IDENTIFYING FISHING ACTIVITIES FROM AIS DATA WITH CONDITIONAL RANDOM FIELDS

by

Baifan Hu

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
November 2016

To my parents

Table of Contents

List of Tables

List of Figures

Abstract

Fishing activity detection is important for fishery management to maintain abundant oceans. The rising demand for fish and advanced fishing technologies has led to overfishing, species endangerment and marine habitat destruction. Illegal, unreported and unregulated (IUU) is one example of an important ecological and economic issue that requires the understanding of fishing behavior of ships. Our proposed approach to detecting fishing activities uses Conditional Random Fields (CRF) on Automatic Identification System (AIS) data. We generate features from selected attributes and combine different features based on their relationships and dependencies. We present three experiments on trawlers and longliners respectively as well as comparisons between CRFs and methods such as Autoencoder and Hidden Markov Model (HMM) to demonstrate the stability and effectiveness of the CRF models. Furthermore, we develop a geo-visualization with interaction and animation of these AIS data and our experimental results.

List of Abbreviations Used

AE Autoencoders

AIS Automatic Identification System

BIO Beginning, Inside and Outside tag

COG Course Over Ground

CRF Conditional Random Field

DAG Directed Acyclic Graph

DGM Directed Graphical Model

DM The Data Mining approach

FAO The United Nations Food and Agriculture Organization

GRU Gated Recurrent Unit

HMM Hidden Markov Model

ILOBO Iterative Leave One Batch Out

IMO International Maritime Organization

ITU International Telecommunication Union

IUU Illegal, Unreported and Unregulated

L-BFGS Limited-memory Broyden-Fletcher-Goldfarb-Shanno

MCS Monitoring Control and Surveillance program

MEMM Maximum Entropy Markov model

MMSI Maritime Mobile Service Identity, the unique identification of a vessel

NER Name Entity Recognition

NLP Natural Language Processing

NPV Negative Predictive Value

PGM Probabilistic Graphical Model

POS Part Of Speech

PPV Positive Predictive Value

QGIS Quantum GIS is a geographic information system (GIS) application

RNN Recurrent Neural Network

ROT Rate Of Turn

SLOBO Stratified Leave One Batch Out

SOFIA the State of World Fisheries and Aquaculture

SOG Speed Over Ground

SVM Supported Vector Machine

UGM Undirected Graphical Model

VMS Vessel Monitoring Systems

Acknowledgements

First, I would like to express my greatest gratitude to my supervisor, Dr.Stan Matwin, for your patience, guidance, encouragement, advice and support throughout my time in my Master program. Thank you for taking me as your student and providing me precious opportunities to work on the projects. I am so proud to have a supervisor who offers brilliant ideas, invaluable suggestions and directions on my research and studies. Your patient guidance and insightful comments significantly contributed to this thesis.

My sincere gratitude also goes to my co-supervisor, Dr.Ronald Pelot, for your constant support, encouragement and constructive suggestions. Your great support and kind advice has greatly helped the accomplishment of the work in this thesis. Thank you for trusting me and always being patient with me.

I would like to thank my thesis committee members, Dr.Evangelos Milios and Dr.Qigang Gao, for serving on my thesis defence as well as their questions and insightful comments on my thesis.

I would also like to extend my gratitude to Dr.Erico N.de Souza, for his advice and comments that guided my work. He has also spent a lot time reviewing my paper, which is a great help to me.

Thanks to all my friends and labmates in the Big Data Institute: Xiang, Xuan, Ahmad, Behrouz, Habibeh, Lulu, Eman, David, Pedram, Parsa, Casey, Fateha, Bala for the discussions and lots of things I learned from you and for all the fun we have had during the past two years. And I would like to thank Dalhousie University for providing a nice environment to do research and study.

I am thankful to my family that always loves and cares about me. To my Mom and Dad, I am really proud to be your child. Thank you for encouraging and supporting me to pursue my dreams. And I am grateful to all my friends. Thank you for the love and companionship.

Chapter 1

Introduction

Substantial global population expansion and fast-growing economics drive an increasing demand for fish products. In addition, global climate change that tends to have an impact on agricultural production pushes people to depend more on fish for nutrient needs. Modern fishing technologies and advanced fishing equipment bring about an excessive and continuous increase in fishing effort. More and more fish are caught than can be reproduced naturally in the oceans. The oceans now are under great pressure from global overfishing which causes a dramatic decline in the fish population. Several major commercial fish species are endangered which disturbs the balance of the food chain and threatens the ocean ecosystem. This also affects millions of people who depend on fish products as their main source of protein and income. According to a report in 2014, the State of World Fisheries and Aquaculture (SOFIA) by The United Nations Food and Agriculture Organization (FAO), fish accounts for 20 percent of animal protein intake for more than 2.9 billion people, and 15 percent of animal protein intake for 4.3 billion people [18]. FAO estimated that fisheries and aquaculture support the livelihoods of 10 to 12 percent of the world's population. According to the 2014 report, more than 90 percent of global fisheries are overexploited. A study published in 2006 in the journal Science [62] extrapolated fisheries catch data into the future and predicted that if the current volume of fishing activities remain unchanged or even increasing, all the world's fish stocks would collapse by the year 2048.

Unfortunately, in many cases these fishing activities are Illegal, Unreported and Unregulated which are referred to as IUU fishing. IUU fishing takes place when fishing takes place in a restricted area without permission, violating conservation and management measures such as ignoring quotas or by-catch limits, catching protected species, fishing certain species without a license, and failing to report or misreporting catches. It is estimated that over 11-26 million tons of fish are caught by IUU fishing each year resulting in a $10-23 billion economic cost [10]. Governments and fishery

management organizations are taking actions to instill long-term sustainable fisheries. Nations work collaboratively on placing restrictions on their territorial waters together with fishery management organizations which take a series of measures to better manage and conserve endangered international fish species [6]. However, IUU fishing could be any size or type of fishery and could happen anywhere, which makes tracing it a challenge. Tracking and monitoring vessels aim to obtain reliable information about activities of vessels from vessel movement data. Thus, in order to make fishing activities more transparent, better fishing activity detection is urgently needed.

There are primarily two types of broadcasting systems for vessel tracking: Vessel Monitoring Systems (VMS) and Automatic Identification Systems (AIS). VMS are satellite based communication systems which are designed for commercial fishing. As an important part of Monitoring Control and Surveillance (MCS) program, VMS provide information on vessels at regular intervals to track and monitor the fishing activities. VMS use bidirectional communications which guarantees the transmission and reception of every signal. However, VMS has some limitations to deal with the task of fishing activity detection. First of all, the time intervals between VMS signals are typically from one hour to several hours which might result in missing important information during the intervals. Also, VMS is not open for anyone to access its signal and the cost of the equipment as well as the satellite connection are expensive. In addition, VMS is run by individual countries and is not global which make it hard to be a global solution for tracking fishing activities.

In 2000, International Maritime Organization (IMO) first introduced the Automatic Identification System (AIS) to enhance the security and safety of maritime navigation. IMO made a mandate that requires AIS to be fitted on board all ships over 300 gross tonnage or passenger ships [2]. AIS is standardized by the International Telecommunication Union (ITU). Ships equipped with AIS can automatically broadcast information, including unique identification (Maritime Mobile Service Identity, MMSI), position (Longitude and Latitude), speed over ground (SOG), course over ground (COG), time and further details of the vessel to nearby ships and coastal authorities [11]. AIS data are openly accessible and not encrypted. In 2008, satellite AIS technology was implemented, enabling the collection of massive and reliable information about vessels in global areas within seconds. It is estimated that over

400,000 ships worldwide are equipped with AIS devices [8]. AIS system produces at least 100 million records per day [7]. Currently there are approximately 40 satellites in orbit equipped with AIS receivers. In 2015, exactEarth Ltd. and Harris Corp. formed an alliance to place maritime ship payloads on 58 low-orbiting satellites and expected to have over 50 satellites with maritime payloads by the end of 2017 [5]. Through an increasing number of AIS equipped satellites, AIS will be able to provide continuous and real-time signals. Consequently, satellite AIS data could be used as an ideal source to monitor vessel movements and detect fishing activities around the world.

Fishing trawlers and longliners are two common types of commercial fishing vessels. Trawlers, also known as draggers, are fishing by operating one or more fishing trawl nets and pulling and trailing them under the surface of the water at a specified depth behind the trawler in order to haul and trap the fish. Today, trawling has become one major powerful and productive fishing technique in global areas and has been employed to capture a variety of fish around the world. The sizes of trawlers vary widely from small open boats to large factory vessels such as freezer trawlers. With sufficient power supply from engines, trawlers are able to fish in a majority of the distant waters. It is also possible for them to fish from shallow waters to deep water at a depth of 2km.

Longliners use a different fishing technique from trawlers. Longliners essentially employ one long fishing cord, known as the main line, with snoods which are short length of branch lines set at regular intervals attached with baited hooks. Different target fish determine the number of lures, the distance of the intervals at which the snoods are set as well as the length of snoods. A single longline can have hundreds or thousands of baited hooks attached. Longlines could be placed at the bottom by an anchor or float at the surface of the water. The length of longlines can vary from hundreds of meters to tens of kilometers depending on whether they are used in coastal fishing or large scale mechanized and specialized fishing.

Since the two types of vessels apply different techniques to catch fish, their behaviors and patterns in fishing activities are also different from each other.

1.1 Aim and Objectives

The overall research aim of this thesis is to detect fishing activities from historical sequential AIS data. To deal with this task, we use a Conditional Random Fields (CRF) model [29] that, to the best of our knowledge, is the first use of CRF on AIS data. To fit the model, we preprocess AIS data and do feature engineering to obtain a representation conducive for the use of CRF. To prove the stability of our model, we make evaluation experiments on AIS data from two different types of vessels and compare it with other models. To better perceive the patterns of fishing activities, we develop an interactive visualization. To meet the research aim, we set the following objectives:

1). Survey existing fishing activities detection methods and methods for similar tasks.

2). Investigate fishing behaviors and the nature of AIS data.

3). Propose a new CRF-based approach for the task of fishing activities detection.

4). Demonstrate the stability and effectiveness of the resulting models.

5). Visualize trajectories and fishing activities.

1.2 Research Methodology

To address our main goals in this thesis as mentioned above, we propose a fishing activity detection framework as shown in Fig. 1.1. The proposed approach includes four procedures: feature engineering, building CRF models as well as comparative models, model evaluation, and trajectory visualization. First, we select features from preprocessed data and generate features from combinations of different features to fit the CRF model. Then with the created features, we train the CRF model. We also build other models such as Autoencoder, HMM model and a data mining method as comparative models. Then we design three different experiments to evaluate our CRF models and compare the results of different models. Finally, we design an interactive visualization to explore the AIS data as well as our model prediction results.

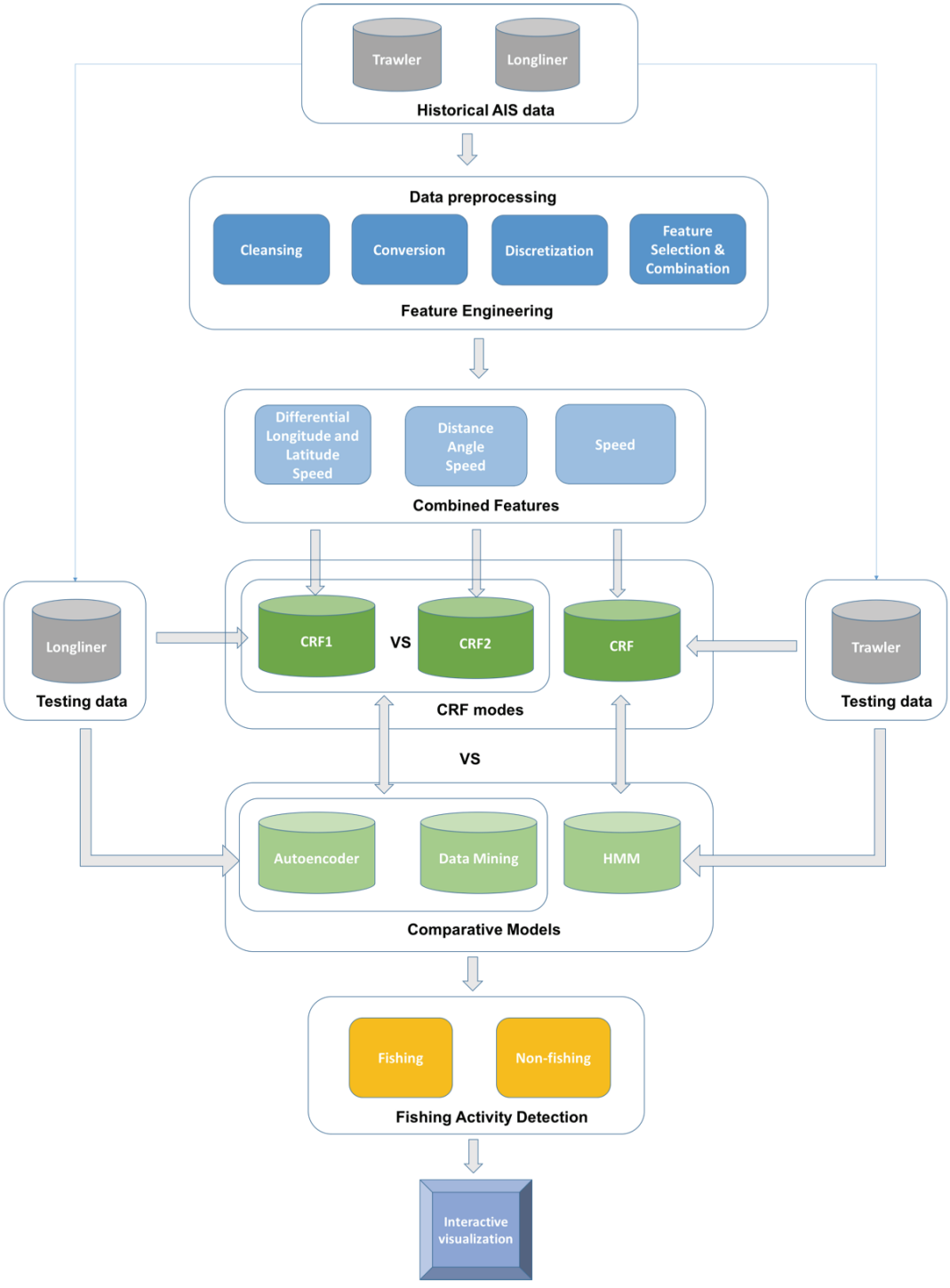


Figure 1.1: The framework of Fishing Activity Detection.

Feature construction creates features based on our analysis of the AIS data and the task. The AIS data we use in this thesis is labeled by an expert. The expert labels some points in a trajectory as representing fishing and some points as representing non-fishing. In general, the nature of AIS data gives us two challenges. One is the irregular granularity of the data with uneven time intervals from seconds to days. The other is the imbalance of the data with domination of one class over the other, where in our case the class ratio of fishing to non-fishing is 1:6 for trawlers and 3:1 for longliners. Among all the attributes including unique identification (MMSI), longitude and latitude, speed (SOG), course (COG), rate of turn (ROT) and so on, we find the speed attribute in AIS data has the closest relationship with fishing activities, especially in trawlers, but they are less correlated in longliners. We consider using longitude and latitude because they could provide information about shapes of trajectories. And we could not use absolute values of longitude and latitude because we want our model to work on global areas. As a result, we use differential longitudes and latitudes which provide approximate information about how direction and distance change from point to point. We then discretize the values of these features. Then we put together such features that their pairs of values have a spatial-temporal meaning, to account for their possible relations. We pair features together into grouped features which generates tuples. For instance, the selected feature A and feature B are paired, a and b are possible values of A and B, which are $a \in A$ and $b \in B$, then the generated tuple F is defined in the formulae:

$$F = A \times B = \{(a, b) | a \in A \text{ and } b \in B\} \tag{1.1}$$

We pair longitude and latitude in such a way that they together provide information about how location changes. We pair speed of consecutive points together because the variations of their speed depend on each other. Similarly for longitude and latitude, we pair them between neighboring points. We also make other pairs based on the relative position to a point. More detailed information is presented in Chapter 4.

