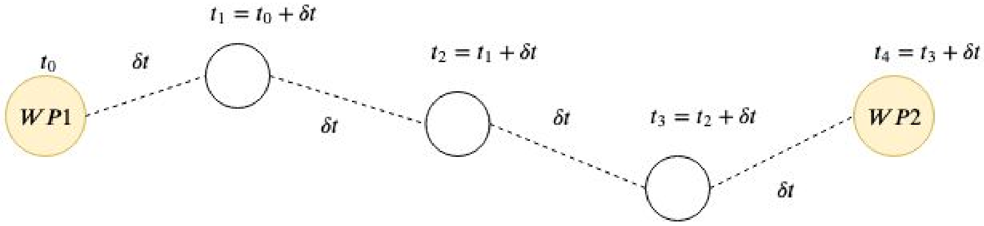


Figure 4.3: Time-Based Method In Neighborhood Search Method

Additionally, we have combined this approach with course clustering. The courses of ships are clustered using K Means to determine the initial inclination value. After clustering the course values in the waypoint, initial course values become the centroid of each cluster.

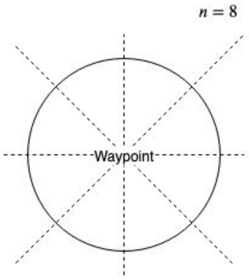


Figure 4.4: The Course Clustering In A Waypoint

In this method, we can find multiple routes between two waypoints. This is because the starting courses are different. Each course value is executed in itself. Afterward, while calculating the next course value from the median of near AIS message, the timestamps values are controlled.

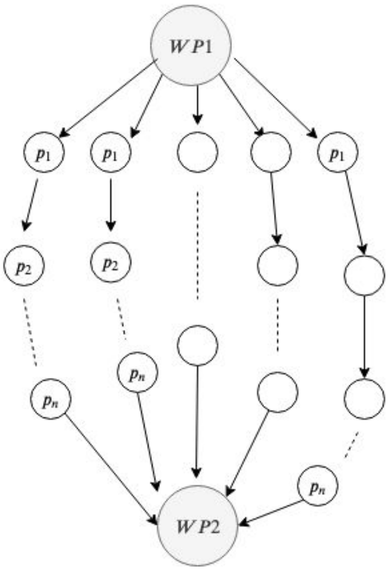


Figure 4.5: The Multiple Trajectories Between Two Waypoints

4.2.2 Dissimilarity with Extracted Trajectories

There is a need to measure the similarity of a new position sequence to the extracted paths. The LCS algorithm is used to evaluate the similarity of two trajectories in this work. The similarity between trajectories is the similarity between the waypoints of the routes. If the waypoints match in the determined radius, then they are accepted as similar. The radius is 10 km in the work. Because the same distance metric is used while trajectory extraction. For instance, T has five waypoints that are $wp_1, wp_2, wp_3, wp_4\ and\ wp_5$. Besides, S consists of four waypoints that are $\overline{wp_1}, \overline{wp_2}, \overline{wp_3}\ and\ \overline{wp_4}$. Like, there are two sequences that are $T = ABCDE$ and $S = ACD$. The LCS ratio is 0.6 for these sequences. For the vessel trajectory similarity, there is a radius for each waypoint. wp_1 and $\overline{wp_1}, wp_3$ and $\overline{wp_2}$ and $\overline{wp_3}$, wp_4 are close to each in both trajectory. Therefore, S and T are similar in 0.6 ratio. The next waypoint of T is expected to similar to the next point in S. The results are discussed in Chapter 5.

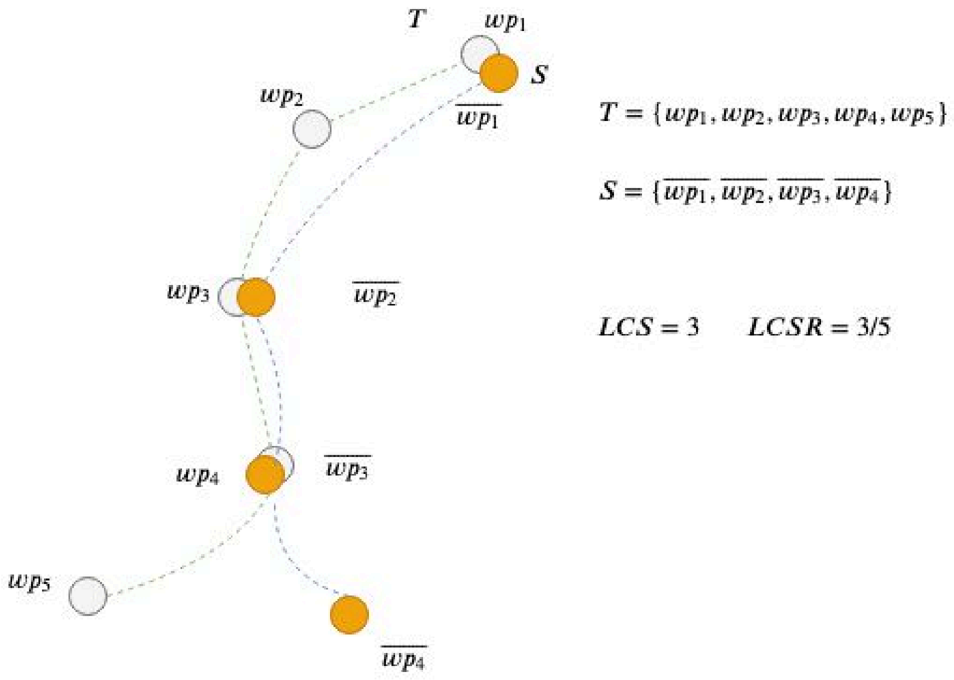


Figure 4.6: The Similarity Between Trajectories

4.2.3 Unusual Stop

In open sea, it is unexpected that the ships stop. As shown in Figure 4.7, the speed value of the moving ship is getting zero suddenly. As in Handayani, Sediono, and Shah's work [37], this situation is regarded as abnormal behavior. However, the routes that ships have usually stopped can be extracted using Neighborhood Search Method. In the algorithm, the trajectories that ships use generally are desired to discover. With the initial waypoint, the mean speed value is calculated. The same process is followed at the next waypoints. However, if the calculated speed value becomes zero, then the trajectory is tagged as unusual stop.

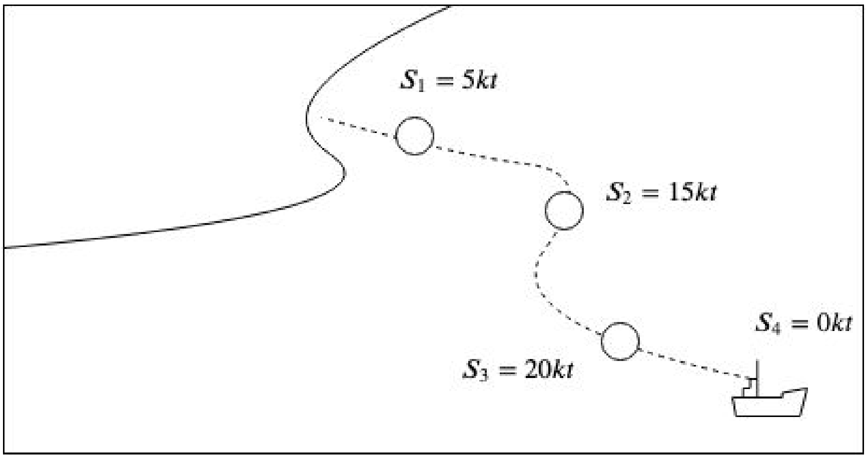


Figure 4.7: Unusual Stop Anomaly

CHAPTER 5

EXPERIMENTS

In Chapter 5, there are results of prediction and anomaly detection methods. In the prediction part, initially, data preprocessing methods are explained. Later, the prediction methods are described. In the last part, there are results of the detection of anomalies that are detected with two different approaches.

5.1 Movement Prediction

In this section, several prediction algorithm results are presented. Initially, data preprocessing is applied. Before the prediction phase, different kinds of clustering and classification algorithms are tested on the data. As a prediction method, the clustering, and LSTM [17] have been applied.

5.1.1 Data Preprocessing

Different kinds of classification methods have been experienced to produce an effective result that is used for eliciting existing route patterns. Although the data contains AIS messages from the Mediterranean sea, it is dense in Europe coasts. The database contains 542153 AIS messages. Furthermore, there are 503 unique ships. Although the most of vessels are cargo ships, there are 20 types of vessels. The distribution of them is in Figure 5.2.

