AIS DATA-DRIVEN GENERAL VESSEL DESTINATION PREDICTION: A TRAJECTORY SIMILARITY-BASED APPROACH

by

Chengkai Zhang

B.Eng., Hohai University, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE COLLEGE OF GRADUATE STUDIES
(Electrical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Okanagan)

September 2019

The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, the thesis entitled:

AIS DATA-DRIVEN GENERAL VESSEL DESTINATION PREDICTION: A TRAJECTORY SIMILARITY-BASED APPROACH

submitted by Chengkai Zhang in partial fulfillment of the requirements of the degree of Master of Applied Science .

Dr. Zheng Liu, School of Engineering
Supervisor

Dr. Chen Feng, School of Engineering
Supervisory Committee Member

Dr. Sina Kheirkhah, School of Engineering
Supervisory Committee Member

Dr. Liwei Wang, School of Engineering
University Examiner

Abstract

Shipping is one of the major transportation approaches around the world. With the growing demand for global shipping service, the vessel destination prediction has shown its significant role in improving the efficiency of decision making in industry and ensuring a safe and efficient maritime traffic environment. Currently, most vessel destination prediction methods focus on regional destination prediction, which has restrictions on destinations and regions. Thus, this thesis proposes a general AIS (Automatic Identification System) data-driven vessel destination prediction method. The proposed method first extracts the vessel's traveling trajectory and departure port from AIS records. The similarities between traveling and historical trajectories are then measured and utilized to predict the destination. The destination of the historical trajectory, which shares the highest similarity with the traveling trajectory, is predicted as the vessel's destination. Compared with related work that using maritime records as input and destination as output, the proposed method is more general, accurate, and updatable. In this thesis, a historical trajectory database was generated from more than 141 million AIS records, which covers 534,824 traveling patterns between ports and more than 5.9 million historical trajectories. Comparative studies were carried out to validate the performance of the proposed method, where eight state-of-the-art similarity measurement methods combined with two different decision strategies were implemented and compared. The experimental results demonstrate that the proposed random forest-based model combined with the port frequency-based decision strategy achieves the best prediction accuracy on 35,937 testing trajectories.

Lay Summary

With continuously increasing demands for global shipping service, vessel destination prediction has come to the fore in the worldwide seaborne trade. The accurate information of when and where the vessel will dock not only give the global commodity trading industry a chance to make timely and efficient decisions, but also enable the port to arrange the dock for vessels more efficiently. However, the existing AIS data-driven destination prediction methods could not be simply applied to global vessels. This thesis proposed a method of predicting the global vessels' destination by conducting the similarity comparison between the vessel traveling trajectory and historical trajectories. The method can handle varied situations, if these situations are available in the database of historical trajectories. Experimental results demonstrate the feasibility and validity of the proposed method.

Preface

This thesis is based on the research work conducted in the School of Engineering at the University of British Columbia, Okanagan Campus, under the supervision of Prof. Zheng Liu. The main content in this thesis is based on our submitted journal paper.

Chapter 3, Chapter 4 and Chapter 5 have been submitted to the Transportation Research Part C: Emerging Technologies.

I am the principal contributor to these works. Prof. Zheng Liu provided me with some advice on research methodology and experiment design and preparation of the manuscripts. Mr. Junchi Bin and Mr. Xiang Peng helped preparation of the manuscripts. Mr. Richard Halldearn, the CTO at the Navarik Corp., and Mr. Rui Wang, the software developer at the Navarik Corp., provided the data used in this thesis.

Table of Contents

List of Tables

List of Figures

Glossary

AIS Automatic Identification System

AMSS Angular Metric for Shape Similarity

ANN Artificial Neural Networks

APDE Average Prediction Distance Error

CityACC City Accuracy

CNN Convolutional Neural Network

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DL Deep Learning

DTW Dynamic Time Warping

EDR Edit Distance on Real sequences

EDTP Edit Distance on Trajectories Pattern

ERP Edit Distance with Real Penalty

ETA Estimated Time of Arrival

GT Gross Tonnage

IndRNN Independently Recurrent Neural Network

LCSS Longest Common Subsequence

ML Machine Learning

MLP Multilayer Perceptron

MSD Maximum Similarity-based Decision strategy

PFD Port Frequency-based Decision strategy

PortACC Port Accuracy

RF Random Forest

RNN Recurrent Neural Network

SAR Search And Rescue

SOLAS International Convention for the Safety of Life at Sea

SPD Shortest Perpendicular Distance

SSPD Symmetrized Segment-Path Distance

Acknowledgments

First of all, I offer my highest gratitude to my supervisor, Prof. Zheng Liu, whose guidance, generosity, kindness, and encouragement throughout my graduate study have been precious and indispensable. He not only gave me the opportunities that have enabled me to develop my professional skills but also guided me on how to think logically.

Thanks also to committee members Prof. Chen Feng and Prof. Sina Kheirkhah for their willingness to serve on the supervisory committee, and Prof. Liwei Wang for his willingness to serve on the university examiner.

In addition, I deeply thank all of my colleagues at the Intelligent Sensing, Diagnostic and Prognostic Research Lab (ISDPRL), who have given me precious suggestions and technical supports over the past two years. Besides, I want to show my special appreciation to Mitacs, NSERC and the company Canscan Inc., Navarik Corp., and Rosen Group. for giving me the opportunities to work on industrial projects.

Finally, I present the highest appreciation to my girlfriend, Ms. Ziyu Li, and my family, for their love, support, and understanding of my graduate studies.

Chapter 1

Introduction

1.1 Background and Motivation

The demand for shipping service has increased in the past several years [2, 3]. In 2016, the total volume of the worldwide seaborne trade reached 10.3 billion tons. The global maritime transportation occupies around 90% of global trading by volume and 70% by value [4]. It's predicted that the total volume of the worldwide seaborne trade will grow at the rate of 3.2% between 2019 and 2022 [5]. With continuously increasing demands for global shipping service, the vessel destination prediction has shown its value in the worldwide seaborne trade. With the accurate information of when and where vessels will dock, the global commodity trading industry can make a timely and efficient decision in business. In addition, more and more ships are built and come into service to meet the growing demands in worldwide seaborne trade. The lack of accurate information regarding vessels' destination and arrival time would subject ports to challenges like arranging wharves for vessels to berth and guiding the traffic routes to ensure the safe and stable maritime traffic environment, etc. Moreover, the vessels' Estimated Time of Arrival (**ETA**) is highly dependent on the destination prediction [6]. Hence, the research on predicting global vessels' destination would be of great value for industry to make timely and efficient decisions and ensure a safe and efficient maritime traffic environment.

Accessing vessels' traveling records and identifying its patterns are essential for predicting vessels' destinations. Automatic Identification System (**AIS**) is a

self-reporting surveillance system installed on board to record the vessels' traveling and return the records for further analysis [7–9]. These records contain information such as timestamps, identification, position, course, and AIS message, etc. The **AIS** message includes the vessel's destination port and **ETA**. However, the manually filled **AIS** messages are not always available or with mistakes as reported in references [10]. For example, literature [11] claimed the accuracy of the destination port and **ETA** filled in **AIS** message was about 4%. Currently, the International Convention for the Safety of Life at Sea (**SOLAS**) requires that the international voyaging ships with 300 or more Gross Tonnage (**GT**), and all passenger ships must carry the **AIS**. Thus, a high volume of vessel trajectory records become available for analysis. With the vessels' maritime records, it is possible to predict the vessel destination by applying the advanced machine learning techniques.

Unlike the vehicles with limited route choices [12], a vessel can move from one port to any other at varied speeds and via different routes [13], which makes it a challenge to accurately predict the vessel's destination. Kepaptsoglou et al. [14] stated that weather conditions would significantly affect the operation of the vessel. According to Lokukaluge P. Perera [15], vessels may change the heading, speed, and routes according to the weather conditions. The external environment, such as wind, wave, and current strongly affects vessels' movements and brings great uncertainties to the vessels' motion [16]. If the vessels are operated in ship-to-ship transfers, these vessels will not head to destination ports directly. The prediction of vessels' destinations from its trajectories will be difficult. Therefore, the uncertainties brought by environmental and human factors become the obstacle for predicting vessels' travel destinations.

Research has been carried out to address the uncertainty issue in a regional area, such as the coast of Mexico region and Florida region [17], North Adriatic Sea area [18] and other areas [19–21]. These work achieved promising prediction results on vessels' destinations with limited options in a specific area. However, there are more than thousands of ports around the world [13], and there are more than millions of possible routes between any two ports. The existing regional vessel destination prediction methods presented in [17–21] could not be simply applied to the vessel destination prediction generally. Herein, regional vessel destination prediction methods represent the methods with the restriction on the specific areas

(would be regions of the world). This study addresses the general vessel destination prediction, which considers the whole world and has no requirements of vessel types, sailing duration, limited potential destinations, etc.

Based on the motivations mentioned above, to predict both short- and long-term global vessel destination, this thesis proposed a method of predicting the vessels' destination by conducting the similarity comparison between the traveling trajectory and the historical trajectories. A historical trajectories database, which stores trajectories that vessels have traveled between every two global ports, was first built by the proposed Density-Based Spatial Clustering of Applications with Noise (**DBSCAN**)-based trajectory segmentation algorithm. The proposed Machine Learning (**ML**)-based approach was then employed to measure the similarities between traveling and historical trajectories. The destination were then predicted based on the measured similarities and potential destinations' frequencies.

1.2 Thesis Outline and Contributions

This thesis is organized into five chapters.

Chapter 1 presents the background of the vessel destination prediction, the motivations, the current challenges, and the brief introduction to the solution proposed in this thesis.

Chapter 2 reviews existing **AIS** data-driven predictive applications (including pattern extraction, event recognition, vessel **ETA** prediction and destination prediction). Related trajectory data mining methods, which consist of trajectory preprocessing and state-of-the-art similarity measurements, are also reviewed and investigated in this chapter.

In Chapter 3, the flow and the mathematical processes of the proposed solution are explained. First of all, the overview of the proposed general vessel destination prediction solution is provided, and the components of the proposed method are clarified. Then, these components are introduced respectively in different sections. Finally, the evaluation metrics used in this thesis are explained.

In Chapter 4, two experiments are set for evaluating the validity and feasibility of the proposed trajectory similarity-based method. The data for experiments are described in detail at the beginning of this chapter. The four pro-

posed methods are then compared with eight state-of-the-art methods. The experimental results reflect the proposed Random Forest (**RF**)-based model combined with Port Frequency-based Decision strategy (**PFD**) performs best among all compared methods. The **RF**-based model combined with **PFD** is then further investigated and discussed. Moreover, the proposed solution is the first application of using trajectory similarity in vessel destination prediction.

Chapter 5 concludes this thesis and provides recommendations of future researches.

The main contributions of this thesis can be summarized as follows:

- This study proposed a novel method for **AIS** data-driven general vessel destination prediction without restrictions on regions, time range, candidate port destinations, etc. The method, for the first time, employs the similarities between traveling and historical trajectories for vessel destination prediction.

- This study proposed a **DBSCAN**-based trajectory segmentation method, which can extract and segment trajectories from **AIS** records with the input of global port locations. In order to predict the world-wide vessel destination, a large dataset containing comprehensive historical trajectories was constructed by the proposed **DBSCAN**-based trajectory segmentation algorithm from 141,892,144 **AIS** records. The dataset contained 5,928,471 historical trajectories between 10,618 ports during the period year 2011 to 2017.

- In this study, four **ML**-based similarity measurement methods were proposed, and compared with the state-of-the-art methods on the performance of vessel destination prediction. Eight state-of-the-art trajectories similarity measurements were studied in this thesis. The experimental results demonstrated that the proposed **RF**-based similarity measurement method significantly outperformed state-of-the-art methods and other three proposed **ML**-based similarity measurement methods.

- This study proposed two decision strategies to predict the vessels destination based on similarities between trajectories.

Chapter 2

Literature Review

This thesis proposes a novel solution for general vessel destination prediction, along with new methodologies in vessel trajectories data mining. In this chapter, state-of-the-art research regarding **AIS** data-driven prediction is reviewed from both application and methodology perspectives. Section 2.1 gives an overview of existing **AIS** data-driven predictive applications. Reviews on two core aspects of **AIS** data-driven predictive applications are presented in Section 2.1.1 and Section 2.1.2 respectively. Subsequently, Section 2.2 provides a review of state-of-the-art trajectory data mining methodologies. Trajectory preprocessing and trajectory similarity measurement methods are reviewed in Section 2.2.1 and Section 2.2.2 respectively.

2.1 AIS Data-Driven Predictive Applications

AIS data-driven predictive applications are widely used in the shipping industry, and can be majorly categorized into two classes: 1. Pattern Extraction and Event Recognition; 2.Vessel ETA and Destination Prediction.

2.1.1 AIS Data-Driven Pattern Extraction and Event Recognition

AIS data-driven pattern extraction and event recognition aim to transform **AIS** records into understandable information, e.g., patterns of shipping routes, vessel class identification, etc. Giannis et al. [22] proposed a method to extract the global

trade patterns from billions of **AIS** records. Chatzikokolakis et al. [23] researched on a novel method of automatically detecting Search And Rescue (**SAR**) activity through using Random Forest. Kostas et al. [24] and Manolis et al. [25] presented the systems for online monitoring of maritime activity from **AIS** records with a component for trajectory simplification. Ljunggren [26] presented a classification-based method to classify vessel type by analyzing the vessel motions. Liao et al. [27] proposed a hierarchical Markov model that could learn and infer the users daily movement and its use of different modes of transportation.

2.1.2 AIS Data-Driven Vessel ETA and Destination Prediction

Generally, the **AIS** data-driven prediction is classified into two categories: the **AIS** data-driven destination prediction and **ETA** prediction using **AIS** data. Dobrkovic et al. [28] conducted a review of existing algorithms on maritime route prediction using **AIS** data-driven prediction. Related work on forecasting ship positions to support steering decision and avoid vessels collision are published [29–31]. Most of these methods estimated the collision between vessels, rather than vessel destination prediction. Alessandrini et al. [6] developed a data-driven methodology to estimate **ETA** of the vessel in port areas using the information of the vessels destination. Therefore, **AIS** data-driven destination prediction is mainly based on **AIS** data-driven prediction. Regarding **AIS** data-driven destination prediction methods, they can be classified into two categories: turning-point based destination prediction methods and trajectory-based destination prediction methods.