We use the tool CRF++ to build the CRF model with our features. The features are converted to nominal values. This is because we regard our task of fishing activity detection similar to Part Of Speech (POS) tagging where speed of a point can be

regarded as a word and the corresponding label can be regarded as a respective tag of the word. Detailed information is presented in Chapter 3.

As for trawlers and longliners, we build different CRF models with different selected features based on their characteristics during fishing activities. From our exploration of AIS data from these two types of vessels, we regard that both geoinformation and speed are useful features in longliner fishing activities while geoinformation seems irrelevant to trawler fishing activity. Thus, we use both location features and speed to build models for longliners but only speed for trawlers. More information is presented in Chapter 4.

As for the comparative experiments, we reproduce and refer to other work as comparisons with our CRF models for trawlers and longliners respectively. We compare CRF models with HMM for trawlers and with autoencoder and data mining approach for longliners. In each comparison, we use the same data source to build CRF models and comparison models but with different selected features based on the methods. In the data mining approach and the autoencoder, only location features are used because these two methods detect fishing activities by making use of the shape of trajectories. In HMM experiments, we use only speed as a single observation corresponding to one state. More details are presented in Chapter 4.

As for visualization, we aim to visualize in a way to better display information of fishing and non-fishing patterns of vessels. We develop interactive designs such as zoom in and out, tool tips, filters as well as animation of trajectories. More information can be found in Chapter 5.

1.3 Scientific Contribution

This thesis is an extension of previous work done by the author. We update the previous work with more grounded evaluation experiments, and apply the same methods on trawler data. We compare the results of our methods with a few popular methods and display the trajectories with an interactive visualization. Here are the main contributions of the thesis:

1. Develop features for the model which incorporate spatial and temporal information.

2. Improve the performance of fishing activity detection as a corresponding problem from text mining.

3. Create an interactive geo-visualization tool for displaying vessel trajectories and fishing activities intuitively.

1.4 Thesis Outline

In this paper, we present a novel approach for identifying fishing activities using Conditional Random Fields and demonstrate its stability of performance using three different evaluation experiments and by comparing CRF models with autoencoder, data mining approach and HMM. The remainder of the thesis is organized as follows: In Chapter 2, we review relevant literature; in Chapter 3, we explain conditional random fields (CRFs) and then elaborate on how to apply them to identify fishing activities; in Chapter 4, we present three evaluation experiments on CRFs and comparative experiments to demonstrate the stability and effectiveness of our models; in Chapter 5, we present an interactive visualization and its designs as well as its functions; finally, in Chapter 6, we make conclusions and discuss practices and provide direction for future work.

Chapter 2

Background and Related Work

In this Chapter, we first describe the background of CRFs, from the perspectives of undirected model and directed model, generative model and discriminative model, related models in the task of sequential labeling and different CRF models and their applications. Then we present some related research and applications involving AIS data. Finally, we describe some related work dealing with the task of fishing activity detection, which we cite as comparisons for our models.

2.1 Background of CRF

Graphical model, also known as probabilistic graphical model (PGM), is a unifying framework that combines graph theory with probability theory for capturing complex dependencies among random variables, and for representation and inference of multivariate probability distributions [58]. The expression of distributions over many variables can be expensive. For instance, if we have n variables and each variable has r possible values, than the naive representation of the distribution over n variables contains r^n elements. The main idea of the graphical model is factorization which represents the probability cost-effectively. Based on conditional independence within graph structure, graphical models factorize the underlying probabilities into a product of a set of local functions (local probability distributions) [28]. Graphical model comprises a collection of vertices and edges, where vertices represent attributes involved in tasks, edges represent the dependency relationship among those attributes. Based on whether the edges have direction or not, the graph model can be categorised as directed graph or undirected graph.

2.1.1 Directed and Undirected Graphical Models

In the Directed Graphical Models (DGM), all the edges are directed from one vertex to another, which interprets how a probability distribution factorizes into products

of local conditional distributions. Let an ordered pair $G = (V, E)$ be a directed acyclic graph (DAG), where $V = \{V_1, V_2, ..., V_n\}$ is the set of vertices in the graph and $E = \{(V_i, V_j) : i, j = 1, 2, ..., n; i \neq j\}$ is the set of directed edges (ordered pair of vertices) between two vertices in V. Each vertex V_i has a set of parent vertices V_{π_i}, where π_i are the indices of the parent vertices of the vertex V_i. The value of each vertex V_i depend on the value of the parent vertices V_{π_i}. According to the concept of conditional independence, given the parent vertices V_{π_i}, the vertex V_i is conditionally independent of V_{V_i}, where V_{V_i} is the set of vertices that appear before V_i excluding the parent vertices V_{π_i}. Then the joint distribution over vertices V can be factorized using the probability chain rule into a set of local conditional distributions which depend only on a subset of vertices in G as follows:

$$p(v_1, v_2, ..., v_n) = \prod_{i=1}^{n} p(v_i|v_{\pi_i}), \tag{2.1}$$

where $p(v_i|v_{\pi_i})$ is local conditional distributions. Directed graphical model is often used in a generative model. In the task of sequence labeling, popular models such as Bayesian Networks [24], Hidden Markov models [46] and maximum entropy Markov models [36] can be regarded as directed graphical model.

In the Undirected Graphical Models (UGM), also known Markov Random Fields (MRFs), Markov networks or Gibbs distributions, as its name implies, all the edges are bidirectional. Let $G = (V, E)$ be an undirected acyclic graph, where E is the set of undirected edges between two interdependent vertices. Unlike a directed graph where a causal relationship lies in between a vertex and its parent vertices, an undirected graph expresses a correlation relationship between two vertices instead of a causal relationship. Thus, a set of conditional probability distributions multiplied together will not guarantee a consistent joint distribution over all vertices V. However, the probability distribution of an undirected graph can be factorized using local functions (also known as factors, potential functions or compatibility functions) that are defined on the cliques in the graph. A clique c is a subset of the vertices V where all the vertices v_c in c are fully connected. The collection of cliques C is often chosen to be all maximal cliques, where each clique $c \in C$ is not included within any other clique. This can reduce a number of factors and enable a general representation of the probability distribution of the undirected graph which is factorized as follow:

$$p(v_1, v_2, ..., v_n) = \frac{1}{Z} \prod_{c \in C} \Psi_{v_c}(v_c), \tag{2.2}$$

where $\Psi_{v_c}(v_c)$ is a local function of the clique c, and Z is a partition function which normalizes the distribution that takes the form:

$$Z = \sum_{v_1, v_2, ..., v_n} \prod_{c \in C} \Psi_{v_c}(v_c). \tag{2.3}$$

Unlike a directed graph where the sum of all probabilities equals to 1, the sum of all probabilities in undirected graph that are factorised by compatibility functions based on cliques does not obey the probability theory (equals to 1), so normalization is needed to ensure that the sum of probabilities equals to 1. In an undirected graph, the compatibility functions are not directly related to marginal or conditional distributions over the cliques [58]. CRFs are undirected graph models.

2.1.2 Generative model and discriminative model

Let X be a set of input observed variables and Y be a set of corresponding output labels that need to be predicted. Distributions are built over the combined set of input and output variables $X \cup Y$.

A generative model learns the joint probability distributions for individual classes $p(\mathbf{x}, \mathbf{y})$ where $\mathbf{y}$ is a set of class labels and $\mathbf{x}$ is a set of features that can be factorized as $p(\mathbf{y}, \mathbf{x}) = p(\mathbf{y})p(\mathbf{x}|\mathbf{y})$, which interprets how to generate the set of features $\mathbf{x}$ given the set of labels $\mathbf{y}$ [13]. For classification, a generative model first infers the class-conditional densities $p(\mathbf{x}|\mathbf{y})$ and priors $p(\mathbf{y})$, then uses Bayesian rules to determine the posterior probabilities $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})/p(\mathbf{x})$. Directed graph models are often generative models. Models such as the Naive Bayes classifier and the Hidden Markov Model are generative models.

A discriminative model works oppositely, learning the conditional probability distribution which is the probability of $\mathbf{y}$ given $\mathbf{x}$ $p(\mathbf{y}|\mathbf{x})$ [27]. A discriminative model does not involve how the data is generated and does not care about the distribution of Y, it only focuses on the differences among categories, from which it learns the boundaries among classes and directly describes how to assign a set of labels $\mathbf{y}$ given the input variables $\mathbf{x}$. Undirected graph models are often discriminative models. Models

such as Supported Vector Machine (SVM) and Logistic Regression are discriminative models. CRFs are discriminative models.

2.1.3 Related models for sequential labeling

Sequence labeling is the task to assign a label to each member of a sequence of observed variables, and thus output a sequence of labels for a sequence of observations. There are three closely related models for sequential labeling, Hidden Markov Model (HMM), Maximum Entropy Markov models (MEMM) and Conditional Random fields.

Hidden Markov Model is a statistical Markov model, also known as probabilistic finite state automata. It defines the joint probability $P(y, x)$ over a sequence of observations x and a sequence of states y [46]. It is a generative model built on transition probability distributions and emission probability distributions which aims to generate data. However, in a sequential labeling task, it is more useful to use conditional distribution $p(y|x)$ over the sequence of labels given observations. To define the joint probability distribution, Hidden Markov Model needs to make two independent assumptions: first, it applies the Markov assumption that the probability of each state depends only on its previous state; second, it assumes that each observed variable depends only on its corresponding state. However, in reality, in many cases elements in an observation sequence have long distance dependencies. In addition, the features of HMM are limited to observations and cannot be easily incorporated with other knowledge in the domain.

To address the above drawbacks, McCallum et al. proposed Maximum Entropy Markov model (MEMM), also known as Log-linear tagging models [36]. MEMMs is a conditional model which directly models the conditional probability distribution $p(y|x)$. It defines one set of probability distributions that represent the probability of moving from the current state to the next given current state and observation. Since the observations are not generated from the model, by using the theory of Maximum Entropy, non-independent and arbitrary features can be applied. MEMMs train the probability distribution of each state separately which means the normalization is performed locally per state instead of globally. When the state has only a few outgoing transitions, it will ignore the observation. This is known as the Label bias problem,

which is the transitions leaving a given state compete only against each other, rather than against all other transitions in the model [29]. Moreover, the performance of the model will drop if there is an unknown tag in the testing case which does not appear in training set.

Conditional Random Field (CRF) was originally proposed in 2001 by Lafferty et al [29]. It is a discriminative model that combine the advantages of HMM and MEMM and overcome their shortcomings. It directly models the conditional probability distribution $p(y|x)$ without wasting effort on modeling $p(x)$ which is in need for the prediction task. It relaxes the strict independence assumption of HMM, allows complex and highly correlated features that can incorporate contextual information. It uses a global normalizer that sums over all possible states instead of per state normalizer which solves the label bias problem of MEMM. More details about definition, training and inference of CRF will be presented in Chapter 3.

2.1.4 Different CRF models and applications

CRFs are prevalent in solving structured prediction problems [54]. CRFs have many types with different structures, the most commonly used type is Linear-chain CRF with linear chain structure. More generally, CRFs have general versions such as skipped-chain CRF, tree CRF, factorial CRF, and so on.

CRFs has been successfully applied to many areas such as Natural Language Processing (NLP), computer vision, and bio-informatics. In natural language processing, CRFs have been applied to part-of-speech (POS) tagging [45], shallow parsing [52], name-entity recognition (NER) [37, 51], speech modeling [22, 25], information extraction [43, 49], semantic annotation [55], Chinese word segmentation and new word detection [42]. CRFs are also popular in solving problems in computer vision such as image labeling [23], gesture recognition [60], human activity extraction [32], image segmentation [44]. In bio-informatics, CRFs have been applied to gene and protein identification [38], gene prediction [15], RNA secondary structural alignment [50].

2.2 Research and applications using AIS data

AIS has plenty of advantages including open accessibility, being unencrypted, massive data volume, high data transmitting frequency, etc., as we introduced in Chapter

1. AIS data contain detailed information that can be divided into three categories, static, semi-static and dynamic data [9], where static data includes MMSI, vessel length and width, semi-static data includes destination and vessel hazard level and dynamic data includes time, location, speed, course, and turning rate. Thus AIS has become an important, inexpensive and abundant source of valuable information. Consequently, AIS data are applied to various research studies and applications, such as marine traffic anomaly detection, route estimation, collision prediction and path planning [56].

Ristic et al use kernel density estimation for anomaly detection and route prediction [47]. AIS data has also been used in maritime surveillance and vessels tracking and monitoring for maritime safety and security [21], ship patterns mining [63], and estimation of navigation patterns [9]. Research has been done to study the risk of ship collision off the coast of Portugal based on AIS data [53], extract traffic pattern for vessel movement prediction and anomaly behavior detection [33,40], discover fishing areas based on historical AIS data broadcasts by fishing vessels [35, 39], model ship engine exhaust emissions in ports and extensive coastal waters [20], trajectory clustering [34] and mapping underwater sound noise [14].

2.3 Fishing activity detection

Several methods have been used to deal with fishing activity detection.

Many studies of fishing activity detection focus on Trawlers. For example, Mazzarella et al. identified fishing events using a clustering method [35]. Russo et al. applied an artificial neural network to assign fishing effort based on VMS data [48]. Wang et al. use a Generalized Additive Model to estimate catches [61]. Several works have been done using the Hidden Markov model to recognize vessel fishing activities [16, 41, 57]. It is found that trawler fishing activities are highly related with speed [30], meaning speed can provide useful information to aid the classification of fishing activities.

However, for longliners, there is no obvious pattern to distinguish fishing activities using speed information alone. Jiang et al. applied a deep learning approach using autoencoders (AE) that are pretrained with restricted Boltzmann Machines on longliners [26]. These methods make use of the geo-location information of the data to

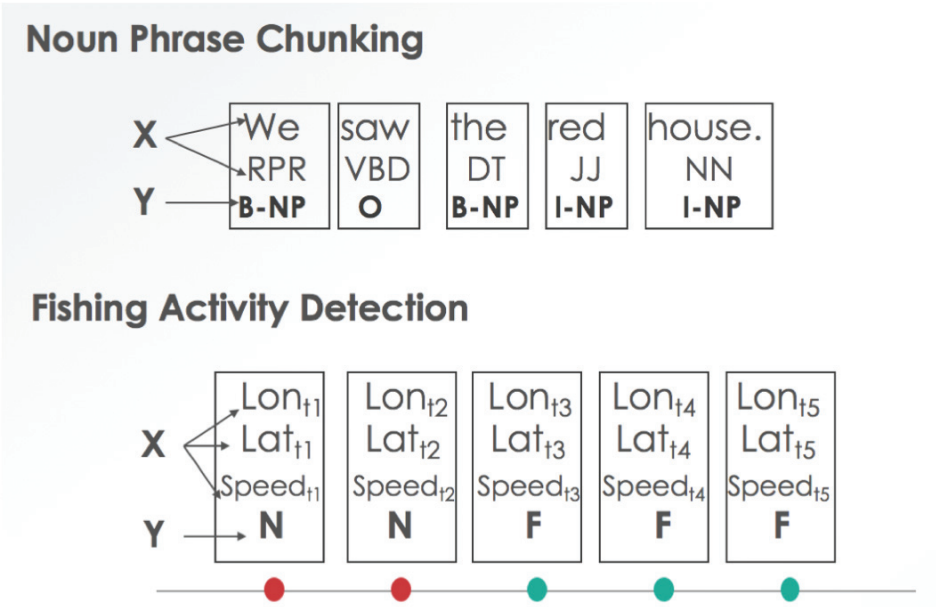


Figure 2.1: The similarity of the task of noun phrase chunking and fishing activity detection.

interpolate the trajectory and turn it into a matrix representation which takes a grid form. Then the task can be regarded as image recognition which segments a sequence of images into fishing and non-fishing parts. The model is developed using restricted Boltzmann Machines where the neural network is tuned by back-propagation. To the best of our knowledge, [26] was the first paper that applied Deep Learning to the task of fishing and non-fishing detecton. Souza et al. applied a Date Mining method – Lavielle's unsupervised trajectory segmentation algorithm, inspired by animal movements, to identify the longliner fishing behavior [16]. Lavielles's algorithm searches for the best segmentation of a time series, then determines the straight or curved segmentation by calculating the relative angles. Afterwards a First-Passage Time algorithm and a Utilization Distribution algorithm are used to reduce false alarms. To the best of our knowledge, [16] was the first paper that we can find that applied Machine Learning to the task of fishing and non-fishing detection. Jiang et al. also applied Recurrent Neural Network (RNN), the Gated Recurrent Unit (GRU) to the task of fishing activity detection. This method integrates gated recurrent units with a proposed partition-wise activation function that can model distinct regions of the feature space.