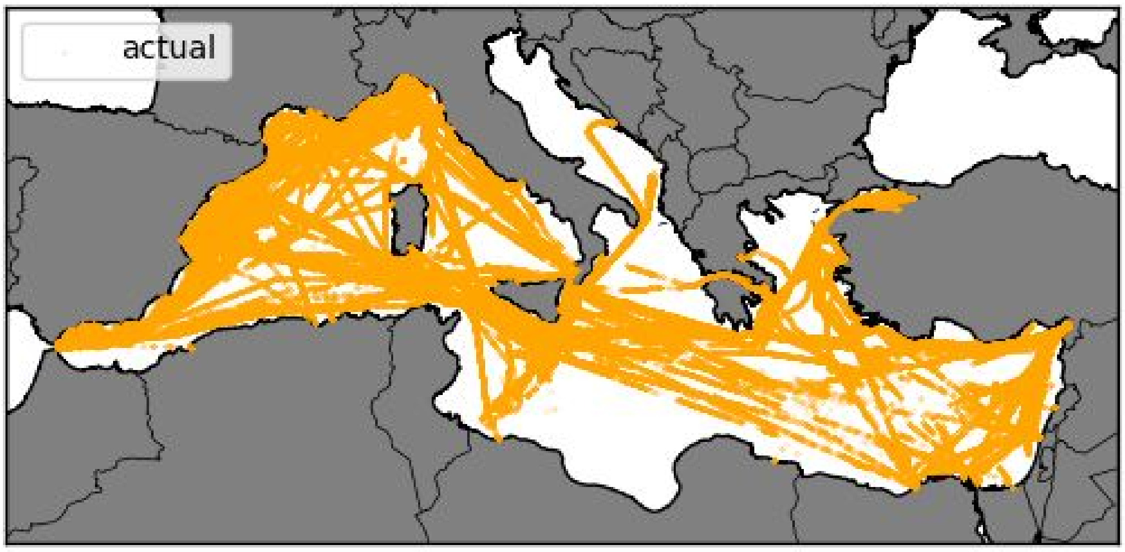


Figure 5.1: Projection of all AIS messages.

Table 5.1: Dataset Specification.

The Number of Message	542153
Time Frame	From 10 March to 19 May 2015
The Number of Ships	503
The Number of Ship Types	20

At the beginning, the data has noisy messages. There are null messages and fields that the dataset consists. First, the null messages are eliminated from the dataset. Moreover, the empty fields, like reported draught are removed. Also, ship type is not necessary for extraction patterns in the first stage. Heading indicates the direction in which way a vehicle is pointing. Therefore, it is continuously varying. Instead of heading value, using the course value is more useful. The timestamp field is used for prediction.

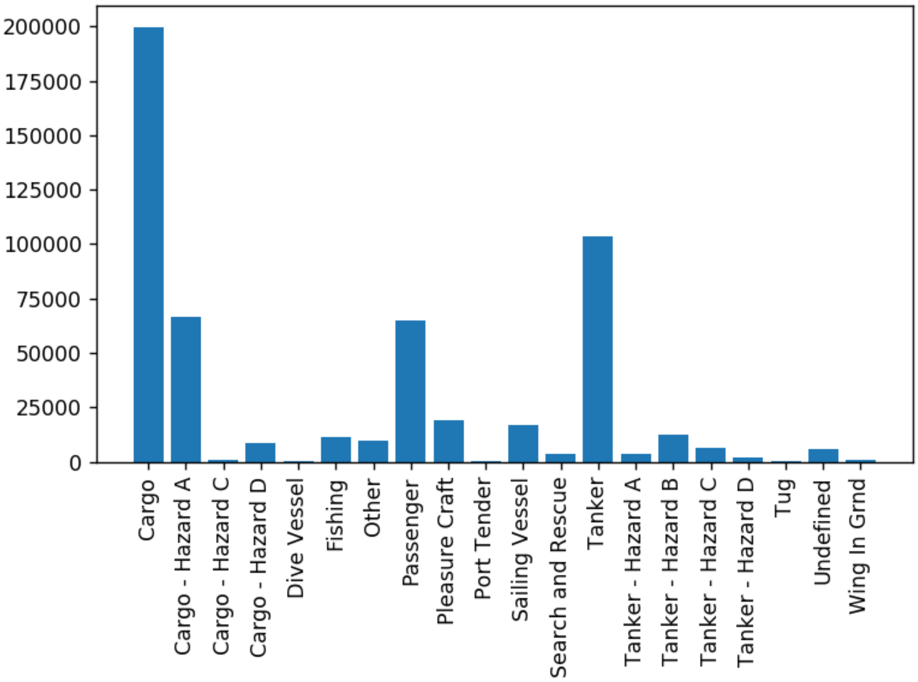


Figure 5.2: Distribution of Ship Types.

5.1.2 Prediction

In the prediction part, there are two different methods, such as clustering approach and Long Short Term Memory algorithm. First, we have tried different supervised and unsupervised algorithms.

After the cleaning process, several attempts have been made to arrival port prediction. First, various classifier algorithms are applied in data. In these classification tests, cross validation method has been applied with 10 fold. The arrival port is used as the prediction label, whereas latitude, longitude, speed and course is used for prediction input. In Table 5.2, the results are given.

Table 5.2: Arrival Port Prediction Accuracy.

Algorithm	Accuracy
K Nearest Neighbors	0.58
Decision Tree	0.61
Random Forest	**0.65**
MLP Classifier	0.53
Ada Boost	0.18
Gaussian NB	0.32
Quadratic Discriminant Analysis	0.01
Logistic Regression	0.30
Linear Discriminant Analysis	0.31
Extra Trees Classifier	0.64
Bagging Classifier	0.62

In this study the Random Forest is used as classification algorithm for the reason that the Random Forest algorithm gives the best prediction result. Also, arrival time prediction has been done. In the arrival time prediction test, several classification methods have been tested. As a label, arrival time has been used. The prediction is made with latitude, longitude, speed and course information.

Table 5.3: Arrival Time Prediction Accuracy.

Method	Accuracy
K Nearest Neighbors	0.0002 (n=3)
Decision Tree	0.75
Random Forest	**0.79**
MLP Classifier	0.024
Ada Boost	0.027
Gaussian NB	0.029
Quadratic Discriminant Analysis	0.0008
Logistic Regression	0.024
Linear Discriminant Analysis	0.37
Extra Trees Classifier	**0.80**
Bagging Classifier	0.77

Grid based approach is highly used in trajectory classification. In this method, the selected area split into grids. For each latitude and longitude pair, current grid is calculated. Firstly, the Mediterranean is split into 15 grids. After that, arrival port is predicted by using KNN, Decision Tree and Random Forest classifiers. Because of the low accuracy, the selected area is split into more grids. The area is split into 150 grids. There are 16 columns and 11 rows in the bounding box. The same classifiers are used for predicting the arrived port. In 150 grids split, the accuracy is higher than 15 grids. To analyze the results, the bounding box has been split different count of grids. The results are shown in Table 5.4.

Table 5.4: Arrival Port Prediction Accuracy in Grid Base Test

Classifier/Size of Grids	8	15	80	150	238	Point Based
K Nearest Neighbors	0.534	0.537	0.558	0.556	0.562	0.589
Decision Tree	0.575	0.573	0.590	0.591	0.598	0.619
Random Forest	**0.598**	**0.597**	**0.615**	**0.619**	**0.622**	**0.656**

In another test, to classify data speed, course, grid number attributes are used. In

Table 5.5, the results are shown that classified with speed, course, area, departure port attributes. According to results, with departure Port, the classifiers gives higher accuracy precision. Hence, these attributes are used for other classifiers. According to the results, point based approach gives more accurate results. Because of splitting the bounding box into 15 grids, many arrived ports have been placed in the same grid. Therefore, arrival port can not been classified correctly. It causes low accuracy ratio.

Table 5.5: Arrival Port Prediction Accuracy with Second Grid Base Test

Classifier	**Gird Size=150**
K Nearest Neighbors	0.56
Decision Tree	0.59
Random Forest	**0.61**
MLP Classifier	0.49
Ada Boost	0.25
Gaussian NB	0.30
Quadratic Discriminant Analysis	0.32
Logistic Regression	0.25
Linear Discriminant Analysis	0.27
Extra Trees Classifier	**0.61**
Bagging Classifier	0.60

After Grid based approaching, arrival time has been predicted with timestamp and departure port, speed, course, grid number that are calculated previous section. K Nearest Neighbour, Decision Tree and Random Forest are been applied to data. The results are presented in Table 5.6.

Table 5.6: Arrival Time Prediction Grid Base Model

Classifier	Prediction Accuracy
K Nearest Neighbors	0.0598
Decision Tree	0.806
Random Forest	**0.846**

For another prediction test, speed, course, departure port and days are used. Days information is calculated by subtraction of arrival time and current time that means number of day to arrival. The results are given in 5.7.

Table 5.7: Arrival Time Prediction Accuracy

Classifier	Prediction Accuracy
K Nearest Neighbors	0.621
Decision Tree	0.659
Random Forest	**0.678**

Different classifiers have been used to predict arrival port with first 100 messages. The first attribute set contains departure port, speed, course, latitude and longitude fields. The second contains departure port, speed, latitude, longitude. The third set contains departure port, course, latitude, longitude. departure port, latitude, longitude fields are used in the fourth attribute set. Latitude, longitude attributes are used in the fifth attribute set. The last set contains only departure port. The model sets can be listed as;

- A = [departure port, speed, course, latitude, longitude]
- B = [departure port, speed, latitude, longitude]
- C = [departure port, course, latitude, longitude]
- D = [departure port, latitude, longitude]
- E = [latitude, longitude]
- F = [departure port]

Each model has been used as input in different classification algorithms. The results of each classification are in Table 5.8

Table 5.8: Arrival Port Prediction Accuracy with Different Models

Classifier	A	B	C	D	E	F
K Nearest Neighbors	0.8712	1.0	0.8613	1.0	1.0	1.0
Decision Tree	1.0	1.0	1.0	0.9900	0.9900	1.0
Random Forest	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
MLP Classifier	0.6336	1.0	0.6336	0.9405	1.0	1.0
Ada Boost	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Gaussian NB	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Quadratic Discriminant Analysis	0.5742	0.5742	0.5742	0.5742	1.0	0.5742
Logistic Regression	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Linear Discriminant Analysis	1.0	1.0	1.0	1.0	1.0	X
Extra Trees Classifier	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Bagging Classifier	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