Turning-point based destination prediction methods. The turning-points are first extracted and then used for processing the **AIS** records. Following that, the processed historical **AIS** records, along with destinations, are fed into Artificial Neural Networks (**ANN**) for training the destination predictions model. Wilson [17] and Pallotta et al. [18] proposed the Bayes-based methods which achieved excellent performance in vessel destination prediction in some regions like the coast of Mexico region, Florida region and North Adriatic Sea area. Besides, Daranda [20] developed a regional vessel destination prediction method by using the Multilayer Perceptron (**MLP**), and this method showed good performance for destination predictions among 24 ports.

Trajectory based destination prediction methods. Kim and Lee [19] and Lin et al. [21] implemented trajectory-based regional vessel destination prediction methods. In these works, maritime data, including historical **AIS** records and destinations, were fed into the **ML** model to train the destination prediction model. In the research [19], an **AIS** data-driven destination prediction method through **MLP** has been proposed. Lin et al. [21] proposed that Recurrent Neural Network (**RNN**) is a promising solution for predicting the regional vessels' destinations. These two methods achieved accurate performances on regional vessel destination prediction with limited destination candidates. Ciprian et al. [32] employed a cell grid architecture essentially based on a sequence of hash tables, specifically built for the targeted region. This method trained the cell with different features such as course, speed, etc. If the cell had not been trained before, then one algorithm would search for the closest best fit trained cell. This method worked well in the regional area and achieved 86% accuracy on the destination prediction. However, in terms of the destination prediction on a global level, it would be difficult and inaccurate to use the cell that was far away from the vessel's location for predicting. Oleh et al. [33] proposed a novel tree-based ensemble learning method to predict the vessel destination. This method takes the vessel destination as the responses for both training and predicting, and achieved an accuracy of 97% for the port destination.

Vessel destination prediction methods mentioned in related work [17–21] have achieved solid performances on regional vessel destination prediction. However, these research have not experimented with general vessel destination prediction. These methods use maritime records as inputs and destination locations as outputs. There are thousands of ports [13] all over the world and could be more than millions of combinations between two ports. Therefore, to expand their methods from predicting vessel destinations regionally to globally, the model has to be trained with more than millions of responses, which makes the model hard to converge. Apart from that, it should be considered that when new global ports are established, or new lanes are opened, the methods proposed in [17–21, 32] require to be reconstructed. Therefore, from the global destination prediction perspective, a

more general and updatable model needs to be constructed for predicting global vessels destinations. Furthermore, the resolution of global turning points clusters would be challenging to decide, and the model would have to be retrained with new turning points. Hence, compared with turning-point based prediction methods, trajectory-based prediction methods are more suitable for generally predicting the vessel destination.

2.2 AIS Data-Driven Vessel Trajectory Data Mining

As described in Section 2.1, compared with the methods via turning points data mining, trajectory data mining-based methods are more general and can be updated by new trajectory data. Focusing on general vessel destination prediction, this thesis proposed the trajectory similarity-based approach. This section is for reviewing the related state-of-the-art trajectory data mining methods, which consist of trajectory preprocessing and similarity measurements. In trajectory preprocessing, stay points should be detected first. Then, trajectories are segmented to measure the similarity between trajectories. 'Stay points detection' and 'trajectory segmentation methods' are reviewed in Section 2.2.1. In addition, existing state-of-the-art trajectory similarity measurements are illustrated in Section 2.2.2.

2.2.1 Trajectory Preprocessing

Stay point detection. Stay point detection aims to extract the points that denote locations where the moving object has stayed for a while. Fu et al. [34] proposed that stay points can be categorized into two types. One kind of points is where moving object remains stationary for a while. The other type of point is where the moving object moves around or remains stationary with positioning readings shifting around [34]. The stay points are widely used to transform the trajectory from the spatiotemporal points set into a sequence of points with the meaningful spaces [35, 36]. For instance, with the implementation of stay point detection on **AIS** data [37], points in the set are given a more detailed description about time and port that ships have ever docked.

Trajectory segmentation. **AIS** records [37] always contain historical trajectories that travel to a large number of ports. Hence, trajectories need to be divided into several segments for further data processing. In other words, segments bring us more knowledge about trajectory such as the sub-trajectory patterns which can contribute to trajectory pattern classification [38]. Segmentation techniques can be divided into four categories [34], namely, time-interval based segmentation [39], turning-point-based segmentation [40], key shape-point based segmentation [41, 42] and stay-point based segmentation [43].

2.2.2 Trajectory Similarity Measurements

The techniques of similarity measurement between trajectories can be classified into three categories: purely spatial similarity measures, purely temporal similarity measures, and spatiotemporal similarity measures [44–46]. For practical applications, the essential part of measuring the similarity of two trajectories is the spatial measurements. It should be noted that in purely spatial similarity measures, only the spatial information of the trajectories is taken into consideration for the similarity measurement. While in the purely temporal similarity measures, the spatial information of trajectories is neglected, and only the temporal information of them is considered for similarity analysis. In the spatiotemporal similarity, the spatial and temporal information from two trajectories is compared to measure their similarity. While predicting the destinations of the traveling vessels, the temporal information of current trajectories and historical trajectories are not likely to coincide. For this reason, purely temporal similarity measures cannot be suitable to be utilized for comparing traveling vessel trajectory with historical trajectories.

Purely Spatial Similarity Measure. The purely spatial similarity measure can be categorized into three distinctive classes [45, 46]: raw representation based similarity measurement, geometric shape based similarity measurement and movement direction based similarity measurement.

Raw representation-based similarity measurements are researched in papers [46, 47]. In work proposed by Faloutsos et al. [47], the sub-sequence matching based similarity measurement requires two trajectories to have the same

length, and it does not consider time-shifting for similarity measurement. Magdy et al. [46] concluded the efficiency of this method is heavily influenced when there exists noise in both trajectories. Hence, the raw representation based similarity measurements are not applicable for measuring the similarities between new trajectory and historical trajectories.

Regarding geometric shape-based similarity measurement, the Hausdorff distance [45], Fréchet distance [48], and Discrete Fréchet distance [49] are widely used for comparing the shape similarity of two given trajectories without the restricts of the length being the same. As researched in [45, 48, 49], these three methods work well when the two trajectories being compared have enough information to reflect the whole shape. When parts of the trajectory records are missing, and the shape of the trajectory cannot be represented, an inaccurate judgment may be given by the geometric shape-based similarity measurements. In the research [50], Symmetrized Segment-Path Distance (**SSPD**) is proposed. **SSPD** is a purely spatial similarity measurement method for comparing geometric shape between trajectories without restrictions on trajectories' length. **SSPD** compares trajectories as a whole, and tends to be less affected by incidental variation between trajectories. Moreover, **SSPD** also considers total length, the variation, and the physical distance between two trajectories into the calculation. The geometric shape-based similarity measurements - Hausdorff distance, Fréchet distance, Discrete Fréchet distance, and **SSPD** are included as the baselines to be compared with proposed **ML/** Deep Learning (**DL**) methods. These four geometric shape-based similarity measurement metrics are explained in details:

Hausdorff distance is a metric that can be used to measures how far two subsets of a metric space are from each other. Given two trajectories t_A and t_B, where t_A and t_B are the sets of trajectory points that denoted by a and b respectively, i.e., $a \in t_A$ and $b \in t_B$, the Hausdorff distance between trajectory t_A and t_B is defined as following:

$$h(t_A, t_B) = \max_{a \in t_A} \{ \min_{b \in t_B} \{ d(a,b) \} \}. \tag{2.1}$$

where $h(t_A, t_B)$ denotes the Hausdorff distance between trajectory t_A and t_B, and a and b are points of t_A and t_B, $d(a,b)$ is the Euclidian distance between point a and b.

Fréchet distance is a metric for measuring the length of the shortest distance that sufficient for both trajectories to traverse their separate paths. The Fréchet distance between two trajectories t_A of vessel A and t_B of vessel B is defined as follows:

$$Fr(t_A, t_B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \{||t_A(\alpha(t)), t_B(\beta(t))||_2\}. \tag{2.2}$$

where $Fr(t_A, t_B)$ denotes the Fréchet distance between trajectory t_A and t_B; inf represents infimum;$\alpha(t)$ and $\beta(t)$ represent continuous and increasing functions; $t_A(\alpha(t))$ and $t_B(\beta(t))$ are the positions of the vessel A and vessel B at time t.

Discrete Fréchet distance is for measuring the discrete trajectories while Fréchet distance is mainly for the consecutive trajectories.

$$Fr_d(t_A, t_B) = \min_{\gamma,\delta} \{\max_{s}\{||t_A(\gamma(s)), t_B(\delta(s))||_2\}\}. \tag{2.3}$$

where $Fr_d(t_A, t_B)$ denotes the Discrete Fréchet distance between trajectory t_A and t_B, and γ and δ are discrete non-decreasing functions for mapping.

SSPD is a shape-based distance between two trajectories, which takes the whole length of trajectories into consideration. Given t_A and t_B, the **SSPD** between t_A (including n_1 points that denoted as a_i, i.e., $a_i \in t_A$) and t_B is as following:

$$D_{SSPD}(t_A, t_B) = \frac{D_S(t_A, t_B) + D_S(t_B, t_A)}{2}, \tag{2.4}$$

$$D_S(t_A, t_B) = \frac{1}{n_1} \sum_{i=1}^{n_1} D_{PD}(a_i, t_B). \tag{2.5}$$

where D_{SSPD} denotes the **SSPD** between t_A and t_B; D_S denotes the segment-path distance distance between trajectories; D_{PD} is the per-

pendicular distance between one point a_i to the other trajectory t_B.

As for the movement direction based similarity measurement, the Angular Metric for Shape Similarity (**AMSS**) [51] is widely used to calculate the similarity based on the directions of the trajectories. This method can be used to compare the trajectories with different lengths and avoid the inaccuracy caused by the missing records. Besides, the Edit Distance on Trajectories Pattern (**EDTP**) [52] is capable of finding similar trajectories with different spatial rotation and scaling factors. The trajectory can have missing records. Under this circumstance, the line pattern could bring inaccuracy to similarity judgment.

*Dynamic Time Warping (**DTW**)* is one algorithm for measuring similarity between two trajectories. Given two trajectories t_A: $[a_1, ..., a_n]$ and t_B: $[b_1, ..., b_m]$ of length n and m, **DTW** between t_A and t_B is notated as $DTW(t_A, t_B)$ and calculated by:

$$\begin{cases} 0, \ \ if\ m = n = 0 \\ \infty, \ \ if\ m = 0\ or\ n = 0 \\ dist_{dtw}(a_1, b_1) + \min \begin{Bmatrix} DTW(Rest(t_A), Rest(t_B)), \\ DTW(Rest(t_A), t_B), \\ DTW(t_A, Rest(t_B)) \end{Bmatrix} & otherwise \end{cases}, \tag{2.6}$$

$$dist_{dtw}(a_i, b_1) = \begin{cases} ||a_i - b_i||_2 & if\ a_i, b_i\ not\ gaps \\ ||a_i - b_{i-1}||_2 & if\ b_i\ is\ a\ gap \\ ||b_i - a_{i-1}||_2 & if\ a_i\ is\ a\ gap \end{cases}. \tag{2.7}$$

where $Rest(t_A)$ and $Rest(t_B)$ represent the rest trajectories that without the first records.

*Longest Common Subsequence (**LCSS**)* is a similarity that defined by the number of time steps that two trajectories match. Given two trajectories t_A and t_B, and these two trajectories are arranged to form a $m \times n$ grid,

the **LCSS** between t_A and t_B is as following:

$$D_{LCSS}(t_A, t_B) = 1 - \frac{LCSS(t_A, t_B)}{\min(m, n)}. \tag{2.8}$$

where $D_{LCSS}(t_A, t_B)$ denotes **LCSS** between t_A and t_B, and $LCSS(t_A, t_B)$ the number of matching points between t_A and t_B.

*Edit Distance with Real Penalty (**ERP**)* uses real penalty for not only for two consistent elements, but also for the case of calculating the distance for gaps. Given two trajectories $t_A : [a_1, ..., a_n]$ and $t_B : [b_1, ..., b_m]$ of length n and m, the **ERP** between t_A and t_B is notated as $ERP(t_A, t_b)$ and calculated by:

$$\begin{cases} \sum_{i=1}^{n} ||b_i - g||_2 & if\ m = 0 \\ \sum_{i=1}^{m} ||a_i - g||_2 & if\ n = 0 \\ \min \begin{cases} ERP(Rest(t_A), Rest(t_B)) + dist_{erp}(a_1, b_1), \\ ERP(Rest(t_A), t_B) + dist_{erp}(a_1, gap), \\ ERP(t_A, Rest(t_B)) + dist_{erp}(b_1, gap) \end{cases} & otherwise \end{cases}, \tag{2.9}$$

$$dist_{erp}(a_i, b_1) = \begin{cases} ||a_i - b_i||_2 & if\ a_i, b_i\ not\ gaps \\ ||a_i - g||_2 & if\ b_i\ is\ a\ gap \\ ||b_i - g||_2 & if\ a_i\ is\ a\ gap \end{cases}. \tag{2.10}$$

where g is a constant value; $Rest(t_A)$ and $Rest(t_B)$ represent the rest trajectories that without the first records.

*Edit Distance on Real sequences (**EDR**)* is the metric used for measuring the similarity between two trajectories by counting the amount of operations, e.g., delete, replace or insert, that are needed for changing one trajectory to the other. Given two trajectories $t_A : [a_1, ..., a_n]$ and $t_B : [b_1, ..., b_m]$ of length n and m, the **EDR** between t_A and t_B is notated

as $EDR(t_A, t_b)$ and calculated by:

$$\begin{cases} n & if\ m = 0 \\ m & if\ n = 0 \\ \min \begin{cases} EDR(Rest(t_A), Rest(t_B)) + subcost, \\ EDR(Rest(t_A), t_B) + 1, \\ EDR(t_A, Rest(t_B)) + 1 \end{cases} & otherwise \end{cases}. \tag{2.11}$$

where $subcost = 0$ if the distance between a_1 and a_2 are smaller than the matching threshold, and $Rest(t_A)$ and $Rest(t_B)$ represent the rest trajectories that without the first records.