We find similarity between sequential labeling tasks in natural language processing

and fishing activity detection from the following perspective. In POS tagging, the goal is to label words in sentences using word-category tags. The labels depend on both the word's meaning and context. This task involves two random variables, X and Y, where X is a sequence of words, and Y is a sequence of POS tags. Linear-chain conditional random fields can model the conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ to predict POS tags. Similarly, the task of fishing activity detection involves two random variables, X and Y, where X is the observed random variable (which represents sequences of coordinates and speeds), and Y is the hidden random variable to be predicted (Y is a sequence of fishing and non-fishing labels). This similarity is shown in Fig. 2.1. Consequently, it is reasonable to test whether the linear-chain conditional random fields can model the conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ of fishing–non-fishing to detect fishing activities.

In this thesis, we apply a supervised machine learning method (CRFs) to detect both longliner and trawler fishing activities. We will compare our longliner results with the autoencoder and the data mining method mentioned above and compare our trawler results with HMM.

Chapter 3

Linear-Chain Conditional Random Fields

In this chapter, we present the theory of the linear-chain CRF models. First, we provide basic principles of linear-chain CRFs; then, we present our feature functions; next, we explain the training process in CRFs; further, we present the inference procedure; finally, we introduce our discretization procedure on the features.

3.1 Basic Principles of Linear-Chain Conditional Random Fields

Conditional Random Fields (CRFs) are discriminative undirected graphical models that are designed for sequence labeling [29]. Linear-chain conditional random fields are CRFs with a chain structure, which model the relationship between input and output sequences. Output sequences are modelled in the form of a linear chain, with a link connecting each adjacent output element. Each output element is linked to its corresponding input. Linear-chain CRFs are defined as follows: Let X be a sequence of random variables that we assume are observed, Y be a sequence of random variables that we need to predict. $\mathbf{x}$ denotes a set of observed variables, where $\mathbf{x} \in X$. $\mathbf{y}$ denotes a set of labels, where $\mathbf{y} \in Y$. The length of X and Y are equal. Then, Linear-chain conditional random fields represent conditional probability distributions $p(\mathbf{y}|\mathbf{x})$ for a label sequence $\mathbf{y}$ that take the form

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \mathbf{x}_t), \tag{3.1}$$

where $Z(\mathbf{x})$ is a normalization or partition function

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \mathbf{x}_t), \tag{3.2}$$

where $\Psi_t(y_t, y_{t-1}, \mathbf{x}_t)$ are compatibility functions (also known as local functions) over a subset of random variables $A \subset V$ that factorize the probability distribution, and T

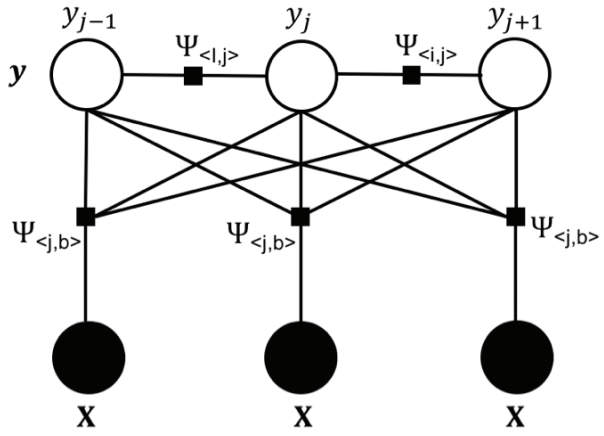


Figure 3.1: Factor graph of linear-chain conditional random fields. $\mathbf{x}$ denotes a sequence of input data, $\mathbf{y}$ denotes a sequence of labels and Ψ_{st} and $\Psi_{\mathbf{i}o}$ denote two kinds of compatibility functions.

is the length of the label sequence $\mathbf{y}$. Given compatibility functions in the log-linear form

$$\Psi_t(y_t, y_{t-1}, \mathbf{x}_t; \lambda) = exp(\sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}_t, t)), \tag{3.3}$$

the conditional probability distribution can be written as

$$p(\mathbf{y}|\mathbf{x}; \lambda) = \frac{1}{Z(\mathbf{x})} exp(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}_t, t)), \tag{3.4}$$

where $f_k(y_{t-1}, y_t, \mathbf{x}_t, t)$ is a feature function, K is the number of feature functions, and λ is a set of weight parameters that help provide weighted average over these feature functions.

In (3.3), the parameter λ_k of compatibility functions $\Psi_t(y_t, y_{t-1}, \mathbf{x}_t; \lambda)$ does not depend on the index t, which means the parameters are shared along the linear chain. Fig. 3.1 visualizes the factor graph of linear-chain conditional random fields with two kinds of compatibility functions. In (3.4), the first sum runs over each position of the linear chain and the second sum runs over each feature function. The conditional probability $p(\mathbf{y}|\mathbf{x}; \lambda)$ can be represented as a mapping function from features to labels. Thus, the selection of feature functions is of great significance for the performance of a model.

3.2 Feature Functions

We use two sets of compatibility functions $\Psi_{<i,j>}$ and $\Psi_{<j,b>}$ to factorize $p(\mathbf{y}|\mathbf{x})$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} (\Psi_{<i,j>}(y_t, y_{t-1}, \mathbf{x}_t) \cdot \Psi_{<j,b>}(y_t, y_{t-1}, \mathbf{x}_t)). \tag{3.5}$$

The first compatibility function $\Psi_{<i,j>}$ is transition compatibility function, which models the transition probability of the labels from one state to another and takes the form

$$\Psi_{<i,j>}(y_t, y_{t-1}, \mathbf{x}_t; \lambda) = exp(\lambda_{ij} f_{<i,j>}(y_{t-1}, y_t, \mathbf{x}_t, t)), \tag{3.6}$$

where $f_{<i,j>}(y_{t-1}, y_t, \mathbf{x}_t, t)$ is transition feature function that takes the form

$$f_{<i,j>}(y_{t-1}, y_t, \mathbf{x}_t, t) = 1_{\{y_{t-1}=i\}} 1_{\{y_t=j\}}, \tag{3.7}$$

where the functions such as $1_{\{y_{t-1}=i\}}$ take the form $1_{\{A=B\}}$ which is defined as

$$1_{\{A=B\}} = \begin{cases} 1 & if A = B \\ 0 & otherwise \end{cases} \tag{3.8}$$

and $< i, j >$ are all possible combinations of labels, y_{t-1} and y_t are the labels of the $(t-1)$-th and t-th position of the linear chain and $\mathbf{x}_t$ is the input sequence. The transition feature functions are parameterized by a set of λ_{ij}. More concretely, if the labels can only take two values: 0 and 1, which is the case in our thesis, the feature functions can be rewritten as

$$f_{<0,0>}(y_{t-1}, y_t, \mathbf{x}_t, t) = 1_{\{y_{t-1}=0\}} 1_{\{y_t=0\}}, \tag{3.9}$$

$$f_{<0,1>}(y_{t-1}, y_t, \mathbf{x}_t, t) = 1_{\{y_{t-1}=0\}} 1_{\{y_t=1\}}, \tag{3.10}$$

$$f_{<1,0>}(y_{t-1}, y_t, \mathbf{x}_t, t) = 1_{\{y_{t-1}=1\}} 1_{\{y_t=0\}}, \tag{3.11}$$

$$f_{<1,1>}(y_{t-1}, y_t, \mathbf{x}_t, t) = 1_{\{y_{t-1}=1\}} 1_{\{y_t=1\}}, \tag{3.12}$$

which are parameterized by $\lambda_{00}, \lambda_{01}, \lambda_{10}, \lambda_{11}$ respectively.

The second compatibility function $\Psi_{<j,\mathbf{b}>}$ is state-observation compatibility function, which models the probability distribution of the labels given a set of observations. $\mathbf{b}$ is a set of input features including selected observations as well as their

combinations. The details of features selection and their combinations are presented in Section 4.2.5. The state-observation compatibility function can be written as

$$\Psi_{<j,\mathbf{b}>}(y_t, y_{t-1}, \mathbf{x}_t; \lambda) = exp(\lambda_{j\mathbf{b}} f_{<j,\mathbf{b}>}(y_{t-1}, y_t, \mathbf{x}_t, t)), \tag{3.13}$$

where $f_{<j,\mathbf{b}>}(y_{t-1}, y_t, \mathbf{x}_t, t)$ is state-observation feature function that takes the form

$$f_{<j,\mathbf{b}>}(y_{t-1}, y_t, \mathbf{x}_t, t) = 1_{\{y_t=j\}} 1_{\{\mathbf{x}=\mathbf{b}\}}, \tag{3.14}$$

where $< j, \mathbf{b} >$ are all possible combinations of input features and its corresponding labels. The state-observation feature functions are parameterized by a set of parameters $\lambda_{j\mathbf{b}}$.

3.3 Model training

The training process comprises the procedure of parameter estimation for feature functions of the model. The parameters can be estimated using maximum-likelihood. Since the probability distribution of CRFs is represented by a product of factors, we use the log-likelihood in the form:

$$l(\lambda) = \log p(\mathbf{y}|\mathbf{x}; \lambda) \tag{3.15}$$

To avoid overfitting, we use L_2 regularizer, which penalizes the weight with a large norm. After substituting the probability distribution of the CRF (3.4) into the log-likelihood (3.15), we get the expression of regularized log likelihood:

$$l(\lambda) = \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}_t, t) - \log Z(\mathbf{x}) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}, \tag{3.16}$$

where σ is a regularization parameter.

To maximize the log likelihood, gradient-based methods are usually chosen for the task. We get the partial derivatives of the log-likelihood (3.15) with respect to λ_k takes the form

$$\frac{\partial l}{\partial \lambda_k} = \sum_{t=1}^{T} f_k(y_{t-1}, y_t, \mathbf{x}_t, t) - \sum_{t=1}^{T} f_k(y_{t-1}, y_t, \mathbf{x}_t, t) p(y_t, y_{t-1}|\mathbf{x}) - \frac{\lambda_k}{\sigma^2}, \tag{3.17}$$

where the first term is the expectation of f_k under the empirical distribution $\tilde{p}$, which takes the form

$$\tilde{p}(\mathbf{y}, \mathbf{x}) = \sum_{t=1}^{T} 1_{\{y=y_t\}} 1_{\{\mathbf{x}=\mathbf{x}_t\}}, \tag{3.18}$$

and the second term can be regarded as the expectation of f_k over the model distribution $p(\mathbf{y}|\mathbf{x}; \lambda)\tilde{p}(\mathbf{x})$ [32].

To calculate the log-likelihood (3.15) and the partial derivatives (3.17), we need to compute the normalization $Z(X)$ and the marginal distributions $p(y_t, y_{j=t-1}|\mathbf{x})$ for t from 1 to T. Lafferty et al use a dynamic programming approach, a forward and backward algorithm, to calculate the model probability $p(\mathbf{y}|\mathbf{x})$ that are presented in terms of matrices.

First, a start label y_0 and a stop label y_{T+1} are added for the sequence of labels $\mathbf{y}$. A set of matrices $\{M_t(\mathbf{x})|t = 1, 2, ..., T+1\}$ is defined, where each matrix $M_t(\mathbf{x}) = [M_t(y_t, y_{t-1}|\mathbf{x})]$ is a $|y| \times |y|$ matrix with random variables and $|y|$ is the number of possible states of y for each position t. Here in our case, we have two possible states $S = \{S_1, S_2\}$. Each element $M_t(y_t, y_{t-1}|\mathbf{x})$ is defined as follow:

$$\begin{aligned} M_t(y_t, y_{t-1}|\mathbf{x}) &= exp(\sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}_t, t)) \\ &= \Psi_{<i,j>} \cdot \Psi_{<j,\mathbf{b}>}, \end{aligned} \tag{3.19}$$

where $M_t(y_t, y_{t-1}|\mathbf{x})$ factorizes the probability from position $t-1$ to t. As shown in Fig. 3.2, the conditional probability $p(\mathbf{y}|\mathbf{x})$ is a product of these matrix elements from the start label to the stop label, which takes the form:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T+1} M_t(y_t, y_{t-1}|\mathbf{x}), \tag{3.20}$$

where $Z(\mathbf{x})$ is written as:

$$Z(\mathbf{x}) = \prod_{t=1}^{T+1} M_t(\mathbf{x}), \tag{3.21}$$

We define the forward vectors as $\alpha_t(\mathbf{x})$ and the backward vectors as $\beta_t(\mathbf{x})$. The base cases take the form:

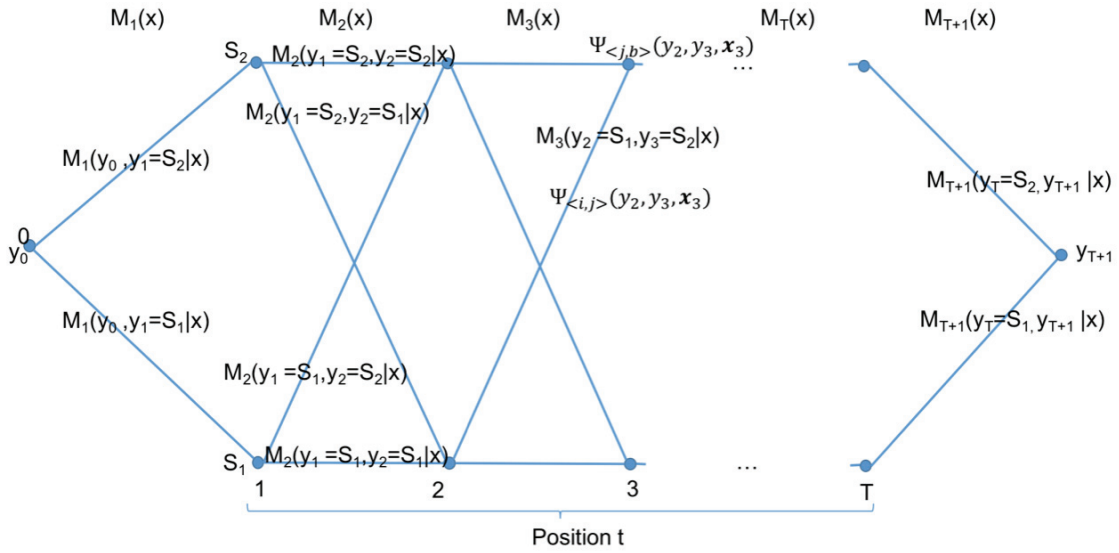


Figure 3.2: Matrix calculation from y_0 to y_{T+1}

$$\alpha_0(y|\mathbf{x}) = \begin{cases} 1 & \text{if } y = \text{start} \\ 0 & \text{otherwise} \end{cases}, \tag{3.22}$$

and

$$\beta_{T+1}(y|\mathbf{x}) = \begin{cases} 1 & \text{if } y = \text{stop} \\ 0 & \text{otherwise} \end{cases}, \tag{3.23}$$

and the recurrent relations take the form:

$$\alpha_t(\mathbf{x})^T = \alpha_{t-1}(\mathbf{x})^T M_t(\mathbf{x}), \tag{3.24}$$

and

$$\beta_t(\mathbf{x}) = M_{t+1}(\mathbf{x})\beta_{t+1}(\mathbf{x}), \tag{3.25}$$

From above equations, given the observation sequence $\mathbf{x}$, the probability $p(y_t, y_{t-1}|\mathbf{x})$ can be written as:

$$
\begin{aligned}
p(y_t, y_{t-1}|\mathbf{x}) &= \frac{1}{Z(\mathbf{x})}\Psi_t(y_t, y_{t-1}, \mathbf{x}_t) \\
&\times \Big(\sum_{\mathbf{y}<1\ldots t-2>}\prod_{t'=1}^{t-1}\Psi_{t'}(y_{t'}, y_{t'-1}, \mathbf{x}_{t'})\Big) \\
&\times \Big(\sum_{\mathbf{y}<t+1\ldots T>}\prod_{t'=t+1}^{T}\Psi_{t'}(y_{t'}, y_{t'-1}, \mathbf{x}_{t'})\Big) \\
&= \frac{\alpha_{t-1}(y_{t-1}|\mathbf{x})M_t(y_{t-1}, y_t|\mathbf{x})\beta_t(y_t|\mathbf{x})}{Z(\mathbf{x})}
\end{aligned}
\tag{3.26}
$$