To predict arrival port, current gird is calculated by using ships latitude and longitude. With using each pair of latitude and longitude, current grids have been calculated. To analyze attributes, different attribute sets have been used. First set contains departure port, speed, course and grid number fields. The second set contains departure port, speed, grid number. The third set contains departure port, course, grid fields. Departure port, grid fields are used in the fourth attribute set. Grid attribute is used as a fifth attribute set. The results of this prediction is given in Table 5.9

- A = [departure port, speed, course, grid]
- B = [departure port, speed, grid]
- C = [departure port, course, grid]
- D = [departure port, grid]
- E = [grid]

Table 5.9: Arrival Port Prediction Accuracy with Models

Classifier	A	B	C	D	E
K Nearest Neighbors	0.9702	1.0	0.9702	1.0	1.0
Decision Tree	1.0	1.0	1.0	1.0	0.9900
Random Forest	1.0	0.9801	0.9900	1.0	1.0
MLP Classifier	0.7524	0.9504	0.6039	0.9306	1.0
Ada Boost	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Gaussian NB	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Quadratic Discriminant Analysis	0.5742	0.5742	0.5742	0.5742	0.4257
Logistic Regression	1.0	1.0	0.9900	1.0	1.0
Linear Discriminant Analysis	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Extra Trees Classifier	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Bagging Classifier	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

The built data models are tested with random 100 AIS messages to calculate arrival port. In Table 5.10, the prediction results are presented for arrival port prediction using latitude and longitude. The Table 5.11 shows the results of arrival port prediction with grid based data model.

Table 5.10: Arrival Port Prediction Accuracy with Different Models

Classifier	A	B	C	E	D	E
K Nearest Neighbors	0.26	0.31	0.26	0.35	0.34	0.27
Decision Tree	0.36	0.38	0.39	0.34	0.36	0.31
Random Forest	0.42	0.42	0.40	0.40	0.38	0.29
MLP Classifier	0.19	0.33	0.18	0.37	0.33	0.22
Ada Boost	0.25	0.27	0.25	0.27	0.27	0.24
Gaussian NB	0.32	0.31	0.35	0.31	0.24	0.15
Logistic Regression	0.24	0.30	0.25	0.27	0.28	0.21
Linear Discriminant Analysis	0.29	0.32	0.28	0.31	0.31	0.22
Extra Trees Classifier	**0.46**	**0.48**	**0.45**	**0.40**	**0.37**	**0.31**
Bagging Classifier	0.46	0.39	0.44	0.37	0.34	0.30

Table 5.11: Arrival Port Prediction Accuracy with Different Models

Classifier	A	B	C	D	E
K Nearest Neighbors	0.34	0.29	0.31	0.36	0.28
Decision Tree	0.43	0.38	0.31	0.27	0.31
Random Forest	0.41	0.40	0.38	0.35	0.26
MLP Classifier	0.21	0.22	0.18	0.22	0.13
Ada Boost	0.28	0.27	0.28	0.27	0.32
Gaussian NB	0.36	0.32	0.32	0.31	0.23
Logistic Regression	0.20	0.24	0.22	0.24	0.27
Linear Discriminant Analysis	0.28	0.30	0.28	0.33	0.29
Extra Trees Classifier	**0.47**	**0.44**	**0.45**	**0.32**	**0.32**
Bagging Classifier	0.36	0.40	0.41	0.29	0.26

Finally, several clustering algorithms are applied to AIS message data. The first 10000 messages are given as clustering input.

Table 5.12: Clustering Silhouette Scores with Different Models

Clustering Methods	Lat,Lon	Lat,Lon,Speed	Lat,Lon,Dept Port
AffinityPropagation	0.424	0.449	0.457
AgglomerativeClustering	0.472	0.472	0.472
Birch	0.578	0.578	0.578
DBSCAN	**0.683**	**0.747**	**0.860**
KMeans	0.668	0.674	0.777
MeanShift	0.638	0.640	0.629
MiniBatchKMeans	0.677	0.675	0.758
SpectralClustering	0.678	0.689	0.774

5.1.2.1 Clustering Method

The clustering methods are applied to detect arrival port clusters. After clustering, the classification methods are executed to catch the branches in trajectories clusters. As a clustering algorithm, DBSCAN [32] is executed. As the distance metric, Haversine distance has been used. The epsilon metric that is the radius of the clusters is tested with different values. The parameters are presented in Table 5.13.

Table 5.13: DBSCAN Parameters

Eps(in meter)	50	500	5k	**50k**	500k
Silhouette Score	-0.36	-0.11	0.51	**0.80**	0.53

At the beginning, we work with all dataset. The silhouette score for the clustering is 0.13. Then, we have decided to reduce the data. After that, the first 10000 rows are used as the input to DBSCAN. As the test data, 1000 rows selected randomly.The latitude and longitude information are used in input array. DBSCAN is clustered the latitudes and longitudes as their density. The score for this clustering is 0.80. In the first 10000 latitude and longitude are clustered as in Figure 5.3.

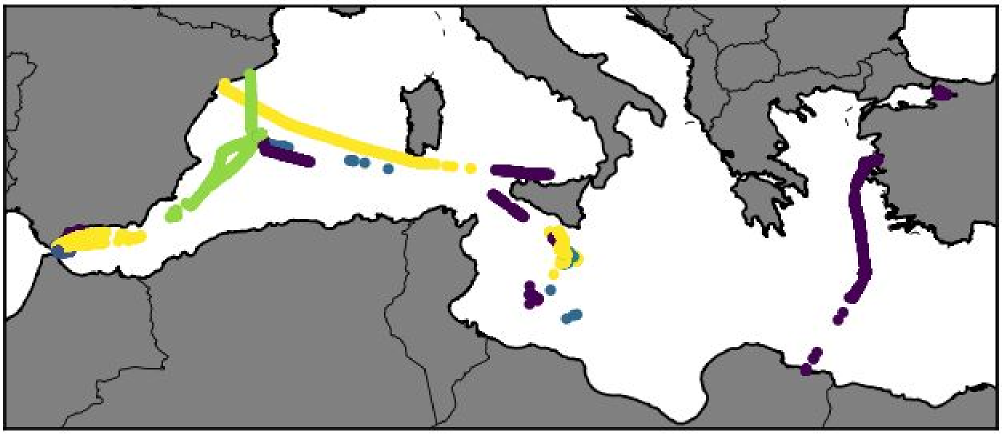


Figure 5.3: The result of 10000 rows clustering with DBSCAN

After observing the clusters, the classification algorithm, Random Forest [41] is applied. The purpose of the classification, revealing the route branches in one trajectory cluster. In the 10000 row clustering, there are 10 clusters. The data of each cluster is used as the input to Random Forest. The latitude, longitude, speed and course attributes are classified according to the arrival ports. In Figure 5.4, the route branches in Gibraltar are shown.

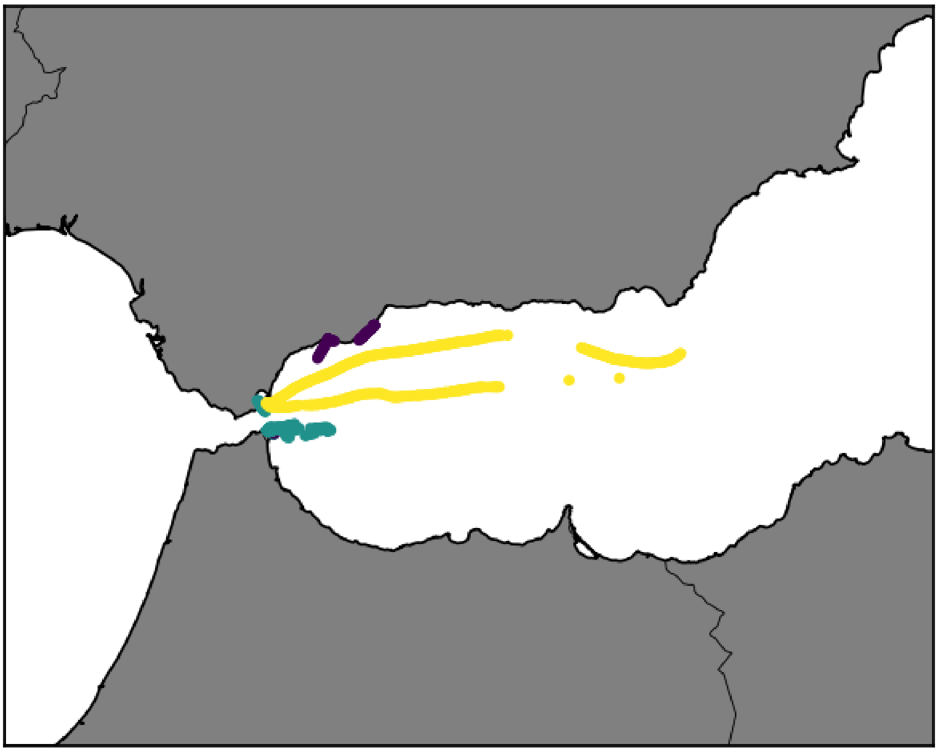


Figure 5.4: The Route Branches in Gibraltar.