Machine Learning (**ML**) refers to a vast set of tools for understanding data and learning the patterns from historical data [53]. For **ML**, the features need to be extracted by some sophisticated methods before the training process. The performance of the **ML** is highly correlated to the feature's quality and model's capability. Deep Learning (**DL**) is a subset of **ML** methods and learns patterns from the massive historical data based on Artificial Neural Networks (**ANN**), whose algorithm is similar to the human brain's functionality [1]. With the guidance of responses, the **DL** can tune itself to capture representations of data for achieving better performance. Three advanced architectures of **DL** [1] are Multilayer Perceptron (**MLP**), Recurrent Neural Network (**RNN**) and Convolutional Neural Network (**CNN**). The state-of-the-art **RNN** structure is Independently Recurrent Neural Network (**IndRNN**) [54], which is dominating in the time series analysis. Moreover, the **CNN** is powerful in image pattern recognition. However, little research has been done on both traditional **ML** and **DL** methods in trajectory similarity measurement. Hence, four **ML**-based similarity measurement models are proposed and investigated in this thesis, of which two are **DL**-based. These four methods are introduced in Section 3.6.

2.3 Summary

This chapter provides a comprehensive review of **AIS** data-driven predictive applications and methodologies. The regional vessel destination prediction is well researched in the area of AIS data-driven prediction. However, these regional vessel destination prediction methods suffer from the limits on regions and candidate port destinations and are scarcely possible to generally predict the destinations for global vessels. For implementing general vessel destination prediction, this thesis proposes a model using the methodologies of stay point detection, trajectory segmentation, and similarity measurement. Hence, the reviews of correlated trajectory data mining research, which includes the trajectory preprocessing and similarity measurement, are also provided in this chapter.

Chapter 3

Methodology for AIS Data-Driven General Vessel Destination Prediction

In this chapter, a novel **AIS** data-driven method is proposed for the general vessel destination prediction. The descriptions for the proposed method are structured as follows. The definitions and notations used in this chapter are explained in Section 3.1. The overall structure of the **AIS** data-driven general vessel destination prediction method is presented in Section 3.2, where a brief overview of the proposed method's components is also given. In Section 3.3, the proposed **DBSCAN**-based trajectory segmentation method is described, along with the illustration of constructing the historical trajectory database from **AIS** records and global ports locations. The preprocessing on traveling and historical trajectories are presented in Section 3.4, where the illustrations of sampling the traveling trajectories and querying the historical trajectories are also provided. The processes of extracting the feature between traveling and historical trajectories are described with illustration in Section 3.5. In Section 3.6, four proposed **ML**-based similarity measurement models are introduced in detail. The two proposed decision strategies are explained in Section 3.7. In the Section 3.8, the employed evaluation metrics are described in detail.

3.1 Definitions and Notations

The definitions and notations used in this chapter are given as follows.

Definition (Trajectory). *A trajectory $\boldsymbol{t}$ is a sequence of points with orders, i.e., $\boldsymbol{t} = l_1 \rightarrow \cdot \rightarrow l_n \rightarrow \cdot \rightarrow l_K$, where l is a location with coordinates (latitudes, longitudes, etc.); K is the number of coordinate records in $\boldsymbol{t}$, and can be any positive integer.*

Definition (Sampled Trajectory). *Given a trajectory $\boldsymbol{t} = l_1 \rightarrow \cdot \rightarrow l_n \rightarrow \cdot \rightarrow l_K$, $\boldsymbol{s} = \ell_1 \rightarrow \cdot \rightarrow \ell_m \rightarrow \cdot \rightarrow \ell_R$ is the sampled trajectory from $\boldsymbol{t}$, denoted by $\boldsymbol{s} \subset \boldsymbol{t}$, and $l_1 = \ell_1$ and $l_K = \ell_R$.*

Definition (Historical Trajectories). *Let $\boldsymbol{T} = \{\boldsymbol{t}^{(1)}_{(a\rightarrow b)}, ..., \boldsymbol{t}^{(P)}_{(\chi\rightarrow\psi)}\}$ be the set of all historical trajectories where a, b, χ, ψ are arbitrary ports and P is the number of historical trajectories in the set. Therefore, the $\boldsymbol{T} \in \mathbb{R}^{P\times K}$.*

For example, if a vessel departs from port a, $\boldsymbol{T}_{(a)} = \{\boldsymbol{t}^{(1)}_{(a\rightarrow b)}, ..., \boldsymbol{t}^{(P)}_{(a\rightarrow\psi)}\}$ is the set of all trajectories from a to all the destination ports $\{b, ..., \psi\}$ in history.

Except the above definitions, other notations used in this chapter and their descriptions are presented in Table 3.1.

Table 3.1. Notations and descriptions.

Notations	***Descriptions***
Indices	
n	Index of coordinate records in a trajectory
m	Index of coordinate records in a sampled trajectory
q	Index of trajectories in a set
a	Index of departure ports
b	Index of destination ports
j	Index of decision trees in the random forest
Variables	
l	Coordinate record in a trajectory
ℓ	Coordinate record in a sampled trajectory

Notations	***Descriptions***
K	Number of coordinate records in a trajectory
R	Number of coordinate records in a sampled trajectory
P	Number of trajectories in trajectory set
$\boldsymbol{t}$	$\boldsymbol{t} = l_1 \rightarrow \cdot \rightarrow l_n \rightarrow \cdot \rightarrow l_K$ trajectory of all coordinate records
$\boldsymbol{s}$	$\boldsymbol{s} = \ell_1 \rightarrow \cdot \rightarrow \ell_m \rightarrow \cdot \rightarrow \ell_R$ sampled trajectory derived from $\boldsymbol{t}$
$\boldsymbol{t}_{(a)}$	Trajectory of one vessel traveling from port a
$\boldsymbol{s}_{(a)}$	Sampled trajectory derived from $\boldsymbol{t}_{(a)}$
$\boldsymbol{t}^{(q)}_{(a\rightarrow b)}$	Trajectory of one vessel traveling from port a to port b
$\boldsymbol{T}$	$\boldsymbol{T} = \{\boldsymbol{t}^{(1)}_{(a\rightarrow b)}, ..., \boldsymbol{t}^{(P)}_{(\chi\rightarrow\psi)}\}$ set of all historical trajectories, χ and ψ represent that ports different from a and b
$\boldsymbol{T}_{(a)}$	Set of all historical trajectories from port a
$\boldsymbol{L}_{AIS}$	Set of AIS coordinate records
$\boldsymbol{L}_{port}$	Set of global port coordinates
$\boldsymbol{d}$	Set of perpendicular distances between trajectory point ℓ_m and vector $\overrightarrow{l_n l_{n-1}}$ in $\boldsymbol{t}_{(a)}$
d_m	$d_m = \min(\boldsymbol{d})$ shortest perpendicular distance (SPD) from trajectory point ℓ_m in $\boldsymbol{s}_{(a)}$ to trajectory $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$
dr	Ratio of haversine distance between the departure port and the traveling vessel to the haversine distance between traveling vessel and the destination port of the compared historical trajectory
$\boldsymbol{cr}$	$\boldsymbol{cr} = \{d_1, ..., d_m, ..., d_R, dr\}$ Set of comparison feature between sampled traveling trajectory $\boldsymbol{s}_{(a)}$ and trajectory $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$
$\boldsymbol{cr}^{(q)}_{(a\rightarrow b)}$	Comparison feature between $\boldsymbol{s}_{(a)}$ and $\boldsymbol{t}^{(q)}_{(a\rightarrow b)}$
$\boldsymbol{C}_{(a)}$	Set of comparison features $\boldsymbol{cr}$ between $\boldsymbol{s}_{(a)}$ and all trajectories $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$
$y^{(q)}_{(a\rightarrow b)}$	Similarity between $\boldsymbol{s}_{(a)}$ and $\boldsymbol{t}^{(q)}_{(a\rightarrow b)} \in \boldsymbol{T}_{(a)}$
$\boldsymbol{y}_{(a)}$	$\boldsymbol{y}_{(a)} = \{y^{(1)}_{(a\rightarrow b)}, ..., y^{(P)}_{(a\rightarrow\psi)}\} \in \mathbb{R}^P$ set of similarities between $\boldsymbol{s}_{(a)}$ and all trajectories in $\boldsymbol{T}_{(a)}$
p	Number of decision trees in the random forest
β_j	$\sum_{j=1}^{p} \beta_j = 1$ Weight assigned to the j-th decision tree

Notations	*Descriptions*
$freq_{(a\to b)}$	Frequency of a vessel traveling from port a to another port b
$N(\boldsymbol{T}_{(a\to b)})$	Number of trajectories in $\boldsymbol{T}_{(a)}$ that traveled from port a to b
$\boldsymbol{freq}_{(a)}$	Set of destination ports frequencies from departure port a
$\zeta^{(q)}_{(a\to b)}$	Normalized similarity for $y^{(q)}_{(a\to b)}$
$\boldsymbol{\zeta}_{(a)}$	Set of normalized similarities of all the destinations from port a
Functions	
$D(.)$	Function of calculating perpendicular distance between point ℓ_m and vector $\overrightarrow{l_n l_{n-1}}$, i.e., $D(\ell_m, l_n, l_{n-1})$ where l_i is a location in $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$ while l_{n-1} is the precedent location of l_n
$H(.)$	Function of calculating distance ratio dr
$Haversine(.)$	Function of calculating haversine distance between two points
$h_j(.)$	Function of bagging (bootstrap sampling) for j-th decision tree in random forest
$f_j(.)$	Function of j-th decision tree in random forest

3.2 Overall Description of AIS Data-Driven General Vessel Destination Prediction

The proposed general vessel destination prediction can be implemented through the following four steps:

1. The vessel's traveling trajectory is first constructed with its AIS records by the proposed **DBSCAN**-based trajectory segmentation method.

2. The traveling trajectory is compared with corresponding historical trajectories in the database, which share the same departure ports. The comparison generate a set (comparison feature) consisting of perpendicular distances and the distance ratio between two trajectories.

3. The derived set (comparison feature) is then fed into the proposed **ML**-based model to measure the similarity between two trajectories. The similarity is defined as the probability that the two trajectories share the same destination.

4. The similarities together with the potential ports' attribute are used to support the decision process. The destination with the highest possibility will be identified.

One example in Fig. 3.1 is given to illustrate the procedure. The red curve represents the trajectory of a vessel traveling from Vancouver. The blue curve is a historical trajectory between Vancouver and Yokohama, and the black one is another historical trajectory between Vancouver and Rupert. To predict the destination of vessel from the red trajectory, the steps are:

1. The AIS records are clustered with the port points. The departure port - Vancouver and the traveling trajectory (red line) are then extracted for further analysis.

2. The traveling trajectory (Red) is compared with historical trajectories (Blue and Black) respectively. The comparisons generate two comparison features that consist of perpendicular distances and the distance ratio between traveling and historical trajectories.

3. The trajectory similarity measurement model then measures the similarity between traveling and historical trajectories based on the derived comparison features.

4. The similarities together with the potential ports' attribute are used to support the decision process. The destination with the highest possibility will be identified.

The flowchart in Fig. 3.2 summarizes the the overall methodology. Given the coordinate sets of one vessel's **AIS** data (records) $\boldsymbol{L}_{AIS}$ and global ports $\boldsymbol{L}_{port}$, the proposed model first extracts the traveling trajectory $\boldsymbol{t}_{(a)}$, the traveling vessel's departure port a, and historical trajectories. Both the traveling trajectory and historical trajectories extractions processes are described in Section 3.3. The extracted historical trajectories are used for updating the historical trajectory database. For predicting the traveling vessel destination, a sampling operation is applied to generate a sampled traveling trajectory $\boldsymbol{s}_{(a)}$ from $\boldsymbol{t}_{(a)}$. Meanwhile, the historical trajectory database $\boldsymbol{T}$ is queried with the key of departure port a, and all historical

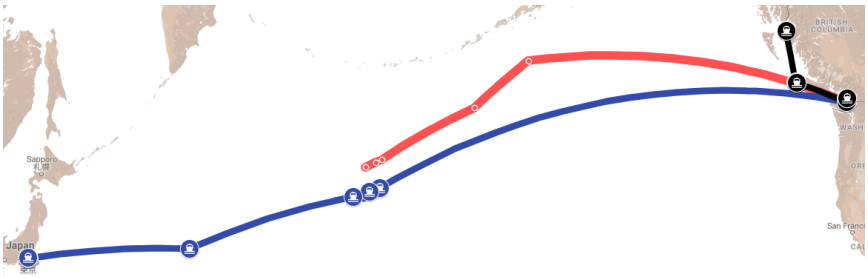

Figure 3.1. Illustration of trajectories comparisons between traveling vessel trajectory (red line) and historical trajectories (blue and black lines).

trajectories from port a are selected and stored in the set, namely $\boldsymbol{T}_{(a)}$. Both the sampling and query process are described in Section 3.4. Then, the sampled traveling trajectory $\boldsymbol{s}_{(a)}$ is compared with each trajectory in $\boldsymbol{T}_{(a)}$, and the comparison features are presented in a set named as $\boldsymbol{C}_{(a)}$. The explanations of the trajectory comparison process and the related terminologies are given in Section 3.5. The comparison feature set $\boldsymbol{C}_{(a)}$ is then fed into the **ML**-based similarity measurement model for calculating the similarity between sampled traveling trajectory $\boldsymbol{s}_{(a)}$ and each historical trajectory in $\boldsymbol{T}_{(a)}$. The similarities are then collected in the set denoted as $\boldsymbol{y}_{(a)}$. The **ML**-based similarity measurement model and the related terminologies are explained in Section 3.6. Besides, the frequencies of destination ports from departure port a, namely $\boldsymbol{freq}_{(a)}$, are obtained from the $\boldsymbol{T}_{(a)}$. Finally, the prediction of destination port of the traveling vessel is determined based on the frequencies of destination ports $\boldsymbol{freq}_{(a)}$ and similarities $\boldsymbol{y}_{(a)}$. The descriptions of both the calculation of $\boldsymbol{freq}_{(a)}$ and the decision process are given in Section 3.7.