In general, the parameters λ can not be found using a closed form solution such as setting the gradient (3.17) to zero. Thus, some iterative techniques are chosen to solve this problem [59]. After we calculate the value of the log-likelihood (3.15) and the partial derivatives (3.17) for λ_k, a quasi-newton numerical optimization method, which is called Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), is used to get the optimal parameters λ in the form:

$$
\lambda^* = \underset{\lambda}{\text{argmax}}\ l(\lambda). \tag{3.27}
$$

3.4 Inference

For testing, the goal is to find the most likely label sequence $\mathbf{y}^*$ given observations $\mathbf{x}$, that maximize the conditional model:

$$
\begin{aligned}
\mathbf{y}^* &= \underset{\mathbf{y}}{\text{argmax}}\ p(\mathbf{y}|\mathbf{x}; \lambda^*) \\
&= \underset{\mathbf{y}}{\text{argmax}}\ \frac{1}{Z(\mathbf{x})}\prod_{t=1}^{T}\Psi_t(y_t, y_{t-1}, \mathbf{x}_t; \lambda^*) \\
&\propto \underset{\mathbf{y}}{\text{argmax}}\ \prod_{t=1}^{T}\Psi_t(y_t, y_{t-1}, \mathbf{x}_t; \lambda^*),
\end{aligned}
\tag{3.28}
$$

where λ^* are the parameters learned from the model training procedure presented in the above section. Since we are only interested in the most likely output sequence, we don't need to calculate the normalization $Z(\mathbf{x})$. Inference is performed using the Viterbi Algorithm [54], a dynamic programming algorithm, as shown in the following

steps:

Define

$$\delta_t(j) = \max_{\mathbf{y}<1\ldots t-1>} \Psi_t(j, y_{t-1}, \mathbf{x}_t) \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, \mathbf{x}_t) \tag{3.29}$$

Initialization

$$\delta_1(j) = \Psi_1(\mathbf{x}_1, y_1 = j) \tag{3.30}$$

Recursion

$$\delta_t(j) = \max_{i \in S} \delta_{t-1}(i)\Psi_t(j, i, \mathbf{x}_t) \tag{3.31}$$

$$\Psi_t(j) = \operatorname*{argmax}_{i \in S} \delta_{t-1}(i)\Psi_t(j, i, \mathbf{x}_t) \tag{3.32}$$

Termination

$$p^* \propto \max_{i \in S} \delta_T(i) \tag{3.33}$$

$$y_T^* \propto \operatorname*{argmax}_{i \in S} \delta_T(i) \tag{3.34}$$

Path backtracking

$$y_t^* = \Psi_{t+1}(y_{t+1}^*) \tag{3.35}$$

3.5 Discretization

Compared with POS tagging where the input is a sequence of discrete words, an AIS trajectory consists of real-valued features that are continuous by nature, such as longitudes and latitudes. Conditional random fields can model real-valued continuous features, but the integration of continuous features is not straightforward [17]. They typically require proper normalization so that the value of the feature function is a linear function of the conditional probability $p(\mathbf{y}|\mathbf{x})$. This means a single weight is assigned for all values of the feature function. Since ranges of the value may vary from different feature functions, the value of a single weighted feature function does

not indicate the importance of the feature. The continuous features will be treated linearly, however, the relationships between AIS features and fishing activity labels are non-linear.

Discretization of the continuous features help relax the normalization constraints and allow the conditional random fields to learn $p(\mathbf{y}|\mathbf{x})$ with a more flexible representation. Replacing the continuous features with a set of binary features will have a non-linear effect on features. A discrete CRF will give a distinct weight for each distinct feature function, thus can provide more specification to features, which subsequently increases the discrimination of classes [12]. In this way CRF can capture more complex relationship between AIS features and fishing activity labels. Also, this helps generate a faster and simpler model. It has been shown that discretizing the feature set could increase the performance of CRF [12, 17].

To discretize the continuous features, we use a variant of equal interval size binning discretization in our work. The sizes of bins are determined based on the density of features. Each bin is associated with a set of parameters to fit the model. More detailed information about discretization is presented in Chapter 4.

Chapter 4

Experiments

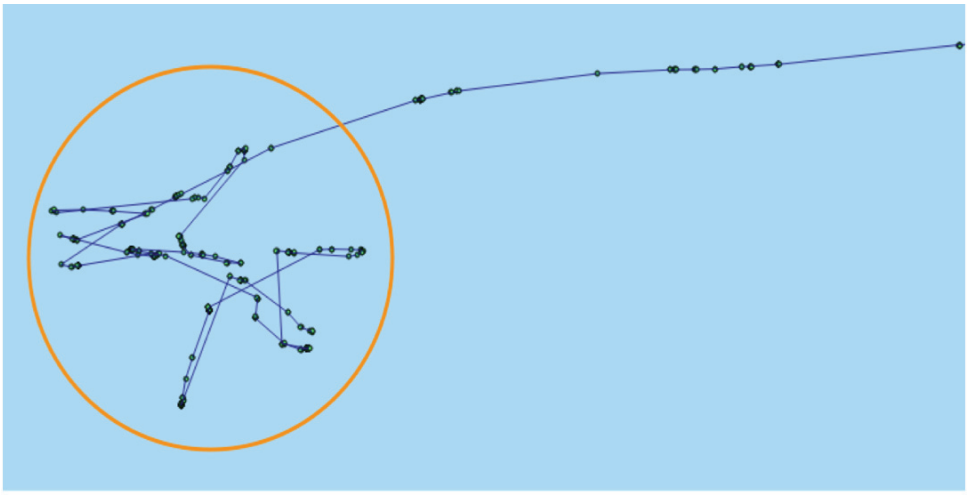
In this Chapter, we present our experiments on both longliner and trawler data to prove the stability and effectiveness of our model. Firstly, we undertake exploratory analysis on the data in terms of trajectory visualization and speed distribution. Next, we preprocess our data to fit the CRF model. Then, we present our experimental results and comparisons with other models for longliners and trawlers respectively, followed by discussions about the models.

4.1 Data Analysis

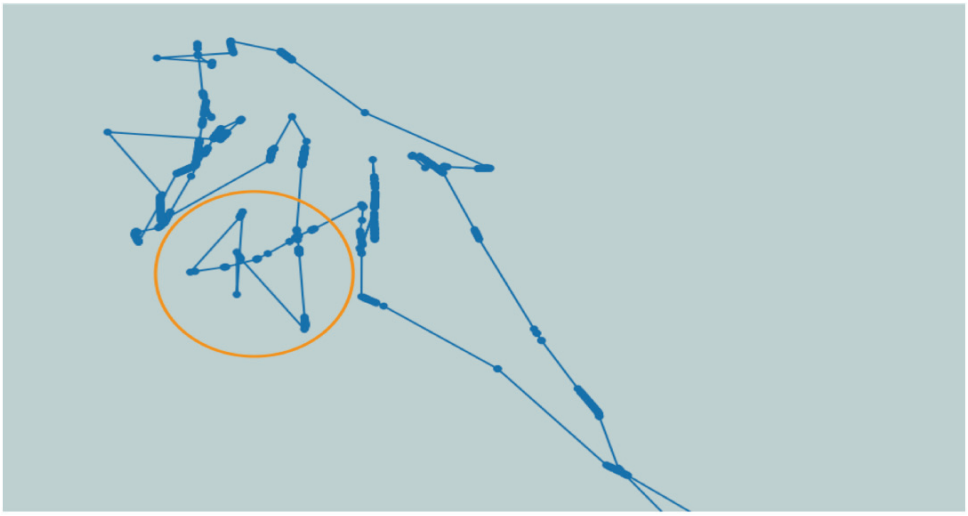
AIS data contain main attributes including MMSI, time, longitude, latitude, speed over ground (SOG) and course over ground (COG). In the experiments, we use historical AIS data from 14 longliners around the world collected from June 1st 2012 to Dec 31st 2013 and 84 trawlers in the North Pacific ocean area collected from Jul 1st 2013 to Jul 31st 2013.

Longliners and trawlers have different fishing patterns. We recovered a part of the tracks for both vessel types from discrete AIS signals. Fig. 4.1 (a) shows the differences between fishing and non-fishing tracks for longliners. The fishing tracks are in the form of a zigzag while the non-fishing tracks tend to follow smooth lines. Fig. 4.1 (b) shows fishing and non-fishing tracks for trawlers. There are no obvious differences between fishing and non-fishing status from the forms of tracks. When fishing, trawlers throw a net into the water, and head in one direction within a certain range of speed to drag the nets.

Besides the distinct fishing patterns of trajectories between longliners and trawlers, the speed distributions of both vessels for fishing and non-fishing are also different. Fig. 4.2 shows the speed density of longliners and trawlers for fishing and non-fishing. Both vessels are not fishing when their speeds are either too high or too low. For longliners, fishing speeds fluctuate, ranging approximately from 1 to 12 knots per



(a)



(b)

Figure 4.1: Differences of fishing and non-fishing tracks. (a) Longliners, (b) Trawlers. Tracks in the orange circle are fishing tracks and the rest are non-fishing tracks.

hour. For trawlers, fishing speed is steady and slow, ranging from about 2 to 5 knots per hour. The overlapping area of speed distribution between fishing class and non-fishing class in the longliners' figure is larger than that of trawlers. Less overlapping area means speed could better indicate fishing and non-fishing status. Consequently, the speed of a trawler can be a good feature in the task of fishing activity detection, while speed of a longliner as a feature cannot reach the same performance.

4.2 Data Preprocessing

To preprocess the data, we perform data cleansing to remove uninformative data, two types of data conversion (one from absolute values to differential values, the other from GPS coordinates to distances and angles), data discretization to transform continuous value into nominal counterparts, and feature selection to fit the model with the most relevant features.

4.2.1 Data Cleansing

For both longliners and trawlers, we sort the data points of each vessel in chronological order. We take four steps to clean our data: First, we remove repetitive data points including points which contain entirely identical information as well as points that have the same longitudes and latitudes, and in this step we removed around 60,000 longliner data points and 20,000 trawler data points; then, we discard data points with incomplete features, such as null values or NAs, and in this step we removed 13 longliner data points and 1 trawler data point; next, we convert the units of speed into a uniform measure which is knots per hour; finally, we calculate the speed by using the calculated great circle distance between two neighboring points divided by the time interval, and we set the threshold of speed to 30 knots per hour, detect and remove outliers if the original speed of the vessel point or the calculated speed exceed the threshold, and in this step we removed around 30,000 longliner data points and 6,000 trawler data points [1].

[1]Note that there are a few data points with the speed exceeding the threshold, but most of the data points with the calculated speed exceeding the threshold. This might because something is wrong with the time stamp or the location of the data points. For instance, two neighboring points have a zero or a very small time interval, the calculated speed might be very large.

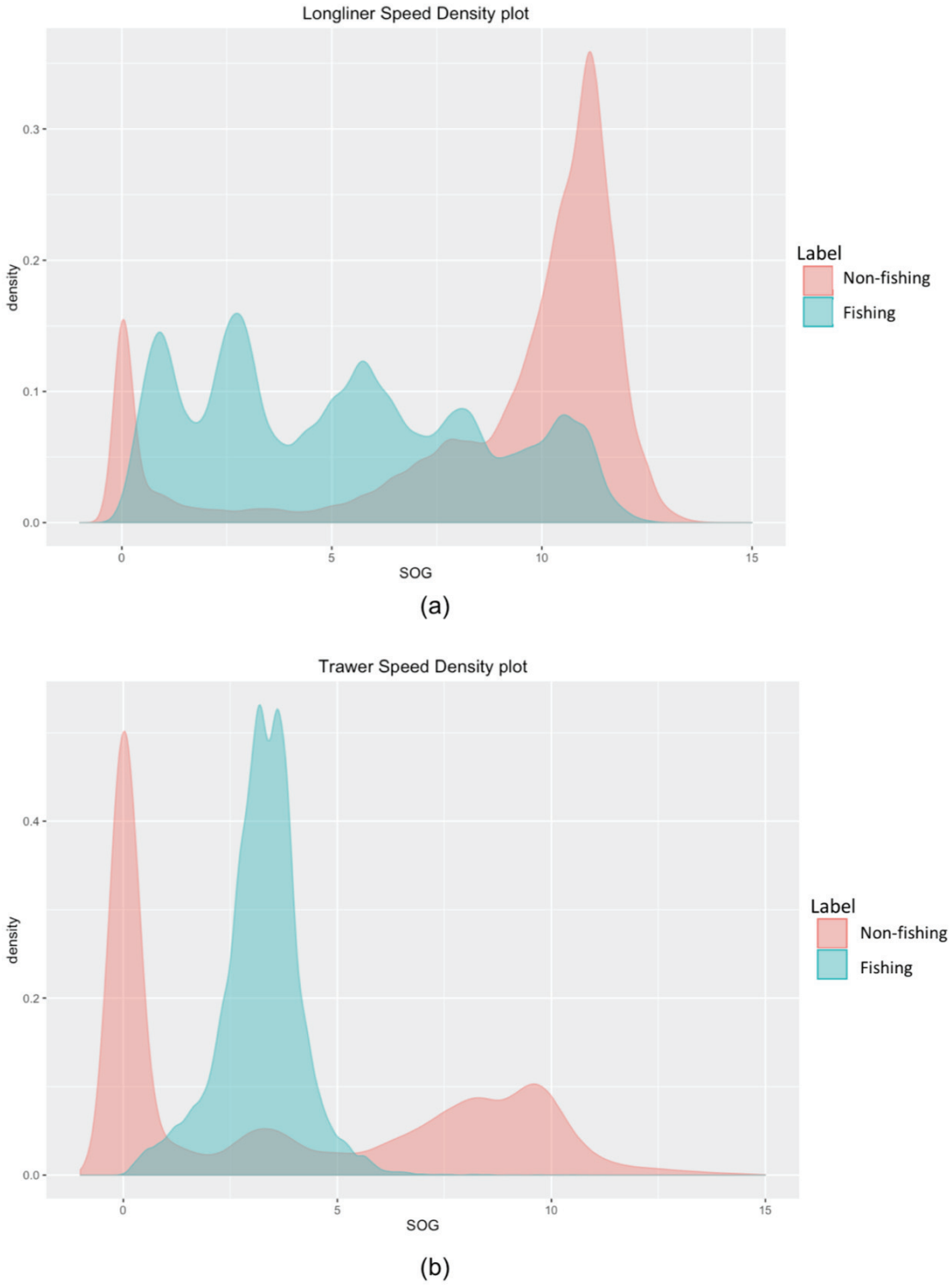


Figure 4.2: Speed distributions for fishing and non-fishing. Color red represents non-fishing and color blue represents fishing. (a) Longliners, (b) Trawlers

Table 4.1: Summary of the 14 longliners data

Track ID	**Track Size**	**# of Fish Points**	**% of Fish Activity**
1	21222	17148	79.6
2	8793	6326	71.7
3	29731	24422	81.0
4	6058	4226	69.4
5	27816	22153	78.6
6	37710	32977	87.4
7	24715	16857	68.2
8	2032	1755	86.4
9	12111	8470	69.9
10	17161	14277	83.2
11	2670	1761	66.0
12	90429	78765	87.1
13	108826	73005	67.1
14	115418	86256	74.7
Total	471892	362026	76.7
Mean	36135	27742	76.5

Before cleansing we have 565,326 longliner data points and 217,860 trawler data points in total. After data cleansing, we have 471,892 longliners data points and 191,840 trawler data points in total. These data points are labeled as fishing and non-fishing tags by a marine biology expert. For longliners, on average, 76.5% of the data points are labeled as fishing. Table 4.1 shows a summary of the 14 longliners after cleansing. For trawlers, 15.7% of all trawler data points are labeled as fishing. Table 4.2 shows a summary of the 84 trawlers after cleansing.