Finally, 1000 rows data are used to test the elicited the trajectory branches. The test data is not used as inputs of the classification. The test data is read row by row. Then, the haversine distance is calculated between the centroid of clusters and position in each row. The cluster that have the least distance is detected as the belonged cluster. Afterwards, the arrival port is determined with the classification models that the cluster has. The F1 score for the 1000 test data is 0.37 and the accuracy is 0.49. This method compares with LSTM arrival port prediction in Evaluation part.

5.1.3 Long Short Term Memory Method

The LSTM is applied in three different ways, such as classification of arrival port, prediction of the next position of the given sequence and prediction of next position sequences according to the given one. We use mean squared error as the metric in the next position and next sequence prediction. The data is split into sequences per each ship. Every sequence includes 50 AIS messages. The train and test data are divided as the ratio of 0.67 and 0.33 respectively. For arrival port prediction, the metric is accuracy. To determine the best version of the model, we compare metric results for each epoch, then we save the best one. After the running process complete, the best prediction model is loaded. To classify the arrival port, the model is built up with two LSTM layers, one dense layer, and one activation layer. The input data consists of latitude, longitude, speed, course and time information. The time is elapsed time between two AIS messages. The loss graphic of the classification is presented in Figure 5.5. The example of prediction is displayed in Figure 5.6.

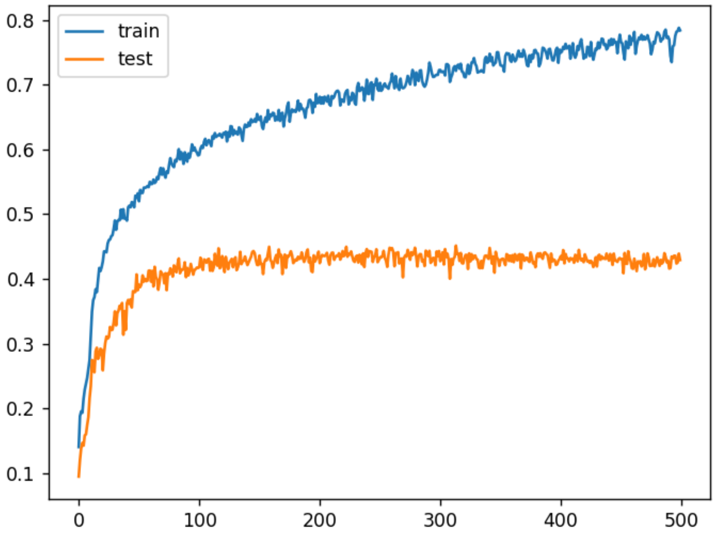


Figure 5.5: Arrival Port Train/Validation Accuracy Over Epochs

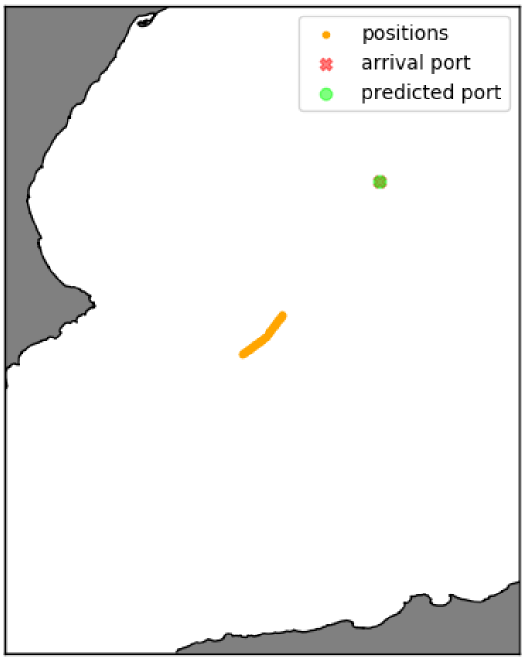


Figure 5.6: The Arrival Port Prediction Example

To compare the first arrival port prediction method with LSTM based port prediction method, a test data has been used. The test 1000 AIS messages have eight destination ports that are Barcelona, Palma De Mallorca, Valletta, Gibraltar, Tuzla, Marsaxlokk, Iskenderun, and Livorno. To evaluate the methods, we use F1 score, recall, precision and accuracy. The F value result is 0.20 for LSTM based prediction. For the first method, the arrival port prediction, the F value is 0.45. Considering the single port, we calculate F1 score for the ports like Tuzla is given as an example based on LSTM prediction. For Tuzla, the confusion matrix is presented in Table 5.14. The precision for Tuzla port is 0.4. The recall is calculated 0.4 as well. Therefore, the F score is for clustering based arrival port prediction. The results per port are presented in Table 5.15.

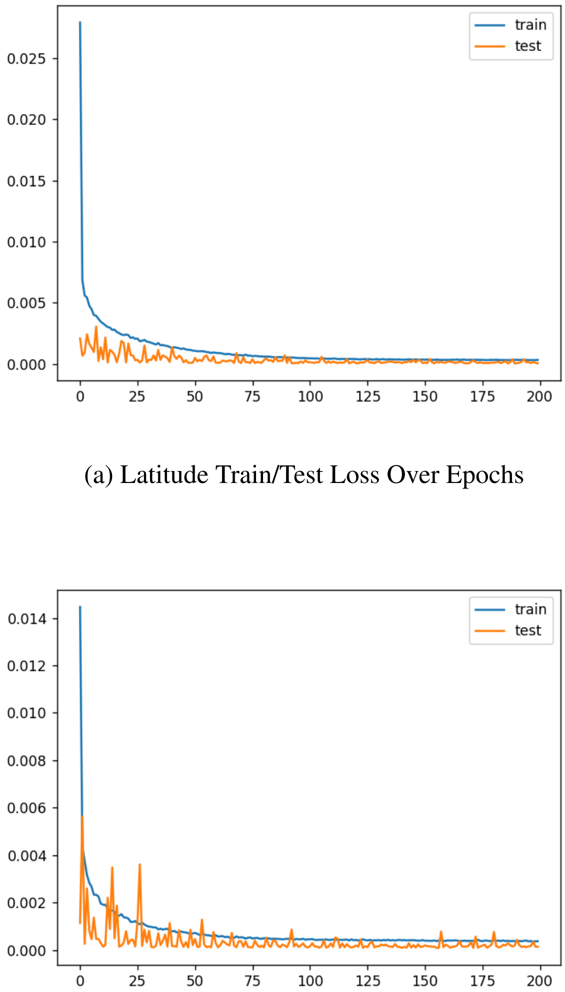
Table 5.14: The Confusion Matrix For Tuzla Port

Tuzla	**Positive**	**Negative**
True	100	600
False	150	150

Table 5.15: The Results For Single Port in Clustering Method

Port Name	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Valetta	**0.88**	**0.82**	**0.77**	**0.88**
Tuzla	**0.83**	**0.75**	**0.69**	**0.83**
Palma De Mallorca	**0.82**	**0.74**	**0.67**	**0.82**
Gibraltar	0.57	0.42	0.33	0.57
Iskenderun	0.11	0.02	0.01	0.11
Marsaxlokk	0.08	0.01	0.006	0.08
All Ports	0.49	0.45	0.44	0.49

Moreover, the prediction of the next single position is performed with the models for latitude and longitude, separately. In Shi, Zhiyuan, and Xu's work[23], it has pointed out that the LSTM was ineffective in the maritime domain. Their model has implemented in this work for the vessel trajectory prediction. The data is split into 50 instances that belong to the same ship. The model learns the next point after 50 points. There are approximately 10000 input sequences. As the metric, we use mean absolute error. The mean absolute error is 0.001 for latitude and 0.002 for longitude. The loss graphic of the model is demonstrated in Figure 5.7b and Figure 5.7a.



(a) Latitude Train/Test Loss Over Epochs

(b) Longitude Train/Test Loss Over Epochs

Figure 5.7: Next Position Train/Test Loss Over Epochs

After finalize the prediction of both latitude and longitude values, the models are tested. The predictions of all next points are placed in Figure 5.9. The LSTM model has been evaluated with 1000 AIS messages that are not used as train or test data. In Table 5.19 predicted locations, Haversine distance between predicted and actual two points are presented for 1000 messages. The train dataset is split by ship id. After separation, for each ship messages are transformed sequences that have 50 messages. The MAE is 16 km for this test data.

Table 5.16: Next Position Results

Coordinates, Actual	**Coordinates, Predicted**	**Distance (in meter)**
[38.22104, 15.63222]	[38.25908 , 15.626723]	**4257**
[36.50253, 27.8519]	[36.5531 , 27.789783]	7901
[37.36572, 15.19324]	[37.384453, 15.096956]	8759
[37.70001, 23.73233]	[37.675148, 23.627747]	9609
[37.91158, 23.60389]	[37.948746, 23.454891]	13705
[36.57535, 28.25002]	[36.62574, 28.10189]	14361
[38.21923, 15.60225]	[38.327244, 15.162796]	40198
[39.09065, 15.32141]	[38.900497, 14.835179]	**47039**

The examples of next position results are given in Figure 5.8. The presented example has 18 km as MAE. This distance can be a good score for the open sea. However, this distance can cause some problems in shores due to vessel density.