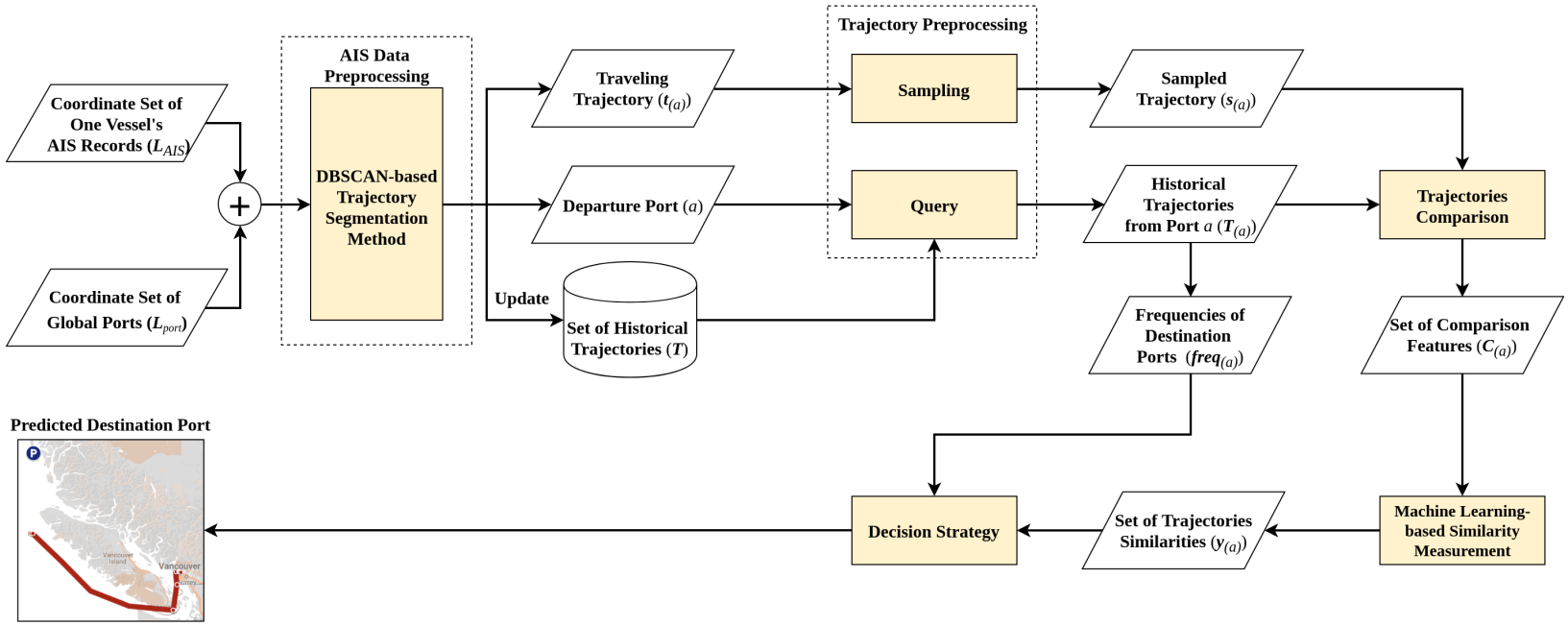

Figure 3.2. Overall framework of the **AIS** data-driven general vessel destination prediction model.

3.3 AIS Data Preprocessing - DBSCAN-based Trajectory Segmentation Method

The information on whether the vessel has arrived at the port and which port the vessel has arrived at is not indicated in **AIS** records. In order to segment the trajectories from **AIS** records, the port stay points need to be first identified. A vessel may stay near a port for a few days with the position shifting around the port before heading to the next destination. When a vessel stays in a port for a few days, it may have multiple positions (points) around the port before heading to the next destination. These shifting "points" are defined as "port stay points." Considering that the positions of the vessel are reported by **AIS** with a fixed frequency, these stay-points in the destination port area are denser than the points reported by **AIS** when the vessel is sailing. In this way, if a vessel enters a zone with dense stay-points and port, this vessel is then regarded as arriving at the destination. In this thesis, we aim to provide a general and flexible method to detect the stay points and construct trajectories of each vessel in history from the **AIS** records database.

The original **AIS** records database contains all historical locations of vessels. The global port list contains all the locations of ports in the world with precise geographical locations such as longitude and latitude. Herein, $\boldsymbol{L}_{AIS}$ and $\boldsymbol{L}_{port}$ represent the coordinate sets of **AIS** records and global ports accordingly. The Density-Based Spatial Clustering of Applications with Noise (**DBSCAN**) [55] is used in this study to cluster the points in $\boldsymbol{L}_{AIS} \cup \boldsymbol{L}_{port}$ with the following parameters:

- eps : 1.095×10^{-5}
- $min\_samples$: 2
- $algorithm$: $ball\_tree$
- $Distance function$: $haversine$

The eps is a parameter specifying the radius of a neighborhood with respect to other points. For the purpose of **DBSCAN** clustering, the points are classified as core, reachable and noise points, as follows:

- A point is a core point if at least $min\_samples$ points are within distance eps of it.

- A point is a reachable point if it is within distance eps from any core point.
- All points not reachable from any other point are noise points.

Herein, the processes of **DBSCAN** algorithm are abstracted into the following four steps:

1. Calculate the distances between every two points.
2. Identify the core points with more than $min\_samples$ neighbors based on the given distance of eps.
3. Find the connected components of core points on the neighbor graph, ignoring all non-core points.
4. Assign each non-core point to a nearby cluster if the cluster is a eps neighbor, otherwise assign it to noise.

Figure 3.3 illustrates the process of extracting the traveling trajectory and historical trajectory from the **AIS** records. First of all, the **DBSCAN** is employed to cluster the points of **AIS** records and global ports. If points concentrate on an area, the area will be labeled as a cluster. Moreover, if the points on an area are not dense enough, the points will be labeled as in a cluster of noise. After labeling all points with clusters by **DBSCAN**, the cluster including a port in $\boldsymbol{L}_{port}$ will be regarded as a stay-point cluster which are the circles of red dashed line in Fig. 3.3. Otherwise, the rest of the points will remain as a trajectory point in $\boldsymbol{L}_{AIS}$. Finally, the trajectory points of a vessel are grouped as a whole trajectory between every two stay-points in history. As shown in Fig. 3.3, the historical trajectory (black line) has been extracted and segmented as the trajectory from Port A to Port B. The trajectory of the vessel that just departs from one port and has not arrived at the destination port is then regarded as the traveling trajectory, e.g., the trajectory in green line in Fig. 3.3 is regarded as one traveling trajectory from port A. Through duplicating the proposed **DBSCAN**-based trajectory segmentation method on **AIS** records of different vessels, the historical trajectories of vessels between different ports are generated. The historical trajectory database is constructed by collecting these extracted historical trajectories and denoted as $\boldsymbol{T}$.

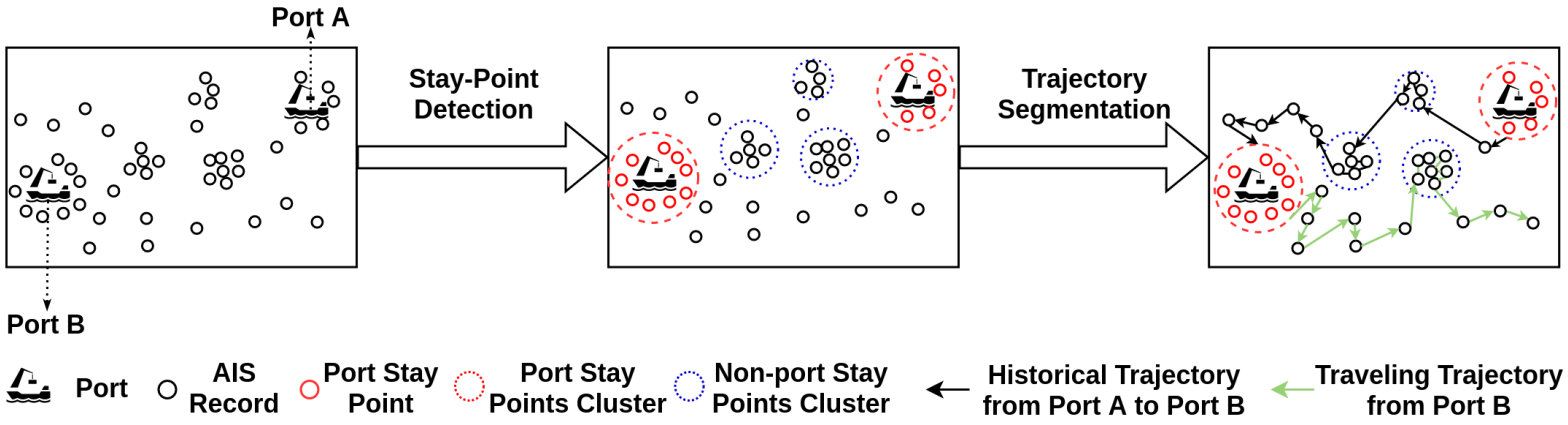

Figure 3.3. Representation of historical trajectories and traveling trajectory extraction.

3.4 Trajectory Preprocessing - Sampling and Query

Sampling. Due to different regulations and other conditions among vessels, different traveling vessels report their locations in different frequencies, which indicate the lengths of trajectory sequences are various. The average number of trajectory points within a day in all historical trajectories is 2.834. Therefore, in order to make the trajectory sequences of the traveling vessels be the same length, only the first point and the last point of the trajectory $\boldsymbol{t}$ within a day are selected for constructing the sampled trajectory. When the first point and the last point are the same point, this point will be duplicated in the sampled trajectory $\boldsymbol{s}$. Figure 3.4 shows the procedure of sampling from trajectory $\boldsymbol{t}$ to sampled trajectory $\boldsymbol{s}$. Besides, the trajectories missing records over one-day or lasting less than one day, which are rare, are excluded from the training and testing processes.

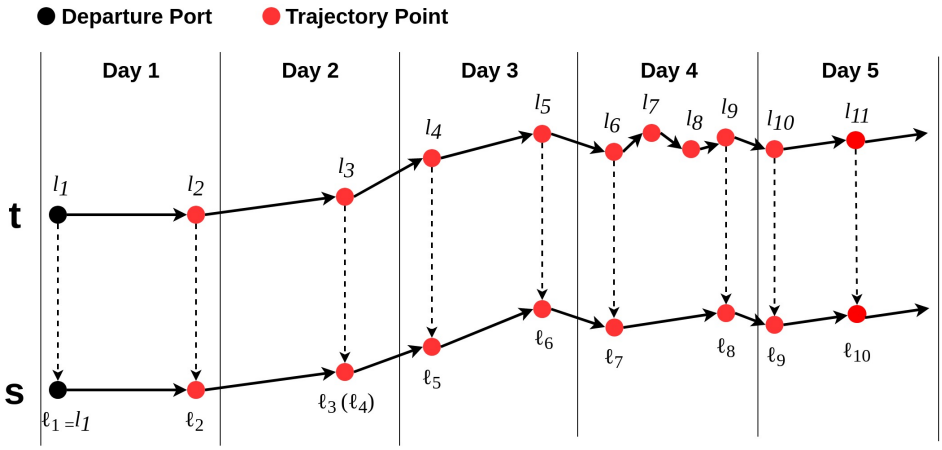

Figure 3.4. The illustration of sampling trajectories from $\boldsymbol{t}$ to $\boldsymbol{s}$; Oversampling happens when there is only one point in one day, e.g., l_3 of t in Day 2 is oversampled as ℓ_3 and ℓ_4 in s.

Query. The historical trajectories, which have the same departure port with the traveling trajectory, are selected to be compared against the traveling trajectory. Given a sampled trajectory $\boldsymbol{s}_{(a)} = \ell_1 \rightarrow \cdots \rightarrow \ell_R$ departing from port a. The key - departure port a is then used to query all historical trajectories from port a in $\boldsymbol{T}_{(a)}$. The queried historical trajectories are finally collected in a set $\boldsymbol{T}_{(a)}$ for being compared with the sampled trajectory $\boldsymbol{s}_{(a)}$.

3.5 Trajectories Comparison

A sampled trajectory $\boldsymbol{s}_{(a)}$ is compared with each trajectory in the set of historical trajectories $\boldsymbol{T}_{(a)}$ from port a. Two types of distances, i.e., haversine distance [56] and perpendicular distance [57], are employed to generate the "comparison feature" $\boldsymbol{cr}$ between $\boldsymbol{s}_{(a)}$ and each historical trajectory $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$. Then, the comparison features between $\boldsymbol{s}_{(a)}$ and all historical trajectories in $\boldsymbol{T}_{(a)}$ are collected in one set (denoted as "set of comparison feature" $\boldsymbol{C}_{(a)}$). The set of comparison feature $\boldsymbol{C}_{(a)}$ is then fed into the similarity measurement model that described in Section 3.6 for generating the similarities between $\boldsymbol{s}_{(a)}$ and all historical trajectories in $\boldsymbol{T}_{(a)}$. For making the reader better understand trajectory comparison, the process of extracting the comparison feature between traveling and historical trajectory is defined as shown in Algorithm 1, and illustrated in Fig. 3.5.

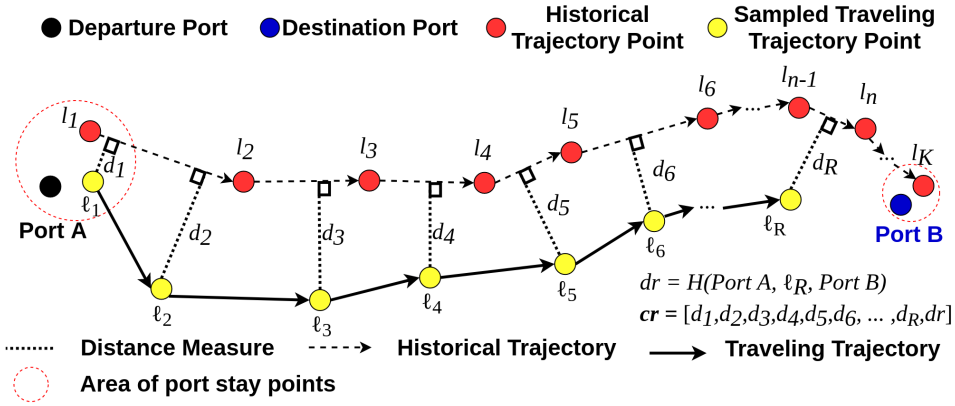

Figure 3.5. The demonstration of generating the comparison feature cr (including perpendicular distances$\{d_1, ..., d_m, ..., d_R\}$ and distance ratio dr) between traveling and historical trajectories.