4.2.2 Differential longitude and latitude

In our early experiments, we trained our predictive models using absolute value of longitudes and latitudes. However, we found that the resulting models were overfitting on the training data and cannot generalize to different locations. In order to incorporate the model with geo-information and generalize the model into other areas, we use differential longitude and latitude. Differential longitude is the difference

Table 4.2: Summary of the 84 trawlers data

Measures	Track Size	# of Fish Points	% of Fish Activity
Min	27	0	0
1st Qu	149	0	0
Median	1394	0	0
Mean	2284	360	10.7
3rd Qu	2776	435	18.0
Max	12023	4387	84.4

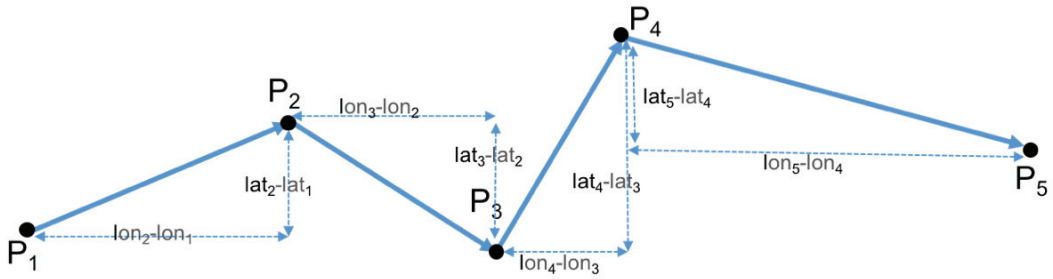


Figure 4.3: Representation of differential longitudes and latitudes from part of a trajectory.

of absolute longitudes between the current and previous data points. Thus the absolute longitudes $LON = [lon_1, lon_2, ..., lon_n]$ are transformed to differential longitudes that takes the form $LON_d = [lon_2 - lon_1, lon_3 - lon_2, ..., lon_n - lon_{n-1}]$. Similarly, we transform absolute latitudes $LAT = [lat_1, lat_2, ..., lat_n]$ to its differential form $LAT_d = [lat_2 - lat_1, lat_3 - lat_2, ..., lat_n - lat_{n-1}]$. Fig. 4.3 shows the representation of differential longitudes and latitudes.

For differential longitude and differential latitude, their values could represent the geographic relationship between two neighboring points. Differential longitude $lon_d \in LON_d$ denotes a location change in horizontal direction while differential latitude $lat_d \in LAT_d$ denotes a change in position in vertical direction. The numeric values of lon_d and lat_d of a point indicate the distance of the point moves from its previous point in horizontal direction and vertical direction respectively. The positive and negative properties of lon_d and lat_d indicate whether the subsequent point move upwards, downwards, forwards and backwards. lon_d and lat_d together suggests how far and in what direction the current point moves from its previous point. Fig. 4.4 describes the

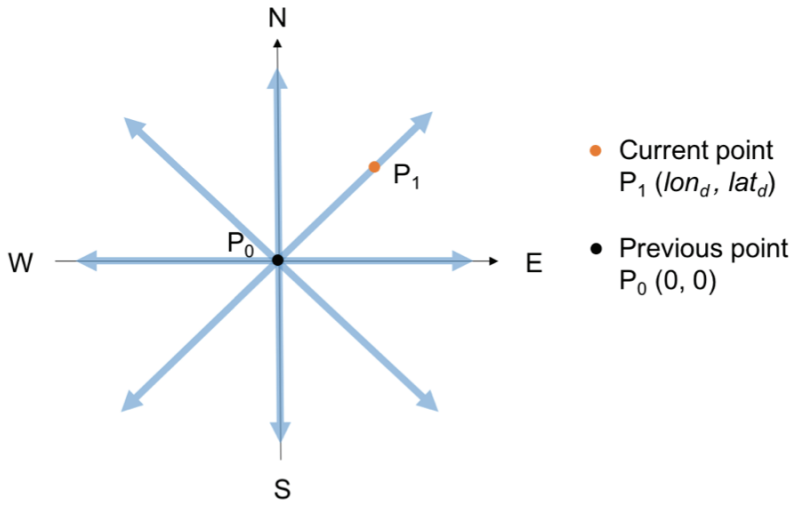


Figure 4.4: Geographic relationship between current point and previous point. Previous point is regarded as origin point P_0 in black. Current point P_1 in orange with coordinates (lon_d, lat_d) gives its position relative to the previous point.

meaning of differential longitude and differential latitude.

4.2.3 Distance and Angle

Besides using differential longitude and latitude, we find a representation of trajectories using distance and angle. As shown in Fig. 4.5, for each point $P_i \in [P_1, P_2, ..., P_n]$, the next point P_{i+1} can be determined by the distance d_i and the counterclockwise angle α_i from trajectory $P_{i-1}P_i$ to trajectory P_iP_{i+1}.

Considering the spherical surface of the earth, we use great-circle distance instead of euclidean distance. We calculate the great-circle distance between two consecutive points based on their longitudes, latitudes and the radius of earth using the **fields** package in R [19]. Then we calculate the angle from the longitudes and latitudes of three consecutive points using the **maptools** package in R [31]. The calculated angle is the angle inside the triangle of three coordinates which is less than 180 degrees. It is not enough to represent the trajectory accurately since it could be either clockwise or counterclockwise. Consequently, we need to convert all the angles to either a counterclockwise angle or a clockwise angle. We obtain the counterclockwise angle for each point using formulae of transformations on the Cartesian Plane.

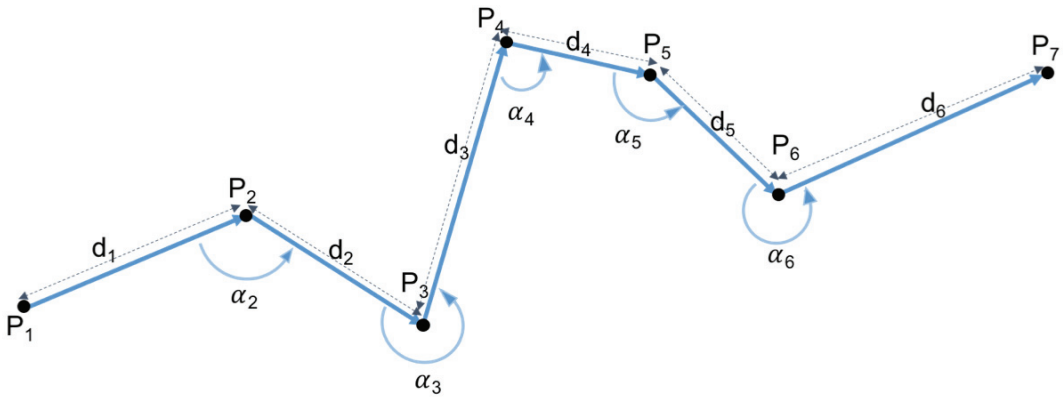


Figure 4.5: A representation of an example of trajectories using points, distances and angles.

4.2.4 Discretization

As mentioned in Section 3, we discretize the values of attributes of data using a variant of equal size interval binning. First, for each attribute, we divide the data into a number of ranges based on the density distribution; Second, we choose different interval sizes for values among different ranges, and choose one interval size for values within the same range.

For each attribute, in order to separate data with high density and low density we divide the data into different ranges. Within a certain range, interval size is inversely proportional to the number of values of that attribute. For instance, we discretize the values range in $[a, b]$ with the equal interval size m, then we will have $(b - a)/m$ discrete values as our feature values after discretization. If the m is too small, we will have lots of feature values and preserve more information from original data but this might cause over-fitting problems. On the contrary, if the m is too large, we will have a small number of feature values but we might lose a lot of information.

For differential longitudes and latitudes, the value of over 95 percent of different longitudes and latitudes falls into [-1, 1]. Thus, we set the interval size m to be

$$m = \begin{cases} 0.05 & l \in [-1, 1] \\ 20 & l > 1, l < -1 \end{cases} \tag{4.1}$$

where fine intervals are selected in the range $[-1, 1]$. Fine intervals provide a larger number of parameters to fit the model compared to coarse intervals, when l is greater

than 1 or less than -1.

Similarly, for distance which is a heavily tailed distribution, we separate the data with three groups, then we set m that takes the form:

$$m = \begin{cases} 0.1 & l \in [0, 5] \\ 5 & l > 5, l <= 100 \\ Inf & l > 100 \end{cases} \tag{4.2}$$

Other attributes such as SOG, COG and angle are more evenly distributed compare to differential longitudes and latitudes and distance. Thus for each of them, instead of separating the data into several parts, we only set one binning size for all values. For SOG, we set m to 0.5. For COG, we set m to 30. For angle, we set m to 30.

Note that we did a number of experiments choosing different sets of parameters m for different attributes, so the value of our parameters presented above are decided based on the best results from our limited experiments. It is not guaranteed to be the best set of parameters for this dataset. But from our experiments, the value of m which are close to the above values have similar results with a variance of no more than 3%, thus we conclude that under our scheme of discretization, the above set of parameters are suitable choices of parameters.

4.2.5 Feature selection and combination

For longliners, we generate two sets of features from differential longitudes and latitudes as well as speed and angle respectively and use them to build our models (CRF1 and CRF2) in the following experiments. First, we perform feature selection to choose useful features from attributes of the AIS data. Then we design our feature functions to create features by making different combinations of features.

In early experiments, we built the model using different combinations of features, such as differential longitude, differential latitude, SOG and COG. We find the models built with differential longitude, differential latitude, and SOG performed the best.

We design our feature functions that incorporate contextual information. In natural language processing, for example, the task of noun phrase chunking, the beginning,

inside and outside tags (BIO) of noun phrases depend highly on the relationship between the word and its contextual information. Analogous to Part Of Speech tagging, in fishing activity detection, vessel behaviors of each time stamp within a certain time period are closely associated with each other. We select neighboring vessel points in the previous two and susequent two time stamps to provide the contextual information of the current point. Then we combine features that had a close relationship with each other, such as longitude and latitude, neighboring longitudes, and latitude, neighboring speeds. After combination, the generated features could provide more information than single original features. More precisely, we use the following feature functions in our experiments, as shown in Fig. 4.6:

1. pairs of differential longitudes and latitudes are selected to represent the positions of data points (colored in orange);

2. pairs of neighbouring differential longitudes are selected to represent the changes of longitudes over time (colored in red);

3. pairs of neighbouring differential latitudes are selected to represent the changes of latitudes over time (colored in green);

4. speed and pairs of neighbouring speed are selected to represent speed information (colored in blue);

5. label of the previous state is selected to help model the transition probability between states (colored in yellow).

The first four feature functions are state-observation feature functions while the last one is state transition feature functions. These feature functions employ spatial and temporal information to aid the classification task. Features are extracted by following the feature functions and moving the sliding window through the time line of data. And we use these features as our first set of features.

Then having distance and angle of each point, we use the distance and the angle as a substitution for differential longitude and latitude to represent geo-information. The feature functions stay the same except we change pairs of longitudes and latitudes into pairs of distances and angles as shown inside brackets in Fig. 4.6. Because we

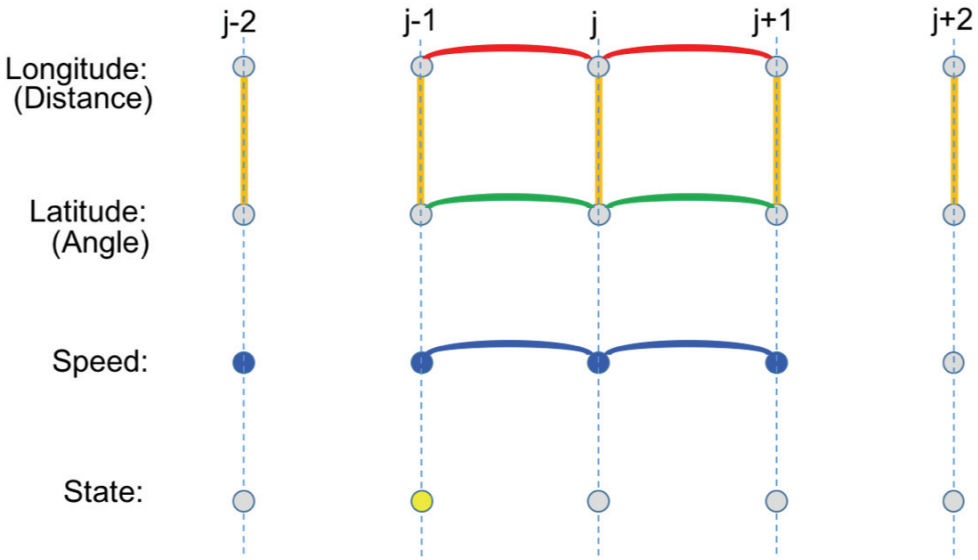


Figure 4.6: Feature functions. When predicting the label of index j, the values of colored dots as well as the paired values connected by solid lines are selected as feature functions.

consider distance and angle to have a close relationship between each other, they together can provide the accurate information of trajectories. We use these features as the second set of features in our three experiments for longliners.

For trawlers, we do not use geoinformation to generate our features. The reason is provided in section 4.4. Trawlers use the last two sets of feature functions from the longliners, namely, speed and state transitions.

4.3 Longliner Results

We train CRF models using CRF++ [4]. For longliners, we design three experiments to evaluate the model: Modified Monte Carlo methods, Iterative Leave One Batch Out (ILOBO) and Stratified LOBO. In each experiment, we use both sets of features as mentioned above in Section 4.2.5. We first build CRF models with features including pairs of differential longitudes and latitudes. Then we build models with features including pairs of distance and angles.

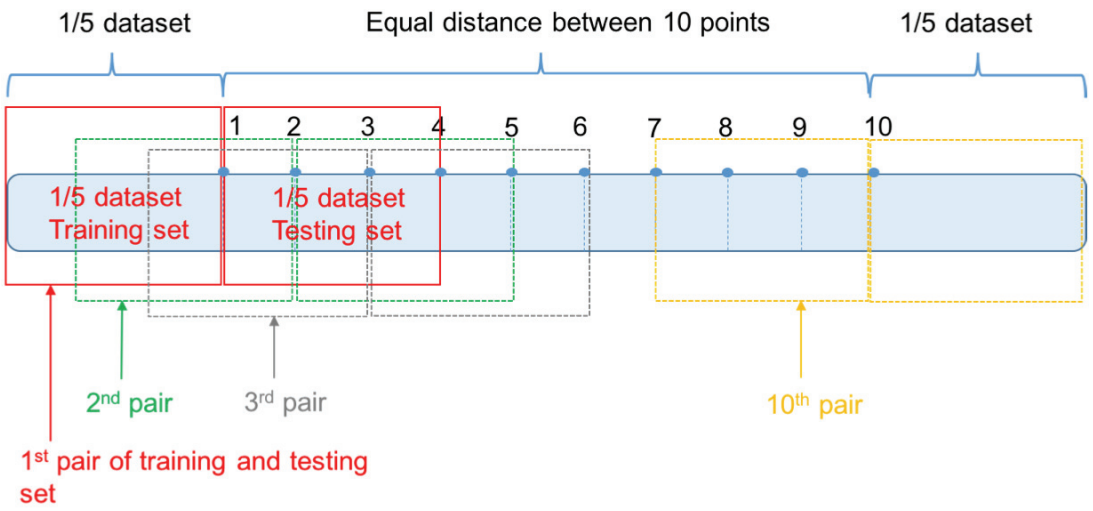


Figure 4.7: Modified Monte Carlo. Choose 10 points as split point, obtain 10 pairs of training and testing sets.

4.3.1 Modified Monte Carlo

Here, we concatenate all the data and apply Monte Carlo methods to get 10 pairs of training and testing data sets. The size of the training and testing set is $\frac{1}{5}$ of the total number of data points. To find the dividing points of the ten Monte Carlo experiments, we first set aside $\frac{1}{5}$ of the whole data set in the front and back of the dataset respectively, then select 10 dividing points with identical intervals that can cover the entire dataset. For each selected point, the $\frac{1}{5}$ portion of the data set to the left of the point and the $\frac{1}{5}$ portion to the right constitute one pair of training and testing set. Fig. 4.7 shows the way we obtain our 10 pairs of training and testing sets.