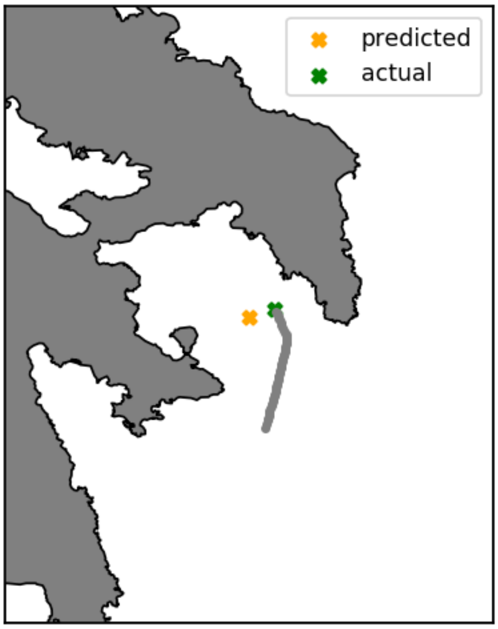


Figure 5.8: The Next Position Results Examples

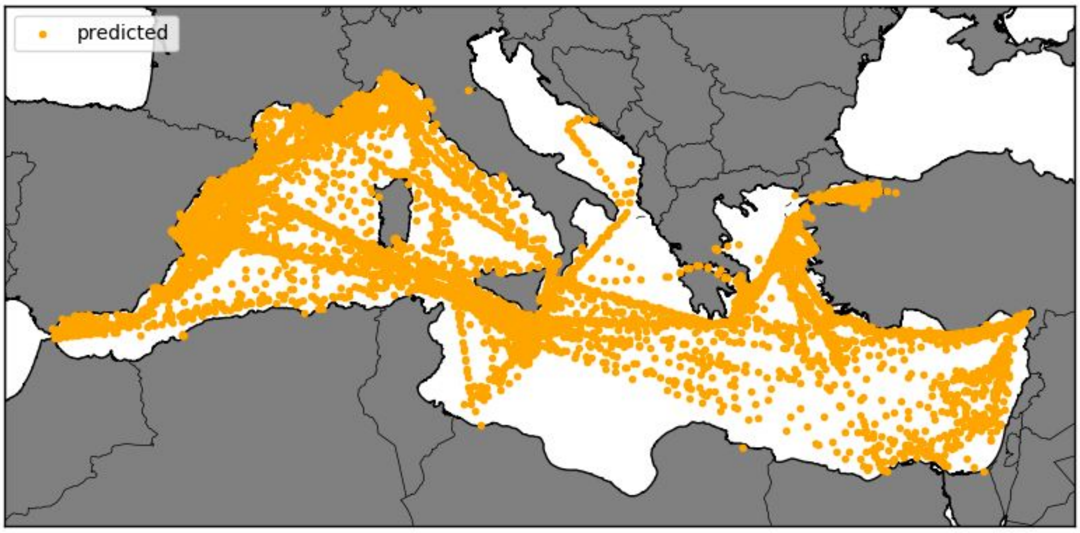


Figure 5.9: The Next Position Predictions

Finally, the next sequence prediction is executed by using 50 instances. The next sequence prediction model has 2 LSTM layer, one dense layer with 5 neurons and 3 activation functions. The model works with latitude, longitude, speed, course and time information and learns prediction of next latitude, longitude, speed, course and time information. Therefore, the model has 5 dimension. The result of this version of LSTM has 0.01 as the mean absolute error . The loss graphic is displayed in 5.10.

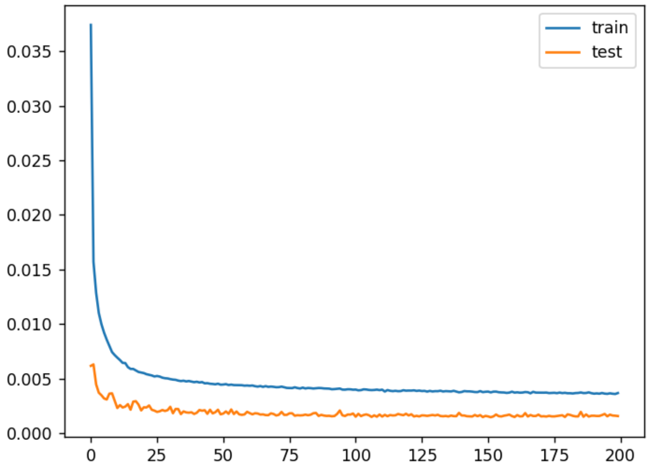


Figure 5.10: The Loss Graphic of The Next Sequences Prediction

In order to predict the next sequences, the sliding window approach has been applied. Initially, the first sequence is the input to the model. The first instance of the second sequence of the ship is given as the 50th instance. The first point of the first sequence has been deleted. Therefore, we can build the sliding window model. In additional experiment, the prediction of the model become a new instance. The process repeats 50 times to obtain next sequence. The example of the next sequence prediction is demonstrated in Figure 5.11.

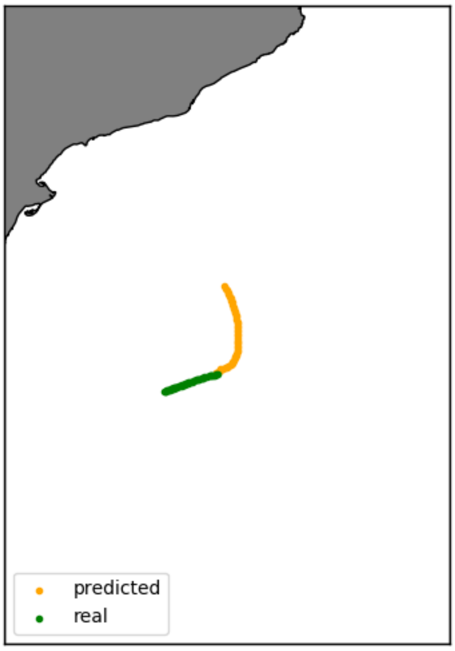


Figure 5.11: The Prediction of A New Sequence

To evaluate the predictions, we have tested the LSTM model with 1000 messages. The AIS messages are divided into 50 sequences by the ship id field. We calculate Haversine distance between predicted next sequence with 50 messages and the actual sequence with 50 messages. Hence, 1000 AIS messages are split into 50 instances, there are 20 sequences. However, to predict the next position, we cannot include the last 50 messages for each ship. Also, for the calculation, we should have 50 messages per vessel. If the messages of the ship are fewer than 50 than we ignore the ship. We calculate mean absolute error between the actual route and predicted. Each route has waypoints. In calculation, we calculate the distance between each waypoint respectively. All distances are calculated by Haversine Distance in the meter. The results of the experiments are shown in Table 5.17 and 5.18. The first table shows the result of sliding window model with actual AIS messages. The second one demonstrates the result of the sliding window model with predicted AIS messages.

In the evaluation phase, data has 10 sequences of unique ships. Each sequence has 50 location points. We have predicted the next 50 positions with different approaches. The results for the sliding window with the predicted position are 79 km for average, 118 km for maximum, and 17 km for the minimum. The results for the sliding window with actual next position are 24 km for average, 46 km for maximum and 9 km for the minimum. Both versions have bad results for stationary points.

Table 5.17: The Results For Sliding Window With Actual Next Position

Avg. Distance(m.)	**Max. Distance(m.)**	**Min Distance(m.)**
6807	**8878**	**5974**
12037	16000	8935
14057	19917	4756
15989	18089	13207
18910	23299	11455
20489	37077	8701
31907	49531	6126
36316	125583	12106
41182	94596	10905
46472	**75336**	**9983**

Table 5.18: The Results For Sliding Window With Predicted Next Position

Avg. Distance(m.)	**Max. Distance(m.)**	**Min Distance(m.)**
13497	**26197**	**6683**
17973	21420	9928
33377	43903	16456
50185	112736	6234
69507	128346	17382
86447	156934	11455
98754	135808	6126
105107	125836	57683
128349	223499	25014
186985	**208353**	**14746**

5.2 Anomaly Detection

In the anomaly detection section, there are two different approaches. The Longest Common Subsequence [44] algorithm is applied to detect differences between two trajectories. The neighborhood search algorithm is also working for the detected the unusual stop routes and generating base for comparison of abnormal sequences.

5.2.1 Neighborhood Search Algorithm

The neighborhood search method is applied to all the data. The first stage of the algorithm is finding the waypoints. In order to determine the waypoints, the speed value of each row is checked whether it is equal or lower than 0.1 knots. the AIS data has the speed information. However, to ensure the value, it is calculated with division distance and time between two messages. Using the speed value, both calculated and extracted, waypoints are created. There are approximately 35760 waypoints extracted from 542000 AIS messages. The waypoints are presented in Figure 5.12.

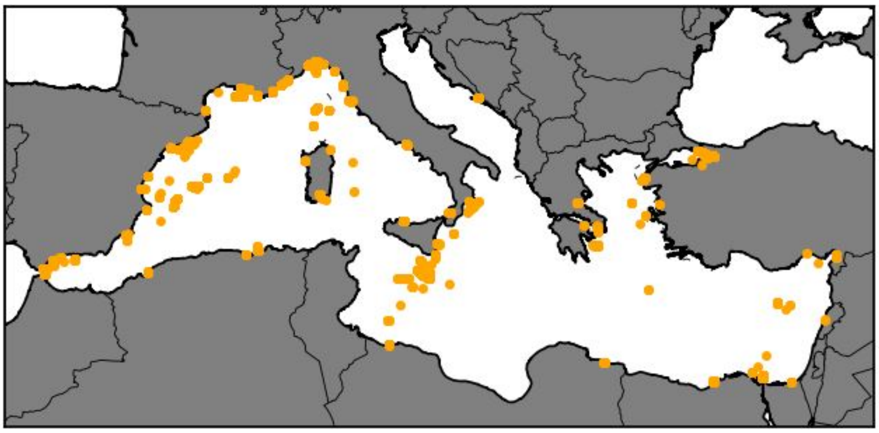


Figure 5.12: The All Waypoints On The Map.