Algorithm 1 Representation of extracting the comparison feature between traveling vessel trajectory and historical trajectories

Input: Traveling vessel trajectory: $\boldsymbol{s}_{(a)}$
Historical trajectories: $\boldsymbol{T}_{(a)}$
Output: $\boldsymbol{C}_{(a)}$

$\boldsymbol{C}_{(a)} \leftarrow [\,]$
for $\boldsymbol{t}$ in $\boldsymbol{T}_{(a)}$ **do**
  $\boldsymbol{cr} \leftarrow [\,]$
  for traveling location ℓ_m from ℓ_1 to ℓ_R in $\boldsymbol{s}_{(a)}$ **do**
    $\boldsymbol{d} \leftarrow [\,]$
    for historical locations l_n from l_2 to l_K in $\boldsymbol{t}$ **do**
      $\boldsymbol{d} \leftarrow \boldsymbol{d} + [D(\ell_m, l_n, l_{n-1})]$
    end for
    $d_m \leftarrow \min(\boldsymbol{d})$
    $\boldsymbol{cr} \leftarrow \boldsymbol{cr} + [d_m]$
  end for
  $dr \leftarrow H(Port\ A, \ell_R, Port\ B)$
  $\boldsymbol{cr} \leftarrow \boldsymbol{cr} + [dr]$
  $\boldsymbol{C}_{(a)} \leftarrow \boldsymbol{C}_{(a)} + [\boldsymbol{cr}]$
end for
return $\boldsymbol{C}_{(a)}$

Algorithm 1 shows algorithmic processes of extracting comparison feature $\boldsymbol{cr} = \{d_1, ..., d_m\ , ..., d_R, dr\}$, $\boldsymbol{cr} \in \mathbb{R}^{(R+1)}$. $\{d_1, ..., d_m, ..., d_R\}$ is the set of Shortest Perpendicular Distance (**SPD**) between trajectory points $\boldsymbol{s}_{(a)}$ and the compared historical trajectory. In the Algorithm 1, the function D(.) is implemented as Eq. 3.1 to calculate the perpendicular distance between the trajectory point and each vector in $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$, e.g. $\overrightarrow{l_2 l_1}$ in Fig. 3.5.

$$D(\ell_m, l_n, l_{n-1}) = \frac{||\overrightarrow{l_n \ell_m} \times \overrightarrow{l_n l_{n-1}}||}{||\overrightarrow{l_n l_{n-1}}||}, \tag{3.1}$$

where ℓ_m is a location in $\boldsymbol{s}_{(a)}$; l_n is a location in $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$ while l_{n-1} is the precedent location of l_n; $||.||$ represents the norm function, which assigns a strictly positive length or size to each vector in a vector space. Then, the Eq. 3.1 is used to greedily calculate perpendicular distances from the point on traveling trajectory to every

vector $\overrightarrow{l_i l_{i-1}}$ in historical trajectory. The $\boldsymbol{d}$ in Algorithm 1 is set for collecting these perpendicular distances, and the $\min(\boldsymbol{d})$ is defined as the Shortest Perpendicular Distance (**SPD**). According to Fig. 3.5, one example is given for illustrating the calculation of **SPD** d_R for point ℓ_R:

$$d_R = \min\{D(\ell_m, l_2, l_1), ..., D(\ell_m, l_n, l_{n-1}), ..., D(\ell_m, l_K, l_{K-1})\}, \tag{3.2}$$

The traveling distance of the vessel is essential for making the vessel destination prediction. Hence, the distance ratio dr between $\boldsymbol{s}_{(a)}$ and $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$, is proposed for representing whether the traveling vessel is close to departure port or the destination of compared historical trajectory. "distance ratio" (dr) is defined as the ratio of two haversine distances. The numerator is the distance between the departure port and the traveling vessel. The denominator is the distance between the traveling vessel and the destination port from the historical trajectory. According to Fig. 3.5 and Eq. 3.3, the calculation of the distance ratio dr is illustrated.

$$dr = H(PortA, \ell_R, PortB) = \frac{Haversine(PortA, \ell_R)}{Haversine(\ell_R, PortB)}, \tag{3.3}$$

where $Haversine(.)$ is the function to measure the haversine distance between two given points. *PortA* is the coordinate of departure port, and ℓ_R is the latest record in $\boldsymbol{s}_{(a)}$; *PortB* is the destination coordinate of historical trajectory $\boldsymbol{t} \in \boldsymbol{T}_{(a)}$.

The comparison features between $s_{(a)}$ and all historical trajectories are then collected in the set of comparison feature $\boldsymbol{C}_{(a)}$, which is illustrated as follows:

$$\boldsymbol{C}_{(a)} = \begin{bmatrix} \boldsymbol{cr}^{(1)}_{(a\to b)} \\ \vdots \\ \boldsymbol{cr}^{(P)}_{(a\to \psi)} \end{bmatrix} = \begin{bmatrix} d_1^{(1)} & \cdots & d_m^{(1)} & \cdots & d_R^{(1)} & dr^{(1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ d_1^{(P)} & \cdots & d_m^{(P)} & \cdots & d_R^{(P)} & dr^{(P)} \end{bmatrix}. \tag{3.4}$$

where $\boldsymbol{C}_{(a)} \in \mathbb{R}^{P\times(R+1)}$; P is the number of trajectories in $\boldsymbol{T}_{(a)}$. Considering each comparison feature between traveling trajectory and compared historical trajectory has R perpendicular distances and distance ratio dr from a to each historical destination ports in $\{b \cdots \psi\}$. Therefore, $\boldsymbol{C}_{(a)} \in \mathbb{R}^{P\times(R+1)}$.

3.6 Machine Learning-based Similarity Measurements

After obtaining the set of comparison feature $\boldsymbol{C}_{(a)}$ of the traveling vessel, the next step is to calculate the similarity of each historical trajectory based on its correlated comparison feature $\boldsymbol{cr}$. The four proposed Machine Learning (**ML**)-based similarity measurement models are introduced in the following subsections.

3.6.1 Naive Bayes-based Similarity Measurement Method

The Naive Bayes classifier is implemented based on the Bayes theorem [58]. The Naive Bayes classifier predicts the probability of the input sample's class based on the independence assumptions between predictors. In this thesis a Gaussian Naive Bayes classifier is implemented for measuring the similarity between trajectories. Given a set of comparison feature $\boldsymbol{cr}^{(1)}_{(a\to b)} = \{d_1, ..., d_R, dr\}$ in $\boldsymbol{C}_{(a)}$ between sampled traveling vessel trajectory $\boldsymbol{s}_{(a)}$ and the historical trajectory $\boldsymbol{t}^{(1)}_{(a\to b)}$, the calculation process of similarity (two trajectories have the same destination) between $\boldsymbol{s}_{(a)}$ and $\boldsymbol{t}^{(1)}_{(a\to b)}$ is shown in the Equation 3.5.

$$y^{(1)}_{(a\to b)} = P(same) \times P(dr|same) \times \prod_{m=1}^{R} P(d_m|same). \tag{3.5}$$

where the "same" represents the target that two trajectories have the same destination. The likehood of the feature is assumed to be Gaussian.

3.6.2 IndRNN-based Similarity Measurement Method

Recurrent Neural Network (**RNN**) is a class of **ANN** widely used for processing time series that contain non-negligible mutual relation. The connections among neurons in **RNN** are directed cycle for manipulation of time sequential data. The **RNN** is named as "recurrent" because it performs the same task for every part of the input time series. The output of the **RNN** depends on the computation of previous layers, which is different from some common neural networks like **MLP**. The illustraton of **RNN** is shown as Fig. 3.6. **IndRNN** is the state-of-the-art structure of the Recurrent Neural Network (**RNN**) which addresses the gradient exploding

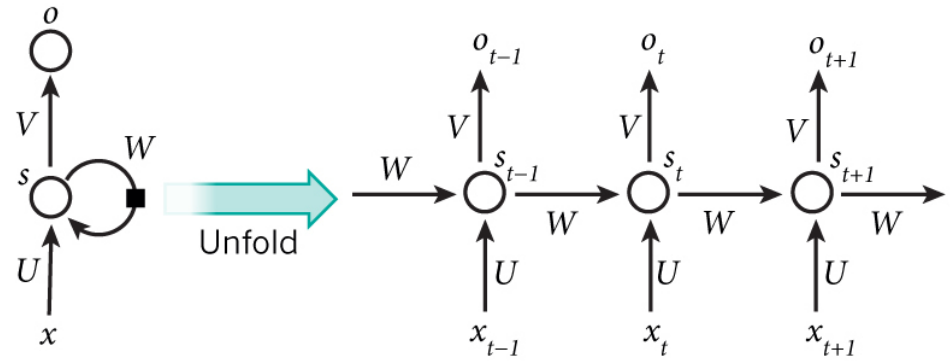

Figure 3.6. A recurrent neural network and its unfolding in time of the computation involved in its forward computation [1].

and vanishing problems [54]. The **IndRNN** can be mathematically defined as:

$$\boldsymbol{h}_t = \Delta(\boldsymbol{W}x_t + \boldsymbol{u} \odot \boldsymbol{h}_{t-1} + \boldsymbol{b}), \tag{3.6}$$

where $\boldsymbol{h_t}$ and x_t are the hidden state at time step t and input accordingly; $\boldsymbol{W}, \boldsymbol{u}, \boldsymbol{b}$ are weights for the current input and recurrent input, and the bias of the neurons respectively; $\Delta(.)$ is an element-wise activation function of the neurons; $\odot$ represents Hadamard product [54]. Given a set of comparison feature $\boldsymbol{cr}^{(1)}_{(a\to b)} = \{d_1, ..., d_R, dr\}$ in $\boldsymbol{C}_{(a)}$ between sampled traveling vessel trajectory $\boldsymbol{s}_{(a)}$ and the historical trajectory $\boldsymbol{t}^{(1)}_{(a\to b)}$, the propagation process between input layer and the first hidden layer $\boldsymbol{h_1}$ of then is defined as the Equation 3.7.

$$\boldsymbol{h}_1(\boldsymbol{cr}^{(1)}_{(a\to b)}) = \Delta(w_0 dr + b_0 + \sum_{m=1}^{R} (w_m d_m + b_m)), \tag{3.7}$$

where w_m and w_0 represent the weights associated with each input feature; u_0 represents the weight associated with recurrent input; b_m and b_0 represent the bias associated with each neuron in the first hidden layer. According to Fig. 3.6, after multiple layers' forward computation, the network outputs the similarity $y^{(1)}_{(a\to b)}$:

$$y^{(1)}_{(a\to b)} = \delta(\boldsymbol{h}_t) = \frac{1}{1 + e^{-\boldsymbol{h}_t}}. \tag{3.8}$$

where $y^{(1)}_{(a\to b)} \in [0,1]$ is the similarity between $\boldsymbol{s}_{(a)}$ and the historical trajectory $\boldsymbol{t}^{(1)}_{(a\to b)}$; $\boldsymbol{h}_t$ is the output of the last hidden layer, which is defined in Eq. 3.6. $\delta(.)$ represents the sigmoid function.

3.6.3 MLP-based Similarity Measurement Method

Multilayer Perceptron (**MLP**) is the most common model of **DL**. The structure of **MLP** includes one input layer, hidden layers, and one output layer. Each **MLP** layer consists of a massive number of neurons. Furthermore, each neuron is fully connected with all the neurons in the previous layer, which is similar to synapses in the human brain [1]. Given a set of comparison feature $\boldsymbol{cr}^{(1)}_{(a\to b)} = \{d_1, ..., d_R, dr\}$ in $\boldsymbol{C}_{(a)}$ between sampled traveling vessel trajectory $\boldsymbol{s}_{(a)}$ and the historical trajectory $\boldsymbol{t}^{(1)}_{(a\to b)}$, the propagation process between input layer and the first hidden layer is defined as the Equation 3.9.

$$\boldsymbol{F}(\boldsymbol{cr}^{(1)}_{(a\to b)}) = \Delta(\omega_0 dr + b_0 + \sum_{m=1}^{R}(\omega_m d_m + b_m)), \tag{3.9}$$

where $\boldsymbol{F}$ is the set of neurons in the hidden layer next to the input layer; The $\Delta(.)$ is the activation function of the input layer, and the details regarding activation function can be found in the book [59]; ω_m and ω_0 represent the weights associated with each input feature. b_m and b_0 represent the bias associated with each neuron in the hidden layer. Figure 3.7 illustrates the structure of the MLP-based trajectory similarity measurement model. The input layer obtains the comparison feature, and each neuron unit in the input layer will adopt one element in the trajectory comparison feature. Then, the information collected by the input layer is propagated to the next layer. The Equation 3.9 shows the propagation between the input layer and the first hidden layer. Then, the results processed by the first hidden layer are propagated to the following hidden layers. Finally, the output layer employs the sigmoid function to generate the similarity, which is defined as the probability of two trajectories having the same destination. The similarity $y^{(1)}_{(a\to b)}$ is defined as follow.

$$y^{(1)}_{(a\to b)} = \delta(z) = \frac{1}{1+e^{-z}}. \tag{3.10}$$

where $y^{(1)}_{(a\to b)} \in [0,1]$ is the similarity between sampled traveling vessel trajectory $\boldsymbol{s}_{(a)}$ and the historical trajectory $\boldsymbol{t}^{(1)}_{(a\to b)}$; z is the output of the last hidden layer. $\delta(.)$ represents the sigmoid function.

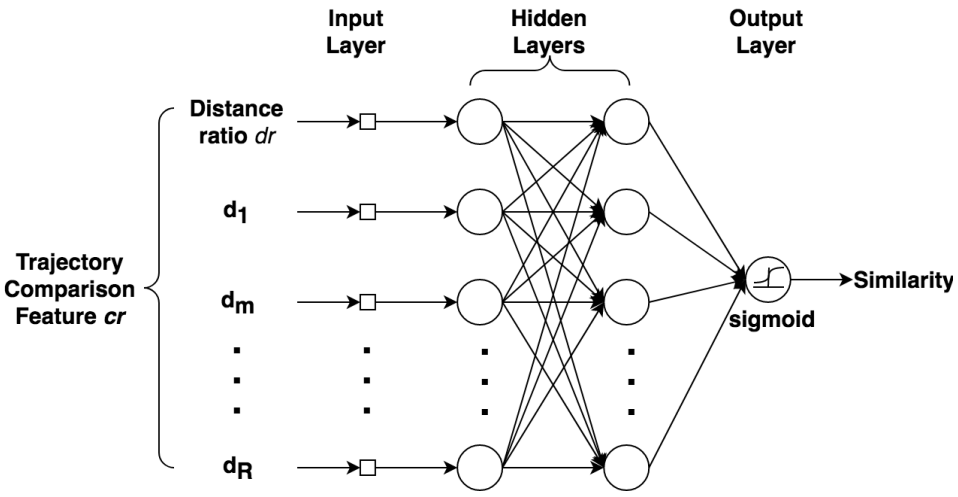

Figure 3.7. The structure of **MLP** in trajectory similarity measurement.