In the thesis, our models are evaluated using accuracy, sensitivity (4.3), specificity (4.4), positive predictive value (PPV) (4.5) and negative predictive value (NPV)(4.6), as shown in Table 4.3, where the positive class is non-fishing. These measurements provide us with information about the overall performance of the model. They are calculated using following formulas:

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{4.3}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}} \tag{4.4}$$

$$\text{Positive Predicted Value (Precision)} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4.5}$$

$$\text{Negative Predicted Value} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Negative}} \tag{4.6}$$

From each experiment and each measure in Table 4.3, we bold the results of the model that has better performance under. We can see that the results of CRF1 which uses differential longitudes and latitudes as features are slightly better that the results of CRF2 which uses distances and angles as features in terms of accuracy, sensitivity, PPV and NPV. We have applied a paired-sample t-test to compare the classification accuracies of CRF1 and CRF2, the resulting p-value is 0.805 which is greater than the significance level 0.05. So we claim that the results of CRF1 and CRF2 are not statistically significantly different. Since there is not much difference between the two results, we take the example of CRF1 to analyse details of its results. According to the formulas above, on average 72.8% of sensitivity (also known as recall) means 72.8% of all non-fishing activities are predicted correctly by the model where 94% of mean specificity indicates 94% of all fishing activities are predicted correctly. 81.3% of mean positive predicted value (also known as precision) indicates 81.3% of all non-fishing predictions are correct predictions where 92.4% of mean negative predicted value indicates 92.4% of all fishing predictions are correct predictions. Also the standard deviations of sensitivity and PPV are higher than specificity and NPV. Predictions of fishing activities (specificity and NPV) are better than non-fishing activity (sensitivity and PPV). This is because the majority (about 77%) of longliners data are labeled as fishing activity, thus less information about non-fishing activity is provided to the model to help predict non-fishing activities. Overall, the model with 89.8% for mean accuracy provides good results for fishing activities detection.

4.3.2 Iterative Leave One Batch Out

Here, we first split the 14 vessels into 2 groups, group one with 10 vessels for iterative LOBO, and group two with four vessels for stratified LOBO.

Table 4.3: Evaluation of CRF1 and CRF2 for longliners using Modified Monte Carlo methods.

ID	Accuracy		Sensitivity		Specificity		PPV		NPV	
	CRF1	**CRF2**	**CRF1**	**CRF2**	**CRF1**	**CRF2**	**CRF1**	**CRF2**	**CRF1**	**CRF2**
1	**0.868**	0.857	0.481	**0.492**	**0.959**	0.951	**0.730**	0.721	**0.888**	0.879
2	0.872	**0.949**	0.460	**0.859**	0.947	**0.980**	0.610	**0.938**	0.906	**0.953**
3	**0.959**	0.865	**0.826**	0.366	**0.985**	0.965	**0.916**	0.680	**0.966**	0.883
4	**0.973**	0.933	**0.876**	0.564	**0.991**	0.979	**0.953**	0.768	**0.976**	0.948
5	0.860	**0.970**	0.622	**0.870**	0.941	**0.983**	0.785	**0.870**	0.879	**0.983**
6	0.765	**0.879**	0.821	**0.841**	0.739	**0.889**	0.598	**0.674**	0.897	**0.953**
7	**0.851**	0.811	0.743	**0.900**	**0.908**	0.774	**0.809**	0.628	0.871	**0.948**
8	**0.944**	0.778	**0.885**	0.441	0.965	**0.973**	0.900	**0.905**	**0.960**	0.750
9	**0.939**	0.914	0.735	**0.822**	**0.981**	0.951	**0.892**	0.869	**0.947**	0.931
10	**0.946**	0.943	0.835	**0.854**	**0.983**	0.972	**0.940**	0.909	0.947	**0.953**
Mean	**0.898**	0.890	**0.728**	0.701	0.940	**0.942**	**0.813**	0.796	**0.924**	0.918
SD	0.065	0.063	0.157	0.209	0.075	0.065	0.131	0.115	0.040	0.067

To split the vessels into two groups, we first take four vessels from the 14 longliners as group two and then take the rest 10 vessels as group one. To get an accurate evaluation of the model, we need the four selected vessels to be representative of the entire data set. Therefore, we use trajectory size as a criterion to help select these vessels. Since the trajectory sizes of the 14 vessels vary from thousands to over a hundred thousand, as shown in Table 4.1, we categorize the 14 trajectories into three sets based on their sizes. For the three sets, the sizes of trajectories are in the range of thousands, ten thousands and hundred thousands respectively. We then randomly select the number of vessels that are proportional to the cardinality of each set: one vessel in set one, two vessels in set two, and one vessel in group set as the four testing vessels.

For the set of 10 vessels, in each iteration, we consider one vessel as one batch to be the test vessel and build one model on the rest of 9 vessels, and repeat this 10 times. Fig. 4.8 shows the way we divide the 14 vessels into two groups and perform Leave One Batch Out experiments. Table 4.4 shows the results of Iterative Leave

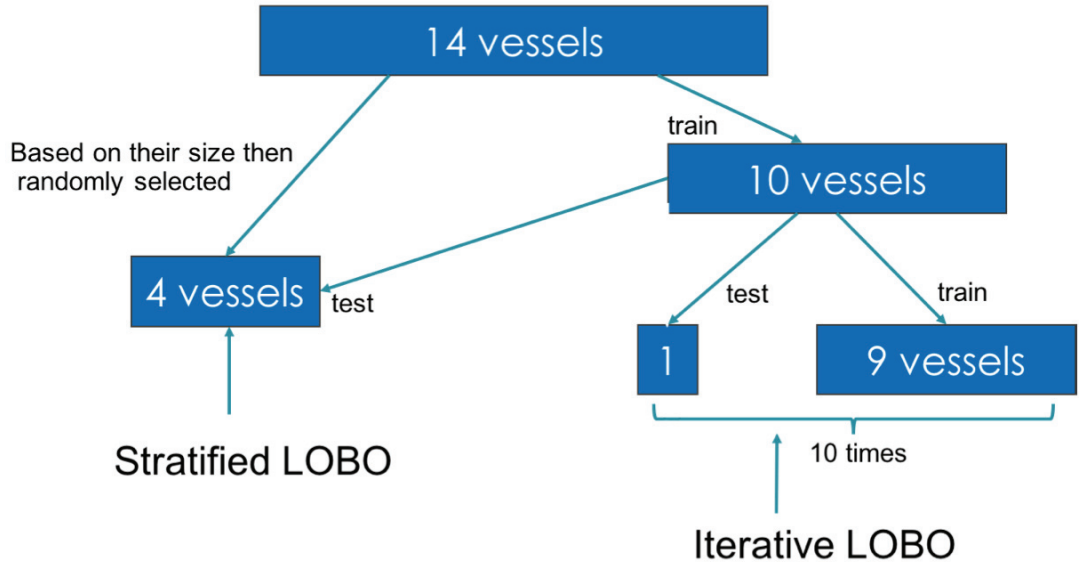


Figure 4.8: How we use 14 vessels for Iterative LOBO and Stratified LOBO experiments.

One Batch Out for the 10 test vessels with CRF1 and CRF2.

From each measures in Table 4.4, the results of CRF2 are slightly better than the results of CRF1 in terms of accuracy, sensitivity and NPV while CRF1 has better results in specificity and PPV. We have applied a paired-sample t-test to compare the classification accuracies of CRF1 and CRF2, the resulting p-value is 0.863 which is greater than the significance level 0.05. We claim that the difference between the results of CRF1 and CRF2 are not statistically significant. Compared to the Modified Monte Carlo experiments, each measure declines a few percentage. Similarly, because specificity and NPV with lower standard deviation are better than sensitivity and PPV with higher standard deviation, predictions of fishing activities are better than non-fishing activity. Overall, the model with 87.2% of mean accuracy provides good results for fishing activities detection.

4.3.3 Stratified Leave One Batch Out

In the Stratified Leave One Batch Out (SLOBO) experiment, We use the selected four vessels from Iterative Leave One Batch Out experiment as independent test vessels and we train the model using the 10 vessels from ILOBO experiment, and evaluate on the rest four vessels individually. The performance of the model on the

Table 4.4: Evaluation of CRF1 and CRF2 for longliners using Iterative Leave One Batch Out.

ID	Accuracy		Sensitivity		Specificity		PPV		NPV	
	CRF1	**CRF2**	**CRF1**	**CRF2**	**CRF1**	**CRF2**	**CRF1**	**CRF2**	**CRF1**	**CRF2**
1	**0.855**	0.806	0.739	**0.784**	**0.902**	0.815	**0.748**	0.625	0.897	**0.905**
2	0.852	**0.860**	0.530	**0.538**	0.927	**0.936**	0.631	**0.665**	0.894	**0.895**
3	0.825	**0.909**	**0.920**	0.840	0.783	**0.940**	0.652	**0.860**	**0.957**	0.931
4	**0.858**	0.845	0.499	**0.561**	**0.956**	0.922	**0.754**	0.660	0.875	**0.886**
5	**0.955**	0.950	**0.743**	0.676	0.986	**0.989**	0.883	**0.899**	**0.964**	0.955
6	0.822	**0.838**	**0.522**	0.509	0.962	**0.984**	0.865	**0.935**	0.812	**0.818**
7	0.919	**0.927**	0.534	**0.635**	**0.979**	0.973	**0.804**	0.787	0.930	**0.944**
8	**0.896**	0.856	**0.441**	0.355	**0.987**	0.957	**0.876**	0.628	**0.897**	0.880
9	0.831	**0.838**	0.552	**0.585**	**0.974**	0.962	**0.918**	0.884	0.808	**0.826**
10	**0.910**	0.905	0.716	**0.888**	**0.976**	0.912	**0.909**	0.784	0.910	**0.957**
Mean	0.872	**0.873**	0.620	**0.637**	**0.943**	0.939	**0.804**	0.773	0.894	**0.900**
SD	0.045	0.046	0.151	0.164	0.063	0.050	0.104	0.120	0.053	0.049

four testing vessels are shown in Table 4.5. We further visualize the classification results in Fig. 4.9.

Similar to the above Modified Monte Carlo experiment, the results of CRF1 are slightly better that the results of CRF2 in terms of accuracy, sensitivity, PPV and NPV while CRF2 has better results in specificity. And on average, both CRF1 and CRF2 have performance with 89% of mean accuracy, close to the results of Modified Monte Carlo experiments.

4.4 Trawler Results

For trawlers, as shown in Fig. 4.1, we observe that there is no obvious difference in trajectories between fishing and non-fishing. We regard geo-information as an unrelated feature in trawlers fishing activity detection. To prove our points of view, we built models using geo-information for trawlers as we did in longliners and the performance of resulting models are not as good as models built without geo-information.

Table 4.5: Evaluation of CRF1 and CRF2 for longliners using Stratified Leave One Batch Out.

ID	Accuracy		Sensitivity		Specificity		PPV		NPV	
	CRF1	CRF2	CRF1	CRF2	CRF1	CRF2	CRF1	CRF2	CRF1	CRF2
1	0.871	**0.877**	0.642	**0.724**	**0.929**	0.917	**0.700**	0.695	0.910	**0.927**
2	0.818	**0.821**	**0.576**	0.512	0.922	**0.953**	0.759	**0.822**	**0.835**	0.821
3	**0.991**	0.960	**0.944**	0.812	**0.998**	0.978	**0.988**	0.817	**0.992**	0.977
4	0.888	**0.905**	0.824	**0.850**	0.919	**0.935**	0.833	**0.876**	0.914	**0.920**
Mean	**0.892**	0.891	**0.747**	0.725	0.942	**0.946**	**0.820**	0.803	**0.913**	0.911
SD	0.072	0.058	0.168	0.151	0.038	0.026	0.125	0.076	0.064	0.065

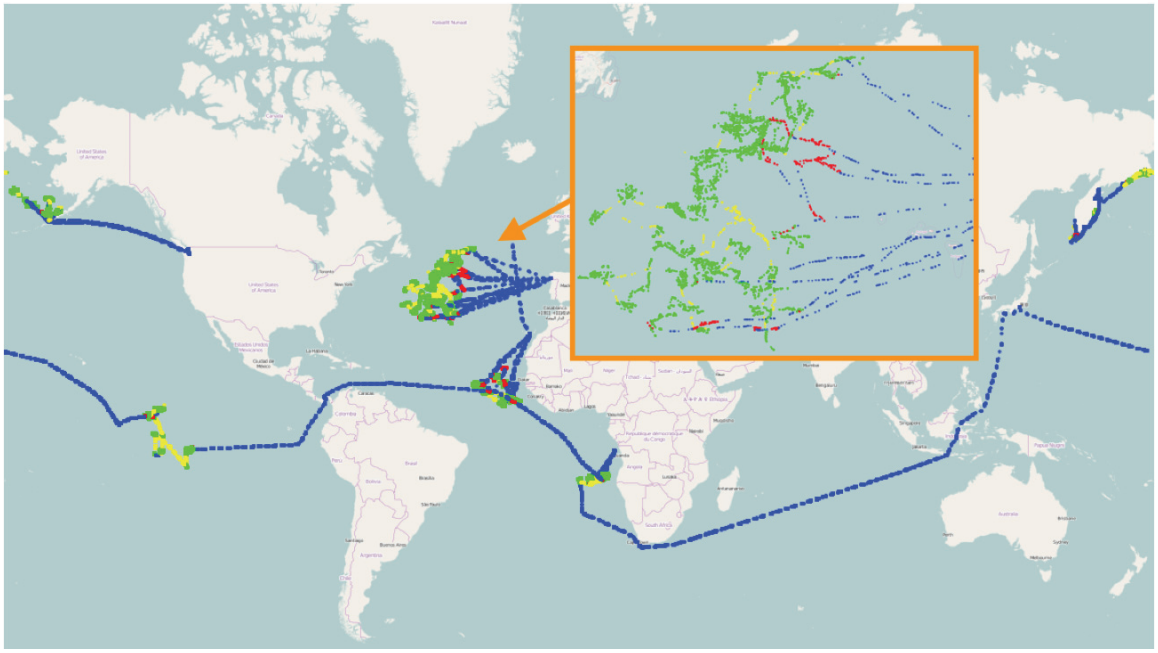


Figure 4.9: The Visualization of four independent testing vessel tracks. Green points mean both the label and the prediction are fishing. Blue points mean both the label and the prediction are non-fishing. Red points mean the label is fishing while the prediction is non-fishing. Yellow points mean the label is non-fishing while the prediction is fishing. The Zoomed-in region gives details of the classification results in the Atlantic ocean area.

Table 4.6: Evaluation of CRF and HMM for trawlers using Modified Monte Carlo methods.

ID	**Accuracy**		**Sensitivity**		**Specificity**		**PPV**		**NPV**	
	CRF	**HMM**	**CRF**	**HMM**	**CRF**	**HMM**	**CRF**	**HMM**	**CRF**	**HMM**
1	0.876	**0.887**	0.952	**0.962**	0.465	**0.480**	0.906	**0.909**	0.642	**0.698**
2	0.906	**0.920**	0.963	**0.982**	**0.289**	0.237	**0.937**	0.934	0.414	**0.548**
3	**0.861**	0.775	0.976	**1.000**	**0.468**	0.146	**0.862**	0.774	0.835	**1.000**
4	**0.885**	0.734	0.983	**0.996**	**0.665**	0.146	**0.868**	0.724	**0.945**	0.944
5	**0.943**	0.937	**0.973**	0.962	0.856	**0.863**	0.952	**0.954**	**0.915**	0.887
6	**0.927**	0.924	**0.924**	0.922	**0.938**	0.933	**0.984**	0.982	**0.754**	0.748
7	0.906	**0.913**	**0.917**	0.912	0.828	**0.917**	0.975	**0.988**	0.579	**0.589**
8	**0.850**	0.842	**0.851**	0.816	0.844	**0.962**	0.962	**0.990**	**0.547**	0.527
9	0.846	**0.851**	**0.847**	0.830	0.842	**0.943**	0.960	**0.985**	0.550	**0.552**
10	0.855	**0.862**	**0.899**	0.895	0.601	**0.672**	0.930	**0.941**	0.501	**0.520**
Mean	**0.886**	0.865	**0.929**	0.928	**0.680**	0.616	**0.934**	0.918	0.670	**0.701**
SD	0.034	0.067	0.050	0.065	0.217	0.371	0.043	0.094	0.185	0.185

In addition, since we have discussed above that the speed distribution is highly related with its fishing activities, we use only speed attributes of the data to build the CRF model for the trawler.

We design three experiments similar to longliners experiments to evaluate the models: Modified Monte Carlo methods, Iterative Leave One Batch Out (ILOBO) and Stratified Leave One Batch Out (SLOBO).