Later, all the waypoints are clustered to obtain ports and stationary points. DBSCAN is executed to cluster the waypoints. After the clustering process, there are 90 clusters. The centroid of each cluster is discovered with cosine distance. The centroids can be interpreted as ports and stationary points in the open sea. In Figure 5.13, all derived centroids are displayed.

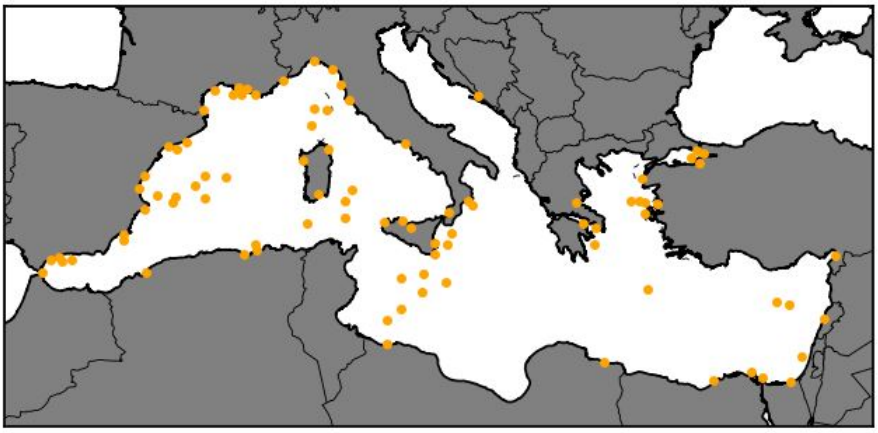


Figure 5.13: The Visualization of Centroids On The Map.

After reaching all centroids, the trajectories are calculated between two of them. Starting with the initial waypoint, legs of route are defined until reaching the final waypoint. By using the AIS messages in a radius of the centroids, each point in a trajectory is identified. In this study, the radius is 10 km. Also, the identified point is moving to next point in a time interval. The time interval is one hour in this work.

First, the neighborhood algorithm performs without improvement. As such unsatisfied density, unusual stop and reaching the arrival point, there are three types of routes. The algorithm finds totally 43 routes. Also, there are 68 unusual stop trajectories. The results are demonstrated in Figure 5.14. The found routes are demonstrated in Figure 5.15.

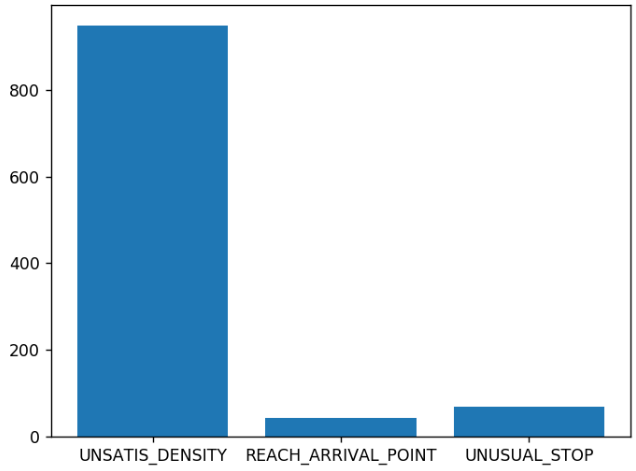


Figure 5.14: The Route Distribution With Base Neighborhood Search Method

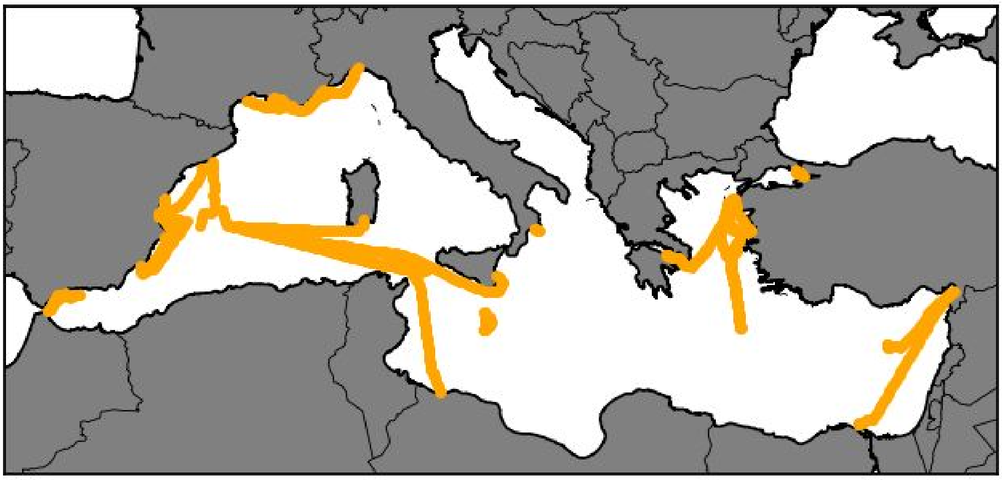


Figure 5.15: The Extracted Routes With Base Neighborhood Search Method

Furthermore, there are two improvements applied to the neighborhood search algorithm. The first is finding bearing based trajectories. In this approach, the bearing between departure and arrival centroid is calculated. In each point of a route, the course information is controlled whether it is in the right way according to bearing value. There are 285 complete trajectories that the algorithm found. The route distribution is given in Figure 5.16. Also, the obtained trajectories are demonstrated in Figure 5.17.

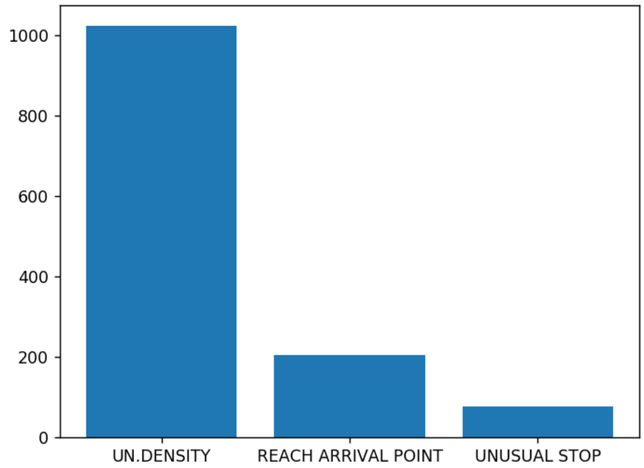


Figure 5.16: The Trajectory Distribution In Bearing Based Neighborhood Search.

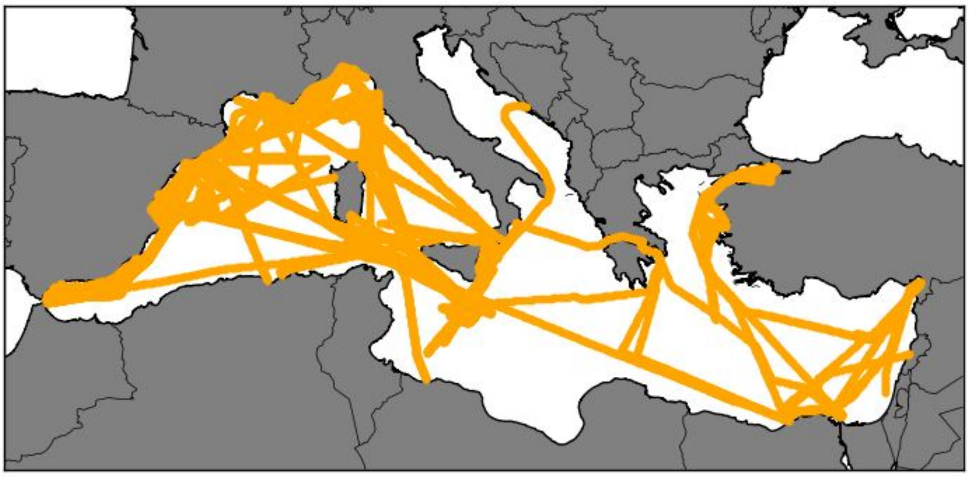


Figure 5.17: The Projection of Bearing Based Trajectories On The Map.

Lastly, the final improvement contains two parts. The first step is to clustering the course values in an initial waypoint. Initially, the course value is more than one. Therefore, there is more than one route between two waypoints. The course values are the centroid of each cluster. Thereby, the course is calculated more accurately. The distribution of routes is given in Figure 5.18. The detected routes are shown in Figure 5.19.

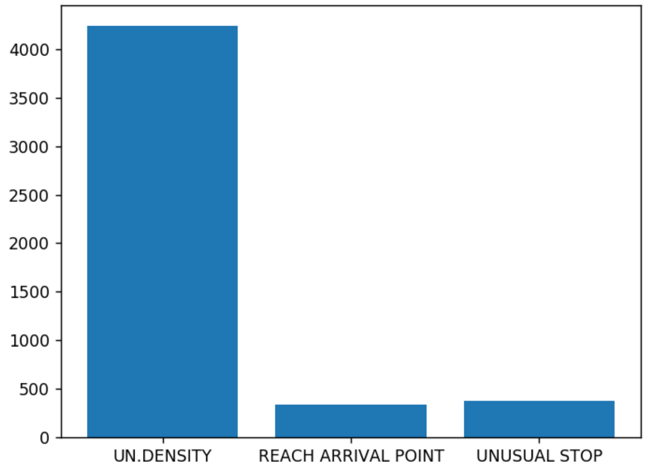


Figure 5.18: The Results of Time Based Routes.