3.6.4 Random Forest-based Similarity Measurement Method

Random Forest (**RF**) is an ensemble learning method combining decision trees into one average prediction. The training algorithm for **RF** employs the sampling method - bagging, which is based on bootstrap sampling for the input features. Bagging repeatedly selects random samples with replacement (bootstrapping) of the training sets and fits trees to these selected samples [60, 61]. Given a set of comparison feature $\boldsymbol{cr}^{(1)}_{(a\to b)} = \{d_1, ..., d_R, dr\}$ in $\boldsymbol{C}_{(a)}$ between sampled traveling vessel trajectory $\boldsymbol{s}_{(a)}$ and the historical trajectory $\boldsymbol{t}^{(1)}_{(a\to b)}$, the structure of Random Forest (**RF**) in trajectory similarity measurement is shown in Fig. 3.8. The mathematical equation of similarity measurement is defined as follows:

$$y^{(1)}_{(a\to b)} = \sum_{j=1}^{p} \beta_j f_j(h_j(\boldsymbol{cr}^{(1)}_{(a\to b)})). \tag{3.11}$$

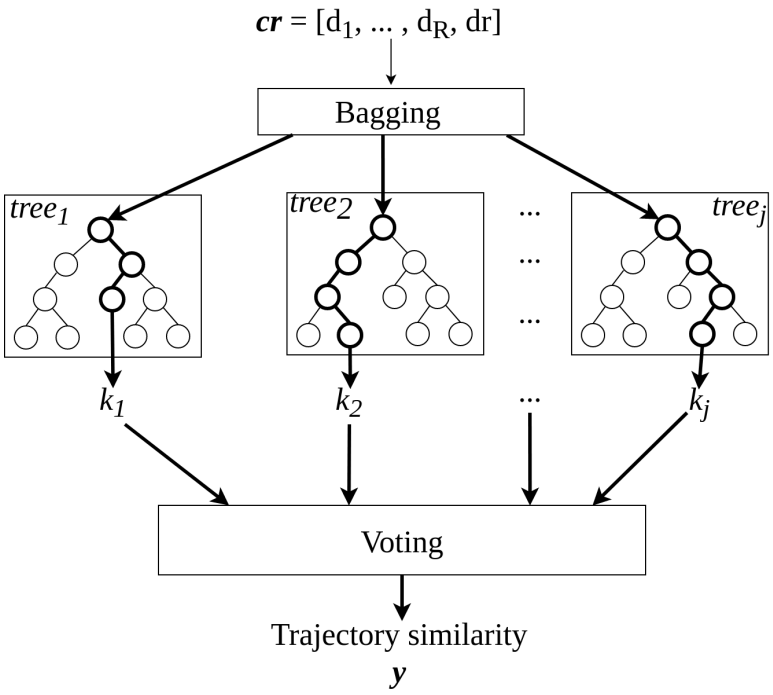

Figure 3.8. The structure of Random Forest (**RF**) in trajectory similarity measurement.

where $y^{(1)}_{(a\to b)} \in [0,1]$ is the similarity between sampled traveling vessel trajectory $\boldsymbol{s}_{(a)}$ and the historical trajectory $\boldsymbol{t}^{(1)}_{(a\to b)}$ with comparison feature $\boldsymbol{cr}^{(1)}_{(a\to b)}$; p indicates the total number of decision trees, and the β_j represents the weight assigned to the j-th decision tree ($\sum_{j=1}^{p}\beta_j = 1$); $f_j(.)$ is the function of the j-th decision tree and $h_j(.)$ is the representation of the bagging (bootstrap sampling) made on the given set of features. More details of random forest can be found in research [60]. For each iteration $j \in [1,p]$, a subset $h_j(\boldsymbol{cr}^{(1)}_{(a\to b)})$ is generated from $\boldsymbol{cr}^{(1)}_{(a\to b)}$ by bootstrap sampling. Then, the decision tree $f_j(.)$ fits the subset $h_j(\boldsymbol{cr}^{(1)}_{(a\to b)})$ with the weight β_j. Finally, through combining all probability estimates from $f_j(.)$ with assigned weight β_j, the final class probability estimate is generated and regraded as the similarity $y^{(1)}_{(a\to b)}$. Therefore, for all the historical trajectories from port a, there is a set of similarities $\boldsymbol{y}_{(a)} = \{y^{(1)}_{(a\to b)}, ..., y^{(P)}_{(a\to \psi)}\} \in \mathbb{R}^P$.

3.7 Decision Strategies for Vessel Destination Prediction

Two decision strategies are proposed for predicting the destination. One is the Maximum Similarity-based Decision strategy (**MSD**), where the destination port of the most similar historical trajectory is predicted as the destination of the traveling vessel.

To further enhance similarity scores of destinations, the other decision strategy - Port Frequency-based Decision strategy (**PFD**) is also proposed. The destination frequency is introduced to normalize the similarity scores. For example, the frequency $freq_{(a\to b)}$ that a vessel travels from port a to another port b is defined as below:

$$freq_{(a\to b)} = \frac{N(\boldsymbol{T}_{(a\to b)})}{P}, \tag{3.12}$$

where $N(\boldsymbol{T}_{(a\to b)})$ counts the number of trajectories from port a to b and P is the total number of trajectories starting from port a where $\boldsymbol{T}_{(a)} \in \mathbb{R}^{P\times K}$.

For a vessel traveling from port a, the proposed method will calculate the similarities $\boldsymbol{y}_{(a)}$ between its traveling trajectory and all historical trajecotories from port a. The frequencies $\boldsymbol{freq}_{(a)}$ of all the destinations from port a are also calculated. Besides, through tuning the parameters, it is found that when limiting the port frequency between 0.1 and 0.55, the decision strategy's performance is the best. The similarities will then be normalized by the frequency of the correlated port. One example is given for illustrating the process described above. The similarity $y^{(1)}_{(a\to b)}$ between $\boldsymbol{s}_{(a)}$ and $\boldsymbol{t}^{(1)}_{(a\to b)}$ is normalized as follows.

$$norm(freq_{(a\to b)}) = \frac{freq_{(a\to b)} - \min(\boldsymbol{freq}_{(a)})}{\max(\boldsymbol{freq}_{(a)}) - \min(\boldsymbol{freq}_{(a)})}, \tag{3.13}$$

$$\zeta^{(1)}_{(a\to b)} = \max(\min(norm(freq_{(a\to b)}), 0.55), 0.1) \times y^{(1)}_{(a\to b)}. \tag{3.14}$$

where $\zeta^{(1)}_{(a\to b)} \in [0, 0.55]$ is the normalized similarity for $y^{(1)}_{(a\to b)}$. For all historical trajectories from port a, there is a set of normalized similarities $\boldsymbol{\zeta}_{(a)} = \{\zeta^{(1)}_{(a\to b)}, ..., \zeta^{(P)}_{(a\to \psi)}\} \in \mathbb{R}^P$ based on the set of similarities $\boldsymbol{y}_{(a)}$. Afterward, the predicted destination features the highest value in $\boldsymbol{\zeta}_{(a)}$, which means it has the highest probability to be the destination.

3.8 Evaluation Metrics

For this particular study on vessel destination prediction, this thesis adopted the established metrics in [32, 33], where Port Accuracy (**PortACC**) is employed to evaluate the performance of the prediction. Besides, City Accuracy (**CityACC**) is defined in Equation 3.16 to present the city-based accuracy. Actually, the **PortACC** and City Accuracy (**CityACC**) are derived from the "top-1 accuracy", which is a well-established evaluation metric for judging the multiclass classifier's performance. Top-1 accuracy gives a rate that the output with the highest probability matches ground truth [62]. The "top-1 accuracy" is a widely used evaluation for the multiclass classification tasks. For instance, the performance of the classifier on the ImageNet [62] are exclusively evaluated by the "top-1" and "top-5" accuracy [63]. As proposed by [62], the images in the ImageNet with different categories of objects are labeled as only one object category. Hence, the "top-5 accuracy" is used for the evaluation to balance the ambiguities brought by inaccurate labeling. However, for the case of the vessel's destination, the vessel can not arrive at two destination ports at the same time. In this way, there is no ambiguity brought by inaccurate labeling. For a more accurate estimate, this study refers to the "top-1 accuracy", and develops two metrics Port Accuracy (**PortACC**) and City Accuracy (**CityACC**) for evaluating the model performance in destination prediction. As for the **PortACC**, this evaluation metric is derived from "top-1 accuracy" and is defined as the percentage of the time that the classifier's highest-confidence destination prediction matches the real destination on the level of the port. Given the number of the predictions that predicted destination port is correct amt^r, the amount of the traveling vessel trajectories that predicted by the framework amt, the **PortACC** is:

$$PortACC = \frac{amt^r}{amt}. \tag{3.15}$$

The **CityACC** is used to judge whether the prediction destination port for the traveling vessel and the real destination port are in the same city. Given the amount of the predictions that predicted destination ports are in the same city with the real destination ports amt^{rc}, the amount of the traveling vessel trajectories that predicted

by the framework *amt*, the **CityACC** is:

$$CityACC = \frac{amt^{rc}}{amt}. \tag{3.16}$$

In addition, Average Prediction Distance Error (**APDE**) (see Equation 3.17) is defined to presents the average distance between the wrongly predicted and targeted ports. This metric resembles the "mean absolute error," and defined as the average haversine distance between the incorrectly predicted destinations and the real destinations. The unit for **APDE** is kilometer(km). Given the wrongly predicted destination ports list ppl^w, the real destination ports list rpl, the amount of the predictions that predicted destination ports are different from the real destination ports amt^w, the **APDE** is:

$$APDE = \frac{\sum_{n=1}^{amt^r} Haversine(ppl^w[n], rpl[n])}{amt^w}. \tag{3.17}$$

where $Haversine(.)$ is the function to calculate the distance (km) between two given coordinates.

3.9 Summary

This chapter describes the proposed general vessel destination prediction method. First, the **DBSCAN**-based trajectory segmentation algorithm is proposed to convert one vessel's AIS records to usable formats - traveling trajectory, departure port, and historical trajectories. The traveling trajectory is then preprocessed by sampling. The correlated historical trajectories, which have the same destination as the traveling trajectory, are queried. Afterward, comparison features between the traveling trajectory and history trajectories are extracted and fed into the proposed **ML**-based model for similarities measurement. These measured similarities are then sent into the proposed decision strategy along with the port frequencies to predict different ports' possibilities of being the vessel's destination. Finally, the port with the highest possibility is predicted as the vessel destination.

Chapter 4

Experimental Results and Discussion

In this chapter, experiments are conducted to validate the proposed methods. Section 4.1 describes the **AIS** data used for the experiments, and the process of train and test dataset preparation. Section 4.2 illustrates the experiment of comparing the proposed methods against state-of-the-art methods on predicting destination with five-day trajectories. The method, which performs the best in Section 4.2, is then further investigated with being applied to cumulative trajectories in Section 4.3. Finally, discussions on the performance and feasibility of proposed methods are presented in Section 4.4.

4.1 Data Description

To validate the proposed method, we used the dataset containing 5,928,471 historical trajectories between 10,618 ports during the period from the year 2011 to 2017 in the experiments. This dataset is generated from 141,892,144 AIS records. As illustrated in Fig. 4.1, the historical trajectories from 10,618 ports were separated into two groups, which contain historical trajectories of 1,125 and 9,493 ports, respectively. The 1,125 and 9,493 ports are separated on the basis that there is no historical trajectory between two groups of ports. In other words, in the history, there are no route between port in the “1,125 ports” and port in the “9,493 ports”.

The 8,210 trajectories were randomly generated from the historical trajectories of 1,125 ports for training while the 18,409 trajectories were created from the 1,125 ports as the testing set one. The testing set two, including 17,528 trajectories, was sampled from the other 9,493 ports. The training set one and two are merged and then employed for testing. In general, there are 35,937 trajectories in total for testing. Table 4.1 shows the distribution of the training and testing data based on

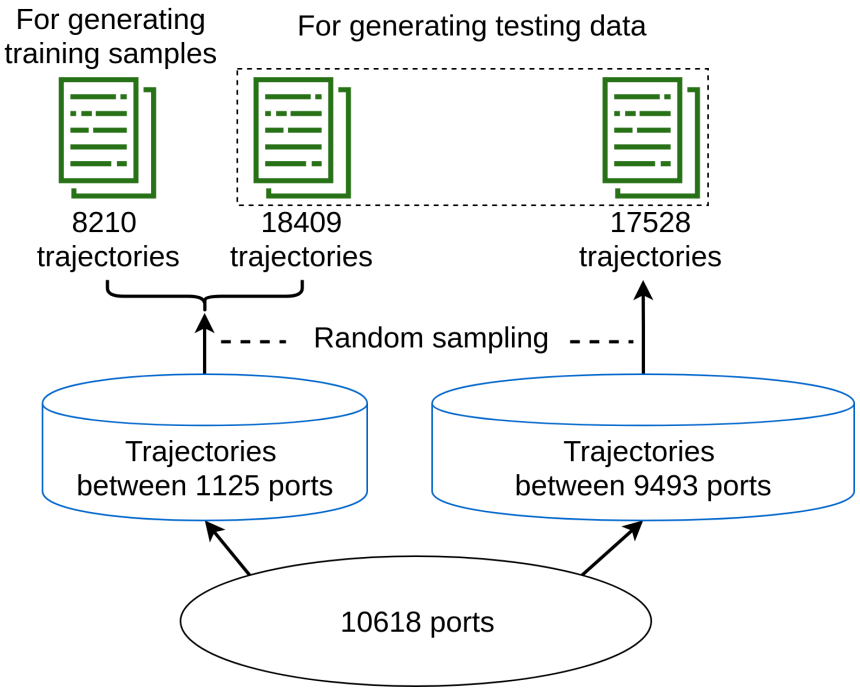

Figure 4.1. Representation of preparing the training and testing data.

traveling time. Table 4.2 gives an example of the historical trajectory between two ports in twelve days with the timestamps and coordinates.

Two experiments were carried out to validate the feasibility and validity of the proposed prediction method.

Table 4.1. Traveling time distribution of trajectories for constructing training and testing data.

Traveling time d[days]	**Training dataset**	**Testing dataset**
$5 < d < 10$	5836	26040
$10 < d < 15$	1329	5527
$15 < d < 20$	453	2001
$20 < d < 25$	228	885
$25 < d < 30$	125	470
$30 < d < 35$	61	279
$35 < d$	178	735

Table 4.2. An example of the historical trajectory with twelve-day traveling.