4.4.1 Modified Monte Carlo methods

We apply the same Modified Monte Carlo methods for trawlers to obtain training sets and testing sets as we did for longliners. The results of the experiments on trawlers are shown in Table 4.6 with a comparison with hidden markov model (HMM) in [16] that we reproduced on the same set of data.

For trawler experiments, the positive class is also non-fishing. On average, CRF models have 92.9% sensitivity means 92.9% of all non-fishing activities are predicted correctly by the model, whereas 68% of mean specificity indicates 68% of all fishing

activities are predicted correctly. 93.4% mean positive predicted value indicates 93.4% of all non-fishing predictions are correct predictions where 67% of mean negative predicted value indicates 67% of all fishing predictions are correct predictions. The standard deviations of specificity and NPV are higher than sensitivity and PPV. Predictions of non-fishing activities (sensitivity and PPV) are better than fishing activity (specificity and NPV). This is because the majority (15.7%) of trawler data are labeled as non-fishing activity, thus less information about fishing activity is provided to the model to help predict fishing activities. Overall, the model with 88.6% of mean accuracy provides good results for fishing activities detection. According to Table 4.6, the results of CRF are better than the results of HMM in terms of accuracy, sensitivity, specificity and PPV while HMM has better results in NPV. We have applied a paired-sample t-test to compare the classification accuracies of CRF and HMM; the resulting p-value is 0.197 which is greater than the significance level 0.05. So we claim that results of CRF and HMM are not statistically significantly different.

4.4.2 Iterative Leave One Batch Out (ILOBO)

Of all 84 trawlers, 45 trawlers do not have fishing activity. We decide to use the remaining 39 trawlers with fishing and non-fishing activities to do our Iterative Leave One Batch Out experiment. In order to keep a similar fashion of experiments using longliners data, we had 4 ILOBO experiments on 4 groups of data. First, we randomly order the 39 trawlers and divide them into 3 groups with 10 vessels per group and 1 group with 9 vessels. We then perform an Iterative Leave One Batch Out experiment on each group, which is at each time we select one vessel as testing data and the rest of nine vessels as training data and repeated until every vessel has been chosen as a testing vessel. Since we have 39 experiments, in order to simplify and better present the results of the 4 groups, we use a bar graph to show the average performance of each group instead of a table listing every experiment. The results of each group using the same five measurements are shown in Fig. 4.10.

The average accuracy for the 4 groups are 89.5%, 84.9%, 86.7% and 81.4% respectively. Compared to the average performance of Modified Monte Carlo experiments, each measure declines a few percentages, especially specificity and NPV. There exists

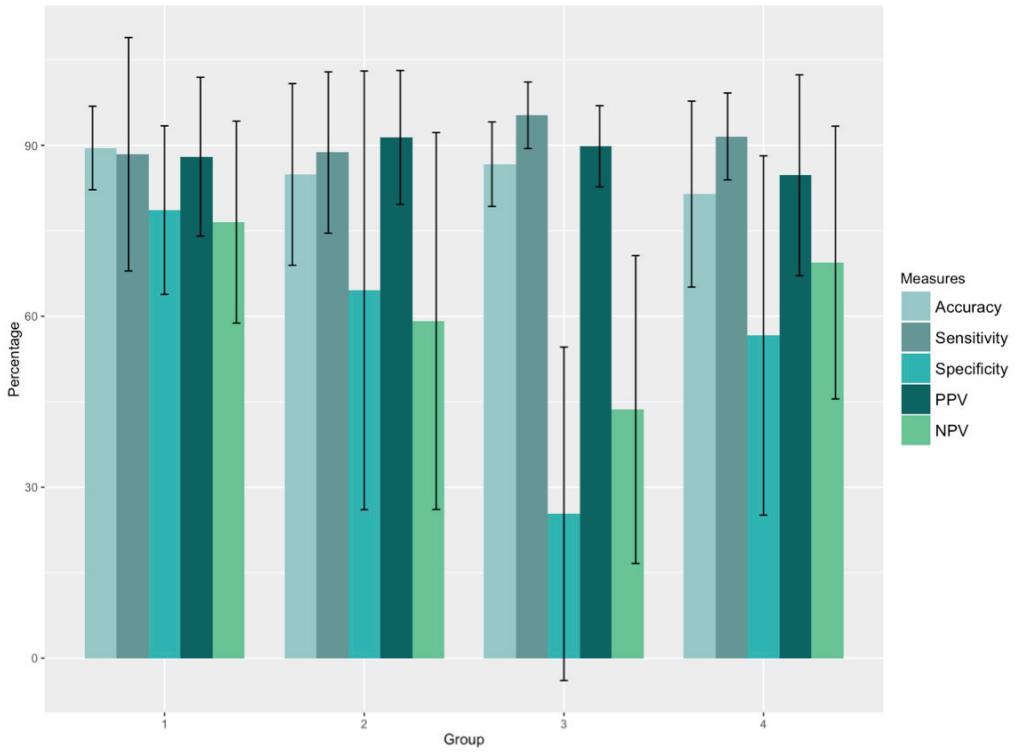


Figure 4.10: The results of ILOBO with CRF on 4 groups of trawler data. X axis corresponds to each group while Y axis corresponds to percentage. Height of each bar represent the average performance of experiments per group. Length of the line on top of each bar shows the standard deviation of the performances per group. Five evaluation measures are shown with five different colors.

Table 4.7: Evaluation of CRF and HMM for trawlers using Stratified Leave One Batch Out.

ID	Accuracy		Sensitivity		Specificity		PPV		NPV	
	CRF	**HMM**	**CRF**	**HMM**	**CRF**	**HMM**	**CRF**	**HMM**	**CRF**	**HMM**
1	0.71	**0.75**	**0.97**	0.90	0.35	**0.71**	0.90	**0.96**	**0.67**	0.47
2	**0.85**	0.84	0.84	**0.93**	**0.86**	0.69	**0.93**	0.85	0.72	**0.83**
3	**0.84**	0.83	0.94	**0.99**	**0.82**	0.40	**0.99**	0.95	0.40	**0.82**
4	**0.92**	0.92	**0.98**	0.94	0.58	**0.69**	**0.83**	0.80	**0.93**	0.89
5	**0.88**	0.87	0.78	**0.82**	0.90	**0.93**	0.93	**0.98**	**0.71**	0.51
6	**0.68**	0.57	**0.87**	0.84	**0.52**	0.51	0.83	**0.93**	**0.60**	0.28
7	**0.85**	0.76	0.99	**1.00**	0.42	**0.76**	0.95	**1.00**	**0.84**	0.01
Mean	**0.82**	0.79	0.91	**0.92**	0.64	**0.67**	0.91	**0.92**	**0.70**	0.54
SD	0.09	0.11	0.08	0.07	0.22	0.17	0.06	0.07	0.17	0.33

variances among each group. For example, group 3 has the lowest performance and the highest standard deviation in terms of specificity and NPV. This is because there is a great variance of fishing percentage per vessel among vessels. Since we randomly divide vessels into four groups, group 3 happens to have vessels with least amount of data labeled as fishing.

Because sensitivity and PPV with lower standard deviation are better than specificity and NPV with higher standard deviation, predictions of non-fishing activities are better than fishing activity. Overall, the model has 85.6% of mean accuracy for identifying fishing activities.

4.4.3 Stratified Leave One Batch Out (SLOBO)

In our SLOBO experiment, which is different from the SLOBO experiments for longliners, we have a second trawler data set including 7 trawlers from Jan 1st, 2011 to Oct 31st, 2015. We applied the same preprocessing techniques on this brand new data set and we use this data set as our test data. After cleansing we have 835736 data points.

According to the work in [16], we use 25000 points of our first trawler data set to build our CRF model, and test on the 7 new trawlers data set respectively. The results

of our 4 models are shown in Table 4.7 together with the results using HMM in [16]. According to the Table 4.7, the results of CRF are better than the results of HMM in terms of accuracy and NPV while HMM has slightly better results in sensitivity, specificity and PPV. On average, CRF has slightly lower variance in results and performs better than HMM. We have applied a paired-sample t-test to compare the classification accuracies of CRF and HMM, the resulting p-value is 0.225 which is greater than the significance level 0.05. So we claim that the difference between the results of CRF and HMM are not statistically significant.

Compare to the modified Monte Carlo experiment, the result of SLOBO has a few percentage declines in each measure and an increase in standard deviation. Again, predictions of non-fishing activities (sensitivity and PPV) are better than fishing activity (specificity and NPV). Overall, the model has 82% of mean accuracy for fishing activities detection.

4.5 Comparison of results

We make comparisons of our CRF models with other models on the task of fishing activity detection. Because there are significant differences of fishing patterns between longliners and trawlers, some methods that work well on longliners might not be applicable on trawlers. Consequently, for longliners and trawlers, we compare CRFs with different methods.

4.5.1 Longliner comparison

We reproduce the autoencoders (AE) [26] and the data mining approach (DM) [16] on the same set of data in the previous ILOBO experiment. The difference is that these two methods only use geo-location information as features. The results of these two methods are shown in Table 4.8 as comparisons with the performance of CRFs in Table 4.4.

According to the Table 4.8, CRF performs the best in terms of accuracy, sensitivity, specificity and NPV, while the data mining method performs the best in PPV and the autoencoder performs the same as CRF in specificity. In addition, on average CRF has a lower standard deviation than the autoencoder and the data mining

method. We have performed a paired-samples t-test to compare the classification accuracies of CRFs and autoencoders, the resulting p-value is 0.057. We also performed paired-samples t-test to compare the classification accuracies of CRFs and the data mining approach, the resulting p-value is 0.032. Though the p-value is below the significance level 0.05, it is greater than 0.01 and still close to 0.05 so that we can not claim that they are systematically different. Despite the fact that both the CRF and AE as well as CRF and DM are not systematically different, CRFs can perform as well as and sometimes better than autoencoders and the data mining approach.

4.5.2 Trawler comparison

For trawlers, we compare CRFs with HMM in [16] using the same data set in both the Modified Monte Carlo experiment shown in Table 4.6 and Stratified Leave One Batch Out as shown in Table 4.7 which are presented in the above section.

4.6 Analysis and Discussion

For longliners, the results of the three experiments are consistent. The average accuracy of CRF1 using differential longitudes and latitudes in the three experiments is 88.7% with 6.1% average standard deviation. For CRF2 which uses distance and angle, the average accuracy is 88.5% with 5.6% average standard deviation. CRF1 and CRF2 have similar results which indicates that differential longitudes and latitudes works similar to distance and angle. Their results also indicates that both sets of features are suitable for the task of fishing activity detection. Further, since we use completely different parameters in discretization of the two sets of features, the results prove that our discretization works well in the experiments and the results won't be impacted a lot if we slightly change the setting of parameters in discretization. Since we use the same feature functions to create pairs of closely related features on both the sets of features, the results indicates that the generated features provide useful contextual information that help the models to predict fishing activities. By comparing the results of the second and the last experiments, we find that the models built using Iterative Leave One Batch Out perform as good as the model built in Stratified Leave One Batch Out. Since ILOBO built models with different vessels and test on different vessels while SLOBO built one model and test on different vessels,

Table 4.8: Evaluation of CRF, AE and HMM for longliners using Iterative Leave One Batch Out.

ID	**Accuracy**			**Sensitivity**			**Specificity**			**PPV**			**NPV**		
	CRF	**AE**	**DM**	**CRF**	**AE**	**DM**	**CRF**	**AE**	**DM**	**CRF**	**AE**	**DM**	**CRF**	**AE**	**DM**
1	**0.86**	0.83	0.65	**0.74**	0.60	0.45	0.90	**0.93**	0.91	0.75	0.77	**0.85**	**0.90**	0.85	0.57
2	0.85	0.85	**0.89**	0.53	0.39	**0.71**	0.93	**0.96**	0.93	0.63	**0.76**	0.70	0.89	0.86	**0.93**
3	0.83	**0.86**	0.46	**0.92**	0.58	0.35	0.78	**0.94**	0.93	0.65	0.76	**0.95**	**0.96**	0.88	0.25
4	0.86	0.86	**0.87**	0.50	0.51	**0.66**	**0.96**	0.95	0.95	0.75	0.74	**0.82**	0.88	0.88	**0.89**
5	**0.96**	0.86	0.89	**0.74**	0.44	0.54	**0.99**	0.94	0.98	**0.88**	0.57	0.83	**0.96**	0.90	0.90
6	**0.82**	0.76	0.76	0.52	0.34	**0.59**	**0.96**	0.94	0.89	**0.87**	0.68	0.81	**0.81**	0.78	0.73
7	**0.92**	0.80	0.54	**0.53**	0.00	0.20	**0.98**	0.93	0.94	0.80	0.00	**0.81**	**0.93**	0.85	0.50
8	**0.90**	0.84	0.86	**0.44**	0.38	0.55	**0.99**	0.94	0.96	**0.88**	0.55	0.80	**0.90**	0.88	0.80
9	0.83	0.86	**0.88**	0.55	0.62	**0.89**	**0.97**	0.94	0.86	0.92	0.76	**0.93**	0.81	**0.89**	0.80
10	**0.91**	0.85	0.74	**0.72**	0.56	0.49	**0.98**	0.94	**0.98**	0.91	0.76	**0.96**	**0.91**	0.87	0.66
Mean	**0.87**	0.84	0.75	**0.62**	0.44	0.54	**0.94**	**0.94**	0.93	0.80	0.64	**0.85**	**0.89**	0.86	0.71
SD	0.05	0.03	0.16	0.15	0.19	0.19	0.06	0.01	0.04	0.10	0.24	0.08	0.05	0.04	0.22

this further proves the stability and potential of the model in future fishing activity detection.

The attribute Course Over Ground is not equal to the angle we calculate from the trajectory. It gives the information about the actual direction that the vessel moves at a certain point. Between two neighboring points, COG might change a lot because of some missing information during the time interval of the two points. Due to some influence such as ocean currents, course over ground might not be in the same direction as the vessels are moving forward. Thus, COG could not provide accurate information about the trajectory. This explains why COG is not a good feature for our model.

From the comparison of the performance in CRFs, autoencoders [26] and the data mining methods [16], as shown in Table 4.8, we find CRFs can have better classification accuracy than the autoencoder and the data mining method, in terms of mean and standard deviation. We also find CRFs do not suffer from imbalanced data as do autoencoders: in experiment 7, autoencoders labeled all data points as positive class with 0% of sensitivity and 0% of PPV, whereas CRFs have 53.4% of sensitivity and 90% of PPV. In these three models, only CRFs incorporate both geo-location and speed information into features while the autoencoders and the data mining approach only use geo-location features. Since CRFs that are built only with geo-location features do not work well, the results of CRFs presented in this thesis prove that speed could also provide useful information in identifying fishing activities.

For trawlers, we have 88.6% of accuracy with 3.4% of average standard deviation in Modified Monte Carlo experiment, 85.6% of accuracy with 3.4% of average standard deviation in Iterative Leave One Batch Out experiment, and 82% of accuracy with 18% of average standard deviation in Stratified Leave One Batch Out experiment. We have compared the performance of CRFs with HMM [16] in Modified Monte Carlo experiment and Stratified Leave One Batch Out experiment, as shown in Table 4.6 and Table 4.7. Since speed has been regarded as the most important feature that is closely related to fishing activity while geo-location information will generate noise in determine fishing activity of trawlers, both CRF and HMM use only speed to build the model. The main difference between CRF and HMM is that CRF incorporates speed information of neighbors while HMM only uses the corresponding speed and

assumes the neighboring speed is independent of each other. We find that CRF has slightly better classification accuracy and lower standard deviation than HMM in most cases. This means CRF has the potential to behave more stable than HMM. Though the advantages of CRF in trawler experiments are not so evident, it might because the value of speed already has close relationship to the fishing activity as shown in Fig. 4.2 (b) and consequently the influence of dependency among neighboring speeds can be ignored.

There exists a higher variance in results (especially specificity and NPV which is related to fishing activity) of ILOBO and SLOBO than in Modified Monte Carlo method, for both longliners and trawlers, especially trawlers. This is because each vessel varies a lot in trajectory sizes as well as the percentages of fishing activities per vessel. Since ILOBO and SLOBO randomly divide the data set into training and testing sets by vessels, it is possible that some vessels provide less useful information or more noise than others that are chosen to built the model or test the model, and thus have an impact on the results.