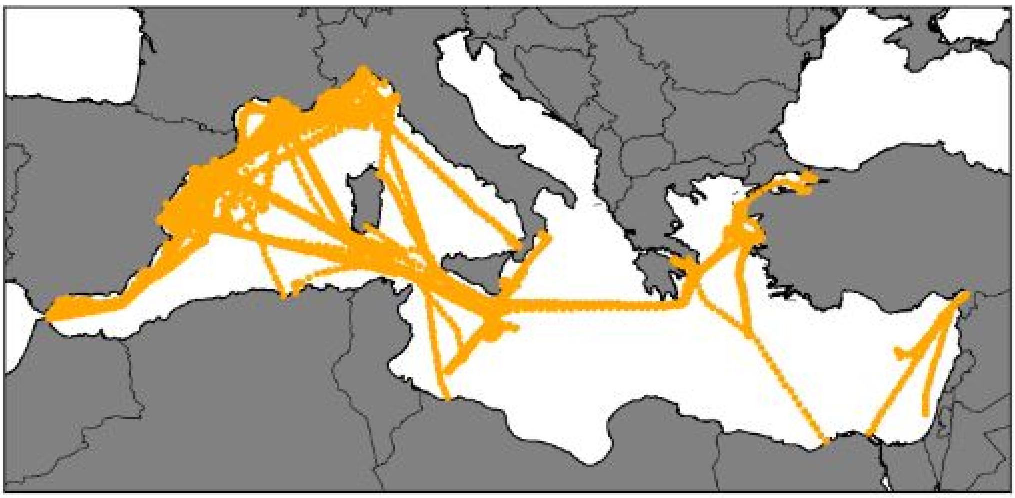
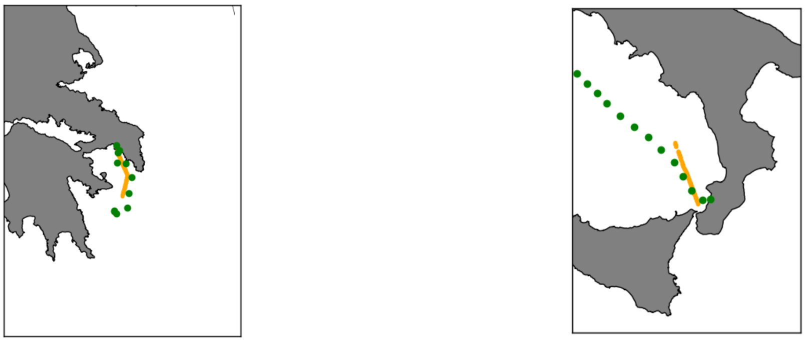


Figure 5.19: Projection of Time Based Trajectories On The Map.

5.2.2 Finding Dissimilar Routes with Extracted Trajectories

LCS [44] method is applied in two sequences. The extracted routes that are saved in the database, are used in measuring the similarity with a new sequence. The similarity of 1000 AIS messages are calculated. There are 51 unique ship in the test data. The 1000 messages are separated according to their ship id. The most similar sequence result is 55%. It is presented in Figure 5.20a. Another sequence similarity result with 25% ratio is displayed in Figure 5.20b.



(a) 55% Similarity Ratio.

(b) 25% Similarity Ratio.

Figure 5.20: The examples of Routes Comparison

Longest Common Subsequence method has been tested with different radius. In Table 5.19 the result of different radius experiments have been given.

Table 5.19: LCS Results

LCS Similarity Radius(in meter)	**100k**	**10k**	**1k**
Match Count For All Sequence	47	23	7
Avg. Sim. For All Sequence	0.96	0.55	0.02
Calculated Maximum Sim.	1.0	1.0	0.07
Match Ratio For Dataset	0.92	0.45	0.13

5.2.3 Unusual Stop Routes

The neighborhood search algorithm can handle with finding unusual stop trajectories. The algorithm works with all data to identify stop routes. There is three types of approach in neighborhood search algorithm. The first one is unimproved version. There are eight unusual stop path. In the second approach, the number has increased up to 100. In the final version, the number of unusual route is 250. In Figure 5.21 the count of found unusual stop route is displayed according to used method. The count of unusual stop routes is related to the given time interval. Using the shorter time interval gets the little unusual routes.

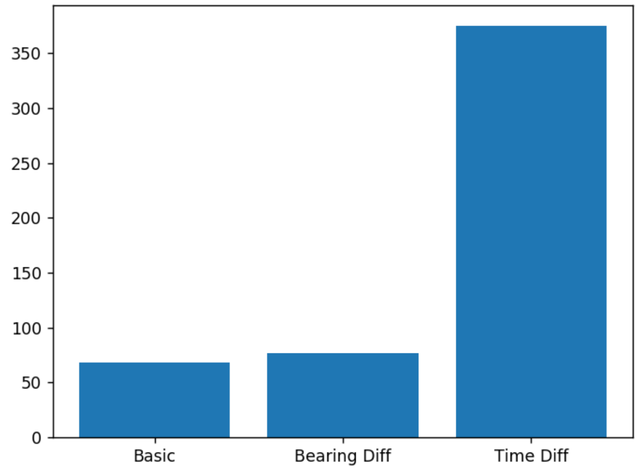


Figure 5.21: Unusual Stop Routes Count.

5.3 Discussion

In this section, we compare the results of each method. Although there are a few experiments on AIS messages, they have not all worked out well. It is difficult to find existing studies that are used the same AIS dataset. Therefore the evaluations of some methods are made with observations.

As a beginning, the basic clustering and classification methods are attempted. The structure of input data has been changed over the methods. This has shown that using AIS messages into sequences is more efficient. The random AIS messages are inefficient as input for any algorithm. Furthermore, the best classification algorithm is Random Forest. The best clustering algorithm is DBSCAN. Therefore, it is used as the first arrival port prediction method.

In the arrival port prediction problem, there are two different approaches. The first one is clustering and classification, and the other one is the LSTM classification. The clustering for all data has 0.13 silhouette score. The reason why that the score is that low is a sparse dataset. The dataset includes whole Mediterranean AIS messages. With DBSCAN clustering, we have found the hot areas in the Mediterranean. However, the distance is highly far for prediction. Therefore, we have been working the first 10000 messages that have 0.8 accuracy ratio. Moreover, in the clustering, the Haversine distance metric has been applied. Different distances have been tried for prediction. The best result is obtained when 50 km has been used. The classification has applied into clusters set. This gives us the route branching in the hotspots. The accuracy result of this method for 1000 data is 0.49. The reason for low accuracy is the built model based on the first 10000 rows. However, the prediction has been made for the next 1000. In the next 1000, the arrival ports can be different from the model knows.

In the LSTM based approach, the data has been split into 50 messages sequences according to ship number. The result of LSTM shows that the model has overfitted. The result of the train has better results than the test set. As a solution, the size of data can be increased in the future. The accuracy of the model is 0.45 for all data. However, when we have decreased the number of input into 10000, the accuracy is decreased to 0.19. The situation of overfitting also continues with fewer data.

The comparison of these two methods has been made with 10000 train and 1000 test data. According to the results, the DBSCAN algorithm has better scores.

The other problem is defined as the next position prediction with LSTM. There are also two different approaches. The first one is single-point prediction and the second is multiple point prediction, In single-point prediction, we have latitude, longitude, speed, course and elapsed time as the training input. The prediction is on the next latitude and longitude. The MAE of the model is 0.002 and 0.001. That means 0.2

km error in single prediction, approximately. For evaluation, 1000 AIS messages are used. The distance between the actual next position and predicted one has been measured. As a result of this, the average of the distances is 18 km approximately. This distance can be a good score for the open sea. However, this distance is vital in shores. The next position problem is interpreted as an anomaly problem. The prediction has been made for an observed sequence. Then, if the vessel moves to a different location, an alert may be given.

The other approach is multiple point prediction. In this version, we use the same inputs for the prediction. However, we want different outputs. The next position, speed, course and elapsed time have been predicted. The reason is the sliding window usage. While forecasting the next sequence, a single point has been predicted. Then, it is used as an input for the next prediction. The MAE score is 0.01 for the model. That means 1 km error occurred for each prediction, approximately. However, the sliding windows method has not successful results. The next single position prediction with this 5d model has great result according to MAE score. But, the more we use the predicted data as input, the more the sequence becomes divergent. With the next sequence LSTM, we predict not only position but also speed course and elapsed time. Due to the use of sliding windows in the evaluation stage, the predicted next sequence becomes far away from the actual route. 5D LSTM is worse than the next position prediction model with test data. In the sliding window with predicted ones, the average distance between predicted and actual for 500 instances is 79 km. The maximum distance average is 118 km and the minimum distance average is 17 km between two sequences. The other version of the sliding window has 24 km average distance. The maximum distance is 46 km and the minimum is 9 km for test data. It is observed that with the stationary points, the prediction of the LSTM model has got bad results. The reason is that the 50 instance sequence of stationary points has inconsistent time, latitude and longitude values. For the complete trajectory prediction, the model has better results in both approaches.

The final problem is anomaly detection from existing routes. To solve this problem, the trajectories should be extracted initially. The previous study, TREAD [10] has been implemented. Then, the results of this version have found 43 routes between two waypoints. In this work, we have developed a new version of the algorithm.