	Coordinate	
Timestamp	**Latitude**	**Longitude**
2017-06-12T15:43:24Z	21.4811116667	111.078446667
2017-06-12T22:09:43Z	21.4708033333	111.087156667
...	...	...
2017-06-18T04:13:30Z	22.0482116667	113.655393333
2017-06-18T04:46:20Z	22.09689	113.65538
2017-06-18T05:02:11Z	22.097275	113.6545
...	...	...
2017-06-23T23:58:32Z	22.7369016667	113.66117
2017-06-24T01:04:22Z	22.8277566667	113.55225

4.2 Destination Prediction with Five-day Trajectories

In this experiment, the four proposed **ML**-based similarity measurement method, together with the eight state-of-the-art approaches in combination with the two proposed decision strategies, were used to predict the destination ports with five-day trajectories. The performances were compared. Figure 4.2 illustrates the process of generating the five-day trajectories from the historical trajectories. Eight state-of-the-art similarity measurement methods include:

- Hausdorff Distance [45]

- Fréchet Distance[48]
- Discrete Fréchet [49]
- Symmetrized segment-path distance (**SSPD**) [50]
- Dynamic time warping (**DTW**) [64]
- Longest Common Subsequence (**LCSS**) [65]
- Edit distance with real penalty (**ERP**) [66]
- Edit distance on real sequence (**EDR**) [67]

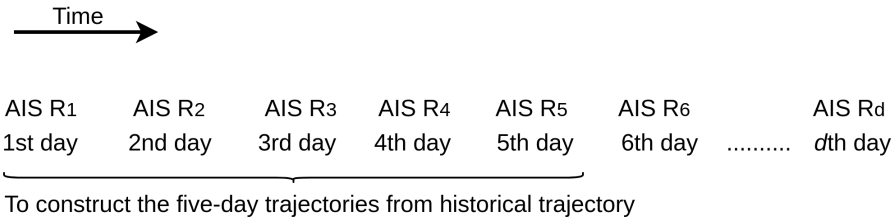

Figure 4.2. Process of generating five-day trajectories from historical trajectories.

The implementation of these eight state-of-the-art similarity measurement methods are decribed in details in Section 2.2.2. In addition, the four proposed **ML**-based similarity measurement methods (described in Section 3.6) employed in the experiments to predict vessel's destination are:

- **Naive Bayes classifier**-based similarity measurement method
- **MLP**-based similarity measurement method
- **IndRNN**-based similarity measurement method
- **RF**-based similarity measurement method

The two decision strategies as described in Section 3.7 were used together with the above twelve methods.

Table 4.3. Experimental results of eight state-of-the-art and four proposed **ML**-based methods on five-day trajectories.

Category	**Similarity measurement method combined with decision strategy**	**APDE[km]**	**PortACC[%]**	**CityACC[%]**
State-of-the-art similarity measurement methods	Hausdorff & MSD	824.45	43.13	65.35
	Hausdorff & PFD	1313.39	29.23	46.11
	Fréchet & MSD	771.52	39.25	60.92
	Fréchet & PFD	1313.95	29.24	46.14
	Discrete Fréchet & MSD	759.91	39.46	61.07
	Discrete Fréchet & PFD	1313.73	29.24	46.12
	SSPD & MSD	803.78	54.70	72.45
	SSPD & PFD	1313.39	29.23	46.12
	DTW & MSD	815.64	48.96	70.26
	DTW & PFD	1311.58	29.29	46.15
	LCSS & MSD	1115.94	36.01	56.13
	LCSS &PFD	1328.37	29.04	45.70
	ERP & MSD	803.01	36.23	58.48
	ERP & PFD	1311.73	29.27	46.21
	EDR & MSD	2346.90	17.77	34.14
	EDR & PFD	1322.15	29.16	46.08
Proposed ML[a] based similarity measurement methods	Naive Bayes-based & MSD	1643.53	24.43	44.47
	Naive Bayes-based & PFD	893.28	43.33	63.48
	IndRNN-based & MSD	783.88	39.41	46.08
	IndRNN-based & PFD	694.04	53.95	59.51
	MLP-based & MSD	764.42	43.89	66.71
	MLP-based & PFD	691.61	59.21	78.26
	RF-based & MSD	723.95	58.36	76.39
	RF-based & PFD	**660.65**	**66.57**	**81.65**

[a] ML is abbreviation for Machine Learning

Table 4.3 presents the experimental results of the total twelve similarity measurement methods combined with two decision strategies on the 35,937 five-day trajectories. A lower **APDE** together with higher **PortACC** and **CityACC** values indicate a better destination prediction performance. From Table 4.3, we have the following observations:

- The **APDE** of the **RF**-based similarity measurement method combined with **PFD** is 660.65 *km*, which is about 30*km* more accurate than the next-to-the-best method. This method also achieves the best performance in terms of both the **PortACC** and **CityACC** metrics.

- Generally, the **PFD** decision strategy outperforms **MSD** for the **ML**-based trajectory similarity measurement methods but is inferior to **MSD** for the conventional similarity measurement methods. **MSD** takes only the trajectory that is most similar to the traveling vessel trajectory into account for decision-making while **PFD** calculates the average of the similarity scores for the historical trajectories linking to the same destination port. Thus, **PFD** is not sensitive to an individual sample. **ML**-based similarity measurement methods, which are trained with large volumes of data, will assign high similarity scores to historical trajectories that have the same destination with the traveling vessel trajectory. In this way, **ML**-based similarity measurement achieves a more accurate and robust performance with **PFD** than with **MSD**. However, conventional similarity measurement methods can only measure the similarity between two trajectories based on their features, e.g., shape and length, etc. The vessel trajectories with the same departure and destination can differ significantly in shape and length. Thus, some historical trajectories that have the same departure and destination as the traveling trajectory receive relatively low similarity scores. The results with **PFD** will be degraded by these similarity scores. The conventional similarity measurement methods achieved better results with **MSD**.

- The efficiency of the **RF**-based approach is much higher than the **MLP**-based approach. As mentioned in Section 4.1, the similarity measurement models are trained by a massive amount of training samples. The **RF**-

based approach shows that it is more efficient in the training process than the **DL**-based approaches, which also have high requirements on the computation resources. Given the same computation resource for testing/application, the **RF**-based approach is much more computationally efficient than the **DL**-based approaches and has similar efficiency with the state-of-the-art approaches. Hence, the proposed **RF**-based approach is feasible for being applied to real-world prediction tasks.

The **RF**-based method combined with **PFD** achieved the best results. In the experiment, there were sufficient training samples to build the prediction model. To better understand the model's accuracy and generalization, extensive samples were used in the testing. A machine learning model may suffer from overfitting when the model is excessively complex, and the training data are incomprehensive or dirty. Overfitting is defined as the production of an analysis that corresponds too closely or exactly to a particular set of data, and may, therefore, fail to fit additional data or predict future observations reliably. According to research [60], random forests do not overfit as more trees are added, but produce a limiting value of the generalization error. However, there is a chance for the proposed random forest-based model to be overfitted with incomprehensive training data.

To identify the possible overfitting, Table 4.4 shows the results with the **RF**-based model for the 18,409 and 17,528 testing trajectories, respectively. These two testing sets were created from two-port sets separately, as illustrated in Fig. 4.1. The **PortACC** and **CityACC** from the testing with 18,409 trajectories are about 2% higher than the results of the 17,528 testing trajectories. The **APDE** result of the 18,409 testing trajectories is about 6 *km* larger than on the result from the 17,528 trajectories. There is no significant difference between the model's performances with the testing trajectory samples from different databases. Hence, the **RF**-based model is not overfitted.

In order to investigate the overfitting issue of the **RF**-based approach, the performances of **RF**-based & **PFD** on the 18,409 trajectories (randomly sampled from the same trajectories dataset with the training trajectories), and 17,528 trajectories (randomly sampled from a separate trajectory database) are evaluated separately. The more details of generating these "18,409" and "17,528" trajectories are re-

flected in Fig. 4.1. The overfitting is defined as the production of an analysis that corresponds too closely or exactly to a particular set of data, and may, therefore, fail to fit additional data or predict future observations reliably. As shown in the Table 4.4, the **PortACC** and **CityACC** of the model on the trajectories sampled from the same database with 8210 training trajectories, are around 2% higher than on the trajectories from the separate database. The **APDE** of the model on the trajectories from the separate database, is around 6 *km* lower than on the trajectories that are sampled from the same database with 8210 training trajectories. This shows that there is no significant difference between the model's performance trajectories sampled from different databases. Hence, it has been proven that the model is not overfitted.

Table 4.4. Experimental results for judging whether the model is overfitted, and validating model's generalization.

Similarity measurement method combined with decision strategy	Testing trajectoires	APDE[km]	PortACC[%]	CityACC[%]
RF-based & PFD	18409 trajectories[a]	663.47	67.65	82.41
RF-based & PFD	17528 trajectories[b]	657.89	65.43	80.87

[a] As shown in Fig. 4.1, these 18409 trajectories are randomly sampled from the same database with 8210 training trajectories.
[b] As shown in Fig. 4.1, these 17528 trajectories are randomly sampled from a separate database.

As presented in Table 4.4, the proposed model can still achieve a high destination prediction accuracy for the trajectories between new ports that have never been trained on. This phenomenon reflects the generalization of the proposed **RF**-based vessel destination prediction model. When predicting the trajectories between new ports, as long as this situation is covered in the historical trajectories, the proposed model can accurately predict its destination.

4.3 Destination Prediction with Cumulative Trajectories

In this experiment, the best method identified from the first experiment, i.e., **RF**-based similarity measurement with the **PFD** method, was further applied to cumulative trajectories. The procedure to generate the cumulative trajectories from historical testing trajectories are represented in Fig. 4.3. For each full testing trajectory, one-day, two-day, three-day, four-day, and five-day trajectories are generated.

Figure 4.4 presents the experimental results of **RF**-based similarity measurement with the **PFD** method on the cumulative trajectories. The "travel days" in Fig. 4.4 refers to the traveling time. For example, the evaluation result for "one travel day" is obtained by applying the combined model on the one-day trajectories.

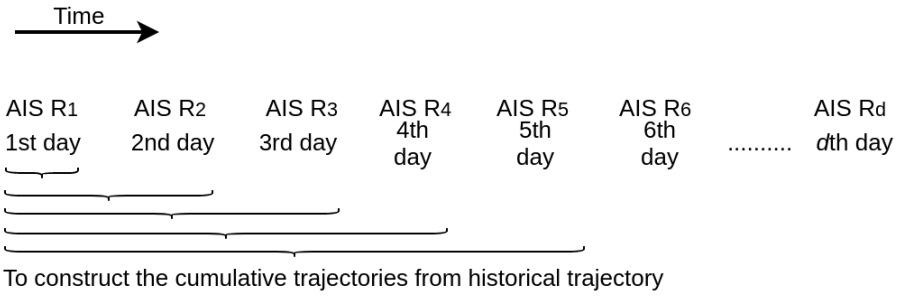

Figure 4.3. Process of generating cumulative trajectories from historical trajectories.

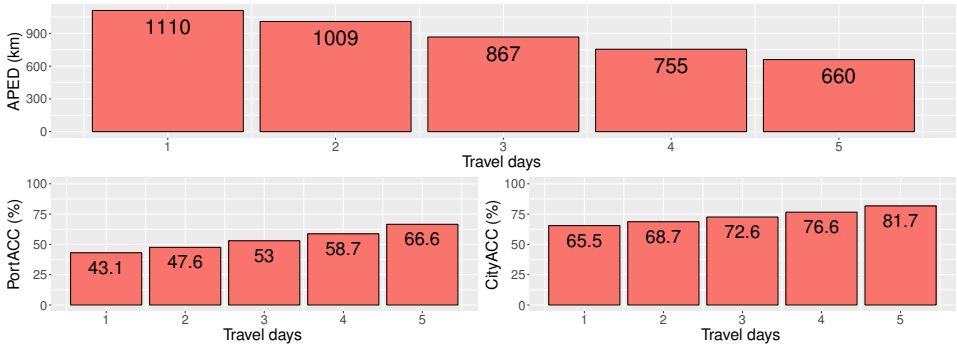

Figure 4.4. APED (top), PortACC (bottom-left) and CityACC (bottom-left) results of RF-based similarity measurement with the PFD method on cumulative trajectories.

As shown in Fig. 4.4, when the vessel moves forward during travel days, the model achieves a better prediction while more data are available. Additionally, it is observed that **CityACC** is about 15% higher than **PortACC** for the same "travel day." The main reason is that the differences between trajectories to different ports would be more obvious when vessels are getting closer to the city destinations.

4.4 Discussion

The experimental results demonstrated that the proposed **RF**-based similarity measurement with the **PFD** method achieved the best performance for vessel destination prediction. It has also been proven that the model is not overfitted, and can generally predict the vessel's destination port.

In the case that a vessel is following a new shipping route, the similarities between the traveling trajectory and the existing historical trajectories will be relatively low. The small similarity value does not support solid decision making and may lead to an incorrect conclusion. This is a limitation of the comparison-based method to deal with a new path that followed by the vessel for the first time. However, once the vessel arrives at the destination by following the new shipping route, the trajectory database will then be updated with this vessels trajectory. Afterward, for other vessels to follow this new shipping route, it is possible to carry out the trajectory comparison and make a reliable destination prediction. Through continuously updating the database with trajectories following the new path, the destination prediction for vessels in this new path will be more reliable and accurate. In the case that a vessel is traveling to a new port $port_{new}$ for the first time, the model will fail to make the correct prediction because this new port has not been covered in the historical trajectory database. However, when the vessel finishes its trip, the historical trajectories database will then be updated with its trajectory. Afterward, the model can give the correct destination prediction for the vessels that depart to $port_{new}$.

The proposed Random Forest (**RF**)-based model measures the similarity based on the distance ratio and perpendicular distances. Due to the consecutive nature between adjacent perpendicular distances, the adjacent perpendicular distances will be probably highly correlated with each other. The multicollinearity of those variables may impact the linear regression model. However, the multicollinearity does not affect the prediction performance of the random forest model, which employs feature selection during the training process [60, 61], e.g., a random subset of features is chosen for each tree. According to the research [68], the random feature selection through bootstrap sampling can reduce the estimation bias due to multicollinearity.