We make some improvements. The first improvement is bearing calculation. With bearing calculation, we have 285 trajectories. Not only calculating bearing but also interpolation has been made as an improvement. We have interpolated the missing AIS messages 10 times. After 10 trying we have stopped. Another improvement is moving vessels in the timeline. We have checked every ship whether it has a right timestamp in the determined time interval. Furthermore, we apply the kNN clustering in the course value of AIS messages that are in a 10 km range in the initial waypoint. With version, there are 334 routes.

After the extraction phase, the anomaly detection has been applied. To detect the abnormal movements of ships, we calculate the similarity in extracted trajectories. LCS is applied to determine the similarity. The Haversine distance has been used to match in a determined radius. We have 1000 messages to test these approaches. Different radius has tested. The wider it is, the more match we have got. Although the best results obtained from the 100 km range, 10 km is more proper than it. In the 10 km range, the average similarity is 0.55. The test data contains AIS messages from both moving and stopping ships. This leads to low similarity because of stopping points. However, 0.55 is a good result for never used test data. This means that the extraction of existing routes should be continuous to elicit high similarity scores.

As another anomaly, the unusual stop trajectories are investigated. While the extraction of trajectories, the unusual stop trajectories have been found. Three different extraction methods have a different number of unusual stop routes. The basic version of the algorithm finds 68 routes. The bearing calculated version has 77 trajectories. The final version has 375 unusual stop routes. As a next step, we can compare the obtained sequences with these routes to give an alert to the operator.

5.4 Implementation Environment and Complexity

The development environment is as follows; The system has 32 GB memory, i7 Intel processor running on Windows 7 operating system. We use Python 3.6 as the programming language. Jupyter is used for development in Anaconda environment. Keras is used for LSTM analysis. we use Scikit-Learn machine learning library to analyze AIS messages.

The complexity of the Neighborhood Search Algorithm is $O(n^2)$ like most search algorithms. We want to use Apache Spark [49] for data parallelism as future work.

The complexity of LCS is $O(2^{nxm})$. To make the algorithm more feasible, we implement LCS with the dynamic programming approach in this study. Therefore, the DP-LCS complexity is $O(nxm)$.

CHAPTER 6

CONCLUSION

In the maritime domain, the need for an auxiliary tool is inevitable. The streaming data is excessively more than operators can handle in marine. Having a tool that captures and reports anomalies missed by operators will assist operators. Therefore, the study has arranged for this purpose.

The study has two different sections. In the first phase, prediction of next position and arrival port has been performed. The common clustering method detects message clusters. The classification specifies branches in clusters. In the result of this method with the first 10000 messages, there are ten clusters. The next point and next sequence predictions should be improved for the shores. However, the results of them are acceptable for the open sea.

In the second approach, the neighborhood search works with all data. The vessel trajectories are extracted in three different approaches. The result of the first form is 43 trajectories. Next, with the improved version of neighborhood search algorithm, there are 285 trajectories that are extracted. Later, the final improved version of the algorithm has detected 334 trajectories.

In the second phase, the recognition of the similarity between a sequence and the obtained trajectories appears with LCS algorithm. The most similar sequence has 55% ratio between 54 sequences in acceptable radius distance. Additionally, the unusual stop paths are presented. There are 68, 77 and 375 routes detected with different neighborhood search algorithm approaches.

As the future work, the route detection algorithms can be enhanced with streaming AIS messages. While the messages arrive, the algorithm improves. Furthermore, other types of anomalies can be inspected. For instance, meeting two or more vessels at the same point in the near future, not arriving destination port in arrival time, incompatibilities of speed and position information, incompatibilities of navigation status information and kinematic information, position and speed, very low speed value, unexplained high speed value anomalies can be studied in the future. Also, we can work with Bayesian Networks [50] for anomaly detection.

REFERENCES

[1] *DEBS 2018*. www.cs.otago.ac.nz/debs2018/calls/gc.html.

[2] *European Maritime Safety Agency*. www.emsa.europa.eu.

[3] "The annual overview of marine casualties and incidents," tech. rep., European Maritime Safety Agency, 2018. 'http://www.emsa.europa.eu/news-a-press-centre/external-news/item/3406-annual-overview-of-marine-casualties-and-incidents-2018.html.

[4] R. K. Tendler, "Cellular phone based automatic emergency vessel/vehicle location system," Sept. 10 1996. US Patent 5,555,286.

[5] *International Maritime Organization*. www.imo.org.

[6] S. Mankabady, *THE INTERNATIONAL MARITIME ORGANIZATION, VOLUME 1: INTERNATIONAL SHIPPING RULES*. 1986.

[7] R. Kjellberg, "Capacity and throughput using a self organized time division multiple access vhf data link in surveillance applications," *Department of Computer and System Sciences*, vol. 53, 1998.

[8] *"AIS Transponders"*. www.imo.org/en/OurWork/safety/navigation/pages/ais.aspx.

[9] *Marine Traffic*. www.marinetraffic.com.

[10] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, pp. 2218–2245, 06 2013.

[11] I. M. de Vreede, "Managing historic automatic identification system data by using a proper database management system structure," 2016.

[12] G. M. Morton, "A computer oriented geodetic data base and a new technique in file sequencing," 1966.

[13] O. Bodunov, F. Schmidt, A. Martin, A. Brito, and C. Fetzer, “Grand challenge: Real-time destination and eta prediction for maritime traffic,” *arXiv preprint arXiv:1810.05567*, 2018.

[14] J. H. Friedman, “Stochastic gradient boosting,” *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[15] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, ACM, 2016.

[16] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[17] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[19] C. Tang and Z. Shao, “Data mining platform based on ais data,” 2009.

[20] S. Mao, E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G.-B. Huang, “An automatic identification system (ais) database for maritime trajectory prediction and data mining,” *ArXiv*, vol. abs/1607.03306, 2018.

[21] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, *et al.*, “Extreme learning machine: a new learning scheme of feedforward neural networks,” *Neural networks*, vol. 2, pp. 985–990, 2004.

[22] *arcGIS*. www.arcgis.com.

[23] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, “Lstm-based flight trajectory prediction,” *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2018.

[24] X. Tang, J. Gu, Z. Shen, and P. Chen, “A flight profile clustering method combining twed with k-means algorithm for 4d trajectory prediction,” in *2015*

Integrated Communication, Navigation and Surveillance Conference (ICNS), pp. S3–1, IEEE, 2015.

[25] J. Roskam, *Airplane flight dynamics and automatic flight controls*. DARcorporation, 1998.

[26] B. L. Decker, “World geodetic system 1984,” tech. rep., Defense Mapping Agency Aerospace Center St Louis Afs Mo, 1986.

[27] T. Soler and L. D. Hothem, “Coordinate systems used in geodesy: Basic definitions and concepts,” *Journal of surveying engineering*, vol. 114, no. 2, pp. 84–97, 1988.

[28] R. Bleck and L. T. Smith, “A wind-driven isopycnic coordinate model of the north and equatorial atlantic ocean: 1. model development and supporting experiments,” *Journal of Geophysical Research: Oceans*, vol. 95, no. C3, pp. 3273–3285, 1990.

[29] L. R. Rabiner and B.-H. Juang, “An introduction to hidden markov models,” *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[30] P. Lockwood and J. Boudy, “Experiments with a nonlinear spectral subtractor (nss), hidden markov models and the projection, for robust speech recognition in cars,” *Speech communication*, vol. 11, no. 2-3, pp. 215–228, 1992.

[31] *NATO Science and Technology Organization, Centre for Maritime Research and Experimentation*. https://www.cmre.nato.int/.

[32] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *KDD*, 1996.

[33] J. Roy, “Anomaly detection in the maritime domain,” in *Optics and Photonics in Global Homeland Security IV*, vol. 6945, p. 69450W, International Society for Optics and Photonics, 2008.

[34] R. Khan, M. Ahmad, and M. Zakarya, “Longest common subsequence based algorithm for measuring similarity between time series: a new approach,” *World Applied Sciences Journal*, vol. 24, no. 9, pp. 1192–1198, 2013.

[35] D. S. Hirschberg, “A linear space algorithm for computing maximal common subsequences,” *Communications of the ACM*, vol. 18, no. 6, pp. 341–343, 1975.

[36] D. S. Hirschberg, “Algorithms for the longest common subsequence problem,” *Journal of the ACM (JACM)*, vol. 24, no. 4, pp. 664–675, 1977.

[37] D. O. D. Handayani, W. Sediono, and A. Shah, “Anomaly detection in vessel tracking using support vector machines (svms),” *2013 International Conference on Advanced Computer Science Applications and Technologies*, pp. 213–217, 2013.

[38] *MongoDB*. www.mongodb.com.

[39] *NoSQL DEFINITION: Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontally scalable*. http://nosql-database.org/.

[40] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” 1967.

[41] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001.

[42] P. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” 1987.

[43] T. Vincenty, “Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations,” *Survey Review*, vol. 23, pp. 88–93, 04 1975.

[44] I. D. Melamed, “Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons,” *arXiv preprint cmp-lg/9505044*, 1995.

[45] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and C. Schmidt, “The geojson format specification,” *Rapport technique*, vol. 67, 2008.

[46] S. Gillies, H. Butler, M. Daly, A. Doyle, and T. Schaub, “The geojson format,” *coordinates*, vol. 102, pp. 0–5, 2016.

[47] F. Chollet *et al.*, “Keras (2015),” 2017.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[49] *Apache Spark*. https://spark.apache.org/.

[50] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.