4.5 Summary

In this chapter, performances of twelve similarity measurement methods (four proposed **ML**-based and eight state-of-the-arts approaches) combining with two decision strategies are evaluated and compared. In order to validate the performance of the proposed model, two experiments are conducted for predicting destination with five-day trajectories and cumulative trajectories, respectively. According to the experimental results in Section 4.2, the proposed **RF**-based similarity measurement with **PFD**-based decision strategy achieves the best performance among all 24 combination methods. Further, when predicting with cumulative trajectory, the performance of the proposed **RF**-based similarity measurement with **PFD**-based decision strategy on vessel destination prediction keeps improving as the knowing records increasing. Moreover, the validity and feasibility of the proposed **RF**-based approach have also been discussed in this chapter.

Chapter 5

Conclusions

In this thesis, an **AIS** data-driven method was proposed to address general vessel destination prediction by using trajectory similarity. Two experiments were carried out to validate the feasibility of the proposed method. In the first experiment, eight state-of-the-art methods were compared with the four proposed **ML**-based approaches on 35,937 five-day trajectories. The experimental results showed that the proposed **RF**-based approach achieved the best results in terms of the three evaluation metrics. In the second experiment, the best method from the first experiment was applied to multi-day trajectories of 35,937 moving vessels, separately. The experimental results demonstrated that with more collected records, the prediction performance was improved. The results presented in this thesis demonstrated the feasibility of using Random Forest (**RF**) to predict the general vessel destination by comparing the trajectories' similarity. This thesis shows that the trajectory similarity-based approach provides a promising way for vessel destination prediction, which is general, accurate, and updateable. However, there are also some limitations in these studies. The challenges and the potential future studies for the research in this thesis are discussed and listed below.

- In order to handle different traveling patterns of the vessels, the proposed method is currently comparing the traveling trajectory with all historical trajectories from the same port. Although the greedy algorithm-based method shows promising accuracy in predicting vessel destinations, the method's efficiency can be further improved. Many vessels could have repetitive pat-

terns in their voyages, e.g., passenger vessels visiting the same ports every day. The vessels follow weather routing-based paths to reduce fuel consumption, to avoid adding extra pressure to the vessel's engines, to have a safer path to their destination or reach faster to their destination. Weather routing depends on the currents and the waves, and these tend to have seasonal patterns. Further research on identifying duplicated traveling patterns and removing them from the compared trajectories will be carried out for the optimization of computational efficiency.

- The extra data sources regarding port attributes, vessel classes, and seasonality can be considered in the decision strategy for a more reliable prediction result. Vessels generally tend to follow paths based on their vessel class and the period of the year. For instance, tanker vessels follow different routes compared to cargo vessels due to their drought but also due to the fact that not all ports can be visited from all vessel classes. When predicting the destination of the tanker vessel that loaded with oil, the preference will be given to candidate ports which are frequently visited by tanker. Besides, the **AIS** data is now challenged by the missing data issue. In future work, the **ML**-based approaches can be applied for filling missing data. With more **AIS** data being available, the accuracy of vessel destination prediction can be further improved.

Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015. → pages ix, 14, 31, 32

[2] Y. Kisialiou, I. Gribkovskaia, and GilbertLaporte, "Robust supply vessel routing and scheduling," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 366–378, May 2018. → page 1

[3] M. K. Msakni and M. Haouari, "Short-term planning of liquefied natural gas deliveries," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 393–410, May 2018. → page 1

[4] J.-P. Rodrigue, *The Geography of Transport Systems, 4th ed.* New York: Routledge, Nov. 2017. → page 1

[5] G. United Nations, *United Nations Conference on Trade and Development*. United Nations, 2017. → page 1

[6] A. Alessandrini, F. Mazzarella, and M. Vespe, "Estimated time of arrival using historical vessel tracking data," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2018. → pages 1, 6

[7] P. Sheng and J. Yin, "Extracting shipping route patterns by trajectory clustering model based on automatic identification system data," *Sustainability*, vol. 10, no. 7, p. 2327, Jul. 2018. → page 2

[8] AIS Hub-AIS Data Exchange, "Ais coverage map," http://www.aishub.net/coverage, 2019, accessed: 2019-08-04.

[9] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G.-B. Huang, "Exploiting ais data for intelligent maritime navigation: A comprehensive survey from data to methodology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1559–1582, May 2018. → page 2

[10] H. Greidanus, M. Alvarez, T. Eriksen, and V. Gammieri, "Completeness and accuracy of a wide-area maritime situational picture based on automatic ship reporting systems," *Journal of Navigation*, vol. 69, no. 01, pp. 156–168, Aug. 2015. → page 2

[11] T. Mestl and K. Dausendschn, "Port eta prediction based on ais data," in *15th International Conference on Computer and IT Applications in the Maritime Industries COMPIT16*, May 2016, pp. 198–201. → page 2

[12] A. Sharm, Z. Zheng, and A. Bhaskar, "A pattern recognition algorithm for assessing trajectory completeness," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 432–457, Nov. 2018. → page 2

[13] World Port Source, "World port list," http://www.worldportsource.com/countries.php, 2019, accessed: 2019-08-04. → pages 2, 7

[14] K. Kepaptsoglou, G. Fountas, and M. G. Karlaftis, "Weather impact on containership routing in closed seas: A chance-constraint optimization approach," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 139–155, June 2015. → page 2

[15] C. G. S. Lokukaluge P. Perera, "Weather routing and safe ship handling in the future of shipping," *Ocean Engineering*, vol. 130, pp. 684–695, Sep. 2017. → page 2

[16] L. Chen, H. Hopman, and R. R.Negenborn, "Distributed model predictive control for vessel train formations of cooperative multi-vessel systems," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 101–118, July 2018. → page 2

[17] P. Wilson, "Accurate prediction of maritime trajectories from historical ais data using grid-based methods," Master's thesis, McMaster University, 2017. → pages 2, 6, 7

[18] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, no. 12, pp. 2218–2245, Jun. 2013. → pages 2, 6

[19] K.-I. Kim and K. M. Lee, "Context-aware information provisioning for vessel traffic service using rule-based and deep learning techniques," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 1, pp. 13–19, Mar. 2018. → pages 2, 7

[20] A. Daranda, "Neural network approach to predict marine traffic," Vilnius University Institute of Mathematics and Informatics, Tech. Rep., 2016. → page 6

[21] C.-X. Lin, T.-W. Huang, G. Guo, and M. D. F. Wong, "Mtdetector: A high-performance marine traffic detector at stream scale," in *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems - DEBS18*, 2018, pp. 205–208. → pages 2, 7

[22] S. Giannis, Z. Dimitrios, and C. Konstantinos, "A big data driven approach to extracting global trade patterns," in *Mobility Analytics for Spatio-Temporal and Social Data*, 2018, pp. 109–121. → page 5

[23] K. Chatzikokolakis, D. Zissis, G. Spiliopoulos, and K. Tserpes, "Mining vessel trajectory data for patterns of search and rescue," in *EDBT/ICDT Workshops*, 2018. → page 6

[24] P. Kostas, A. Elias, A. Alexander, V. Marios, P. Nikos, and T. Yannis, "Online event recognition from moving vessel trajectories," *GeoInformatica*, vol. 21, no. 2, pp. 389–427, 2017. → page 6

[25] P. Manolis, A. Alexander, D. Richard, R. Cyril, C. Elena, and J. Anne-Laure, "Composite event recognition for maritime monitoring," in *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems*, 2019, pp. 163–174. → page 6

[26] H. Ljunggren, "Exploring the capabilities of deep learning in sea surveillance Using deep learning to classify motion trajectories from AIS data," Ph.D. dissertation, KTH Royal Institute of Technology, 2017. → page 6

[27] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines," *Artificial Intelligence*, vol. 171, no. 5-6, pp. 311–331, Apr. 2007. → page 6

[28] A. Dobrkovic, M.-E. Iacob, J. van Hillegersberg, M. R. K. Mes, and M. Glandrup, "Towards an approach for long term AIS-based prediction of vessel arrival times," in *Logistics and Supply Chain Innovation*, Aug. 2015, pp. 281–294. → page 6

[29] W. M. Wijaya and Y. Nakamura, "Predicting ship behavior navigating through heavily trafficked fairways by analyzing AIS data on apache HBase," in *2013 First International Symposium on Computing and Networking*, Dec. 2013, pp. 220–226. → page 6

[30] Y. Wang, J. Zhang, X. Chen, X. Chu, and X. Yan, "A spatial-temporal forensic analysis for inland-water ship collisions using ais data," *Safety Science*, vol. 57, pp. 187–202, Aug. 2013.

[31] S. Hornauer and A. Hahn, "Towards marine collision avoidance based on automatic route exchange," *IFAC Proceedings Volumes*, vol. 46, no. 33, pp. 103–107, 2013. → page 6

[32] A. Ciprian, D. Paul, O. Emanuel, and R. Valentin, "Cell grid architecture for maritime route prediction on ais data streams," in *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, June 2018, pp. 202–204. → pages 7, 36

[33] B. Oleh, S. Florian, M. André, B. Andrey, and F. Christof, "Real-time destination and eta prediction for maritime traffic," in *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, June 2018, pp. 198–201. → pages 7, 36

[34] Z. Fu, Z. Tian, Y. Xu, and C. Qiao, "A two-step clustering approach to extract locations from individual GPS trajectory data," *ISPRS International Journal of Geo-Information*, vol. 5, no. 10, p. 166, Sep. 2016. → pages 8, 9

[35] L. Xiang, M. Gao, and T. Wu, "Extracting stops from noisy trajectories: A sequence oriented clustering approach," *ISPRS International Journal of Geo-Information*, vol. 5, no. 3, p. 29, Mar. 2016. → page 8

[36] M. Riyadh, N. Mustapha, N. Sulaiman, and N. B. M. Sharef, "ONF-TRS: On-line noise filtering algorithm for trajectory segmentation based on MDL threshold," *Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 42–48, Dec. 2016. → page 8

[37] J. Yin, Y.-W. Si, and Z. Gong, "Financial time series segmentation based on turning points," in *Proceedings 2011 International Conference on System Science and Engineering*, Jun. 2011, pp. 394–399. → pages 8, 9

[38] Z. Zhong, "Background noise distribution after high-resolution processing in ship-borne radar," in *Third International Conference on Information Technology and Applications (ICITA'05)*, vol. 14, no. 1, 2005, pp. 115–118. → page 9

[39] J.-G. Lee, J. Han, X. Li, and H. Gonzalez, "TraClass:Trajectory Classification Using Hierarchical Region-Based and Trajectory-Based Clustering," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1081–1094, Aug. 2008. → page 9

[40] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: A recommender system for finding passengers and vacant taxis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2390–2403, Oct. 2013. → page 9

[41] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "T-drive: Enhancing driving directions with taxi drivers intelligence," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 220–232, Jan. 2013. → page 9

[42] Y. Zheng, "Trajectory data mining," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, pp. 1–41, May 2015. → page 9

[43] R. S. P. Gaonkar, M. Xie, and H.-Z. Huang, "Optimizing maritime travel time reliability," *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 227, no. 2, pp. 167–176, Nov. 2012. → page 9

[44] P. Ranacher and K. Tzavella, "How to compare movement? a review of physical movement similarity measures in geographic information science and beyond," *Cartography and Geographic Information Science*, vol. 41, no. 3, pp. 286–307, Mar. 2014. → page 9

[45] N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy, "Review on trajectory similarity measures," in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, Dec. 2015, pp. 613–619. → pages 9, 10, 40

[46] N. Magdy, M. A. Sakr, and K. El-Bahnasy, "A generic trajectory similarity operator in moving object databases," *Egyptian Informatics Journal*, vol. 18, no. 1, pp. 29–37, Mar. 2017. → pages 9, 10

[47] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *Proceedings of the 1994 ACM SIGMOD international conference on Management of data - SIGMOD'94*, 1994, pp. 419–429. → page 9

[48] Efrat, Guibas, S. Har-Peled, Mitchell, and Murali, "New similarity measures between polylines with applications to morphing and polygon sweeping," *Discrete & Computational Geometry*, vol. 28, no. 4, pp. 535–569, Nov. 2002. → pages 10, 41

[49] J. Gudmundsson and N. Valladares, "A GPU approach to subtrajectory clustering using the fréchet distance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 924–937, Apr. 2015. → pages 10, 41

[50] P. C. Besse, B. Guillouet, J.-M. Loubes, and F. Royer, "Review and perspective for distance-based clustering of vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3306–3317, Nov. 2016. → pages 10, 41

[51] T. Nakamura, K. Taki, H. Nomiya, K. Seki, and K. Uehara, "A shape-based similarity measure for time series data with ensemble learning," *Pattern Analysis and Applications*, vol. 16, no. 4, pp. 535–548, Jan. 2012. → page 12

[52] G. Cormode and S. Muthukrishnan, "The string edit distance matching problem with moves," *ACM Transactions on Algorithms*, vol. 3, no. 1, pp. 1–19, Feb. 2007. → page 12

[53] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006. → page 14

[54] L. Shuai, L. Wanqing, C. Chris, Z. Ce, and G. Yanbo, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Mar. 2018, pp. 5457–5466. → pages 14, 31

[55] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231. → page 23

[56] R. W. Sinnott, "Virtues of the haversine," *Sky and Telescope*, vol. 68, no. 2, pp. 158–159, Dec. 1984. → page 27

[57] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *2008 IEEE 24th International Conference on Data Engineering*, Apr. 2008, pp. 140–149. → page 27

[58] A. McCallum and N. Kamal, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1, 1998, pp. 41–48. → page 30

[59] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. → page 32

[60] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. → pages 33, 34, 44, 47

[61] G. Biau, "Analysis of a random forests model," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 1063–1095, Apr. 2012. → pages 33, 47

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. → page 36

[63] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 8697–8710. → page 36

[64] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, Feb. 2012. → page 41

[65] C. Gruber, T. Gruber, S. Krinninger, and B. Sick, "Online signature verification with support vector machines based on LCSS kernel functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 4, pp. 1088–1100, Aug. 2010. → page 41

[66] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings 2004 VLDB Conference*, 2004, pp. 792–803. → page 41

[67] L. Chen, M. T. Ozsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD'05*, 2005, pp. 491–502. → page 41

[68] P. Anita and V. den Poel Dirk, "Random multiclass classification: Generalizing random forests to random mnl and random nb," in *Database and Expert Systems Applications*, 2007, pp. 349–358. → page 47