Chapter 5

Geovisualization

This chapter presents an interactive geovisualization of the vessel trajectories using the AIS data of trawlers and longliners in this thesis. This visualization enables users to explore detailed information and fishing patterns of the longliners and trawlers and provides them with powerful data selection and inspection. In addition, the results of experiments are also displayed in this visualization to assist with error analysis. This chapter is organized as follows: Section 5.1 describes the design of this visualization based on several motivations; Section 5.2 presents the tools and techniques which are used to build this visualization; Section 5.3 displays each part of the visualizations corresponding to the designs; Section 5.4 presents the analysis of the results from the visualization.

5.1 Motivation and design

The original data set and experimental results are presented as tables. With large amount of data points laid out in rows and columns, it is hard to discover patterns behind these numbers. Similarly, from those percentages presented in the results of experiments, it is difficult to analyse the model with great insight such as where and why the model made errors in classification. To gain insights into the data set and the results, we first used QGIS software to visualize the data points on the map. From QGIS we could see the trajectories with different colors for fishing and non-fishing status, but still it is not smooth enough to visualize the target area as well as detailed information. For instance, if we want to see more information on the trajectory, the layout would be a somewhat messy. And once we create the trajectories by connecting discrete data points, the trajectories could not be changed and we can only build a new one if we want to add some other data points. Thus we made this visualization to better reveal the fishing patterns of longlingers and trawlers, easily view the trajectories, quickly find the target, get deep insights into the results,

create a convenient and interactive way to explore the data and show the animated trajectory based on time lines. Our work is also inspired by Aaron Koblin, who implements a series of aesthetic visualizations of flight patterns in United States [1]. By using altitudes, identity, regions of airplanes and airline hubs as indications of colors in flight trajectories, his visualization shows interesting flight patterns during flying, landing and taking off procedures.

This visualization is designed in three parts: The first part is a static visualization of trajectories which shows the geo-location of each vessel trajectory, their fishing and non-fishing status as well as their speed variations among the whole trajectories. The second part is an animated visualization which shows the moving trajectories as well as the different patterns between fishing and non-fishing status based on the time lines. The third part includes the comparisons of experimental results of this thesis with the labels of the original data set displayed on the visualization of trajectories for result analysis.

5.2 Interactive Techniques

We use the Tableau Software to build this visualization. Tableau Software is a graphical system for performing exploration and analysis of customer data sets. It is a commercial system based on the Polaris research project which is powerful tool for building different kinds of interactions [3]. Tableau offers plenty of functions and operations related to maps which makes it an ideal tool for our tasks. Based on our designs, we applied zoom and pan, tool tip, filter, and animation in our interactive visualization. Fig. 5.1 shows the interactive techniques we used in our visualizations.

Zoom and pan enables users to find the targeted area fast, have a clear sight into a high density region and see the details of trajectories. The user can select a particular region to zoom in on and return to the initial view by one click.

Tool tip provides a convenient way to display detailed information for each point. When a mouse hovers over a data point on the map, the respective information including longitude, latitude, speed, time, and fishing and non-fishing status will appear in a box. When the mouse moves away, the box will disappear.

Filter enables users to display the trajectory by their selection. We use three types of filters: time filter, vessel identification (MMSI) filter, fishing and non-fishing status

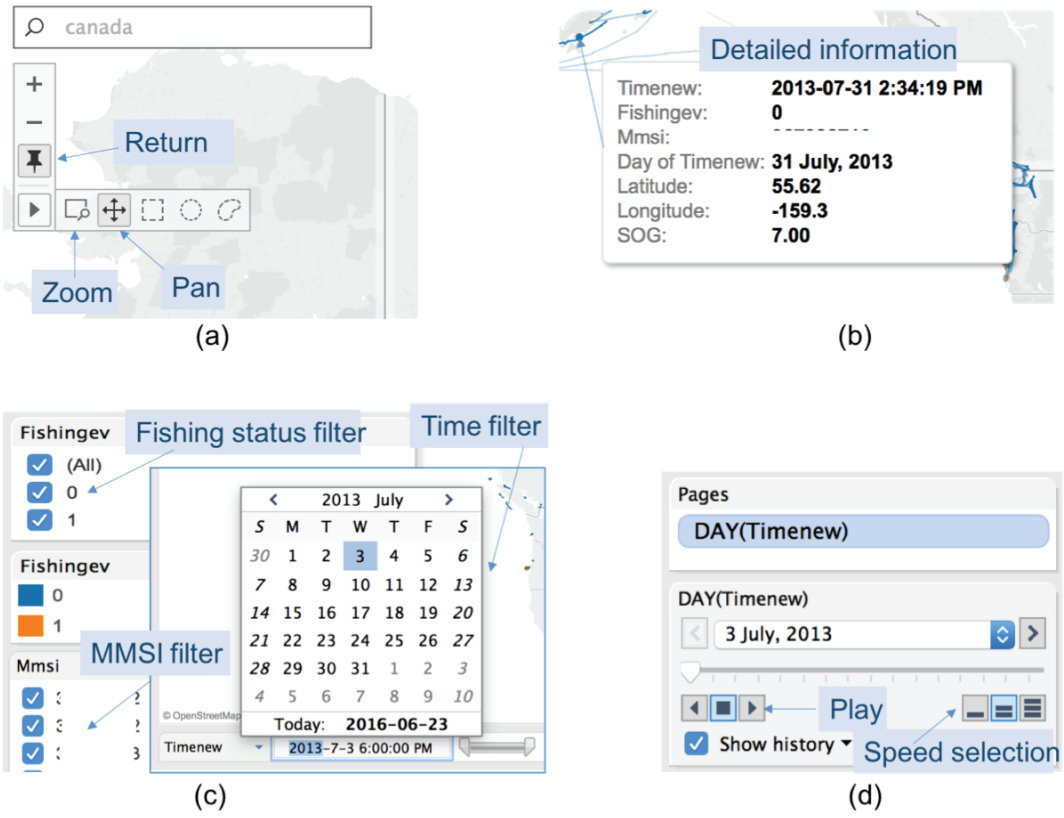


Figure 5.1: Four Interactive Techniques. (a) Zoom and pan. (b) Tool tip. (c) Filters. (d) Animation.

filter. Time filter allows users to choose to display trajectories in a contain period of time. Vessel identification filter allows users to select one or more particular vessels to focus on. Fishing status filter allow users to select to view fishing or non-fishing data individually or both parts.

Animation provides a way to see the dynamic trajectories based on the time lines. The time is divided by day so that each tick represents one day. During each tick, the trajectory of that day will appear. By clicking the play button, the trajectory of each day will appear in an automatic way. Users could stop anytime by clicking anywhere on the map. Users could also choose the speed of tick as well as an amount of history ticks to show.

5.3 Visualizations

We first connect Mysql database to Tableau, extract data points from the database and apply them on to an online map. We design each view of our visualization through a Tableau sheet, and then place them together in a Tableau dashboard. Here we design three kinds of maps both for trawlers and longliners: static maps, animated map and experimental results map. Static maps show the overall vessels, fishing and non-fishing status as well as their speed distributions. Animated maps show above static maps with animation. The experimental results map shows the comparisons between experimental results and the true labels of the data. Users could use the techniques above to obtain information based on their needs in all these maps.

5.3.1 Static Map

In a static map, users could acquire comprehensive information on vessel trajectories from details of points, to selected regions or vessels, to overall trajectories. We made three maps to convey three different categories of information: distributions of different vessels, distributions of fishing and non-fishing trajectories and speed distributions among vessels. To do so, we assign different maps with a different color palette based on three attributes of the AIS data, MMSI, fishing and non-fishing labels, and SOG.

In the map to show distributions of different vessels, we differentiate vessels with 20 colors. Fig. 5.2 shows the distributions of all the longliners. Fig. 5.3 shows the

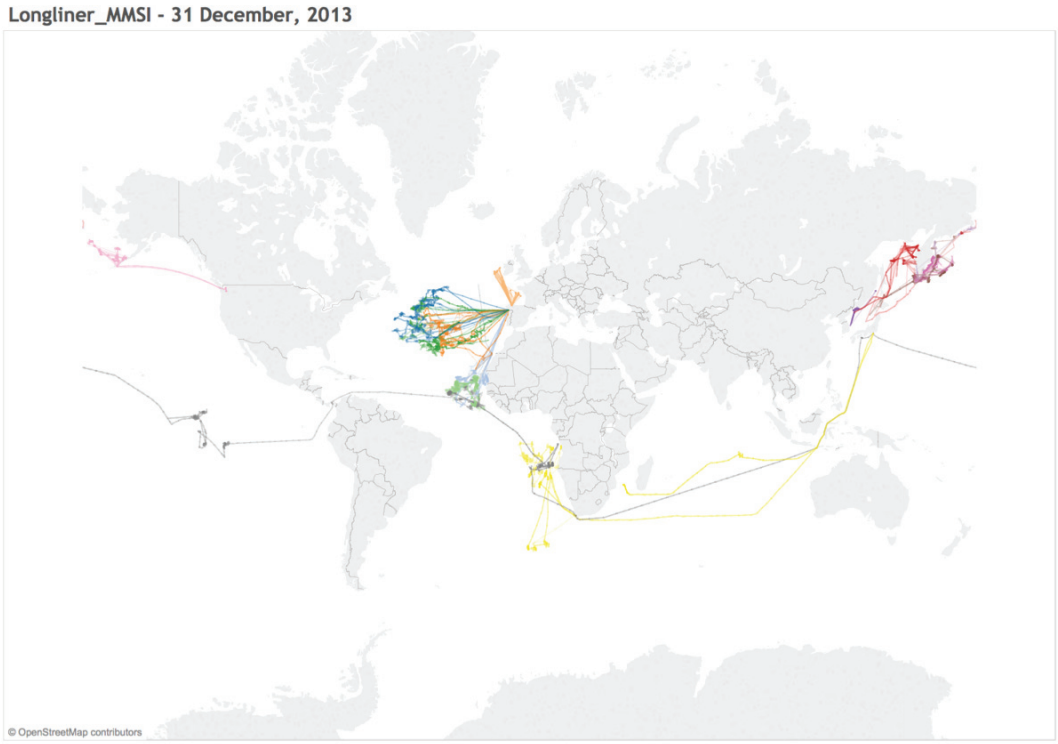


Figure 5.2: Distributions of longliners

distributions of all the trawlers. As mentioned earlier, trawler data has 84 vessels but we could not assign 84 unique colors for each vessels because of the natural limitations of our eyes. We could still use 20 colors to differentiate them as long as two neighboring vessels have different colors.

In the map to show distributions of fishing and non-fishing trajectories, we use two colors to differentiate fishing and non-fishing activities. Fig. 5.4 shows the fishing and non-fishing status among longliner trajectories. Fig. 5.5 shows the fishing and non-fishing status among trawler trajectories.

In the map to show speed distributions over trajectories, we use stepped blue colors to show variations of speed. Dark blue means high speed while light blue means low speed. In order to make light colors more readable, we choose a dark map as the background. Fig. 5.6 shows the speed distributions among longliner trajectories in a selected area.

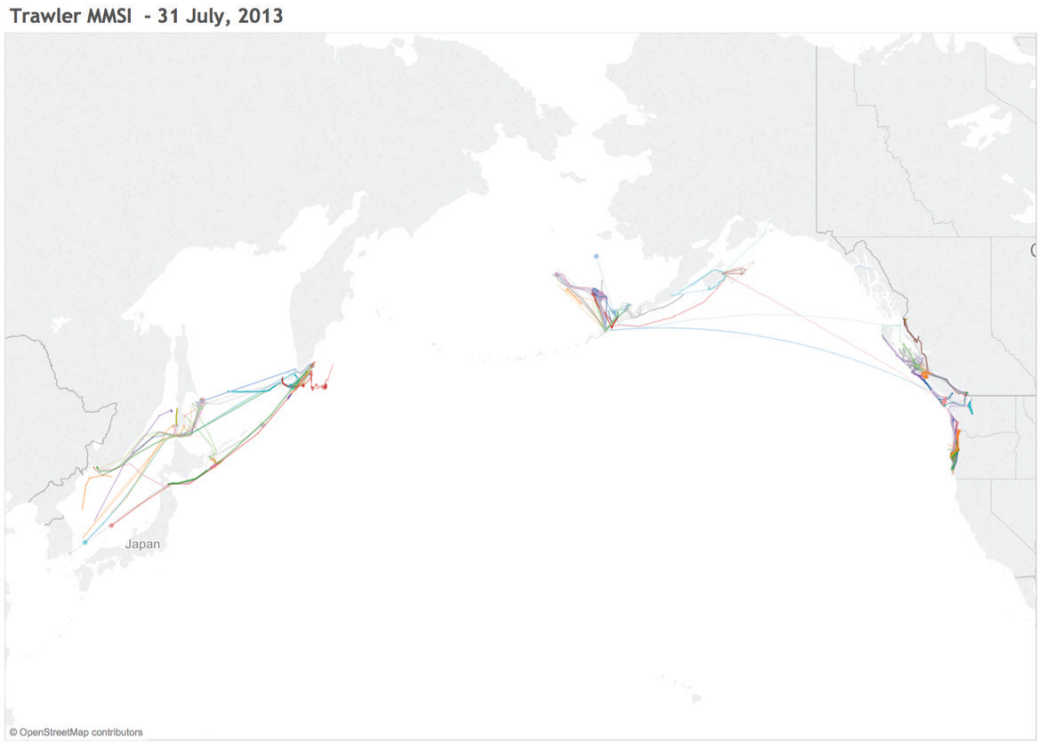


Figure 5.3: Distributions of Trawlers

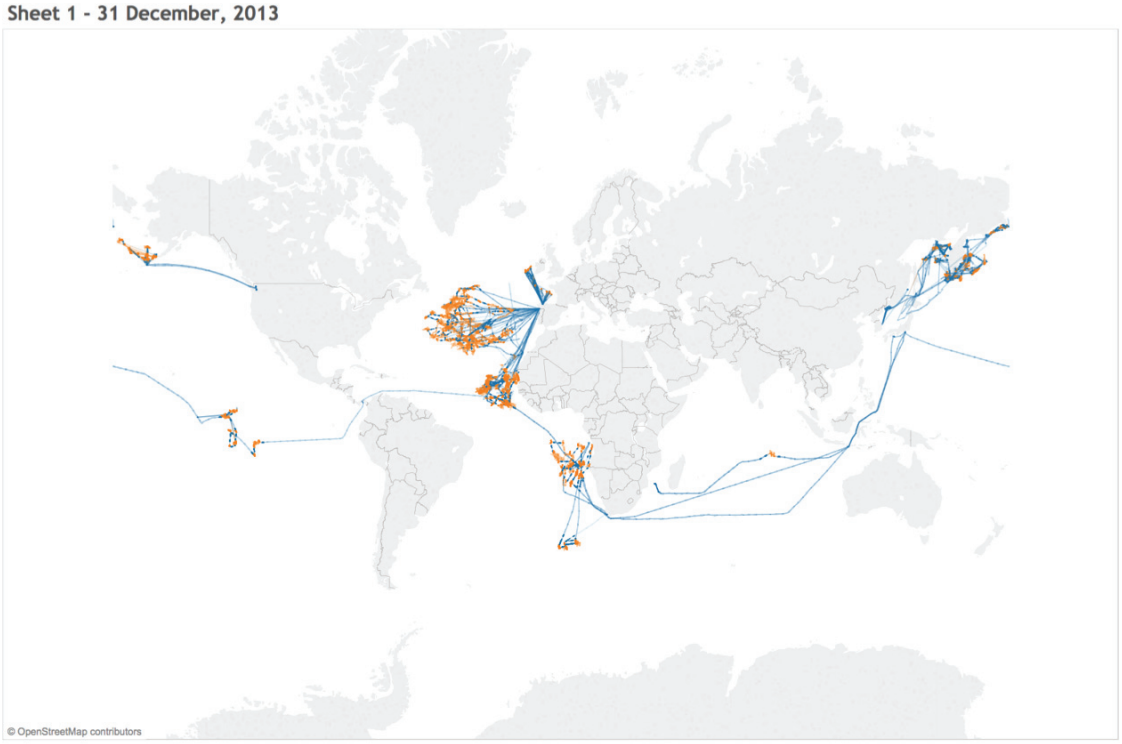


Figure 5.4: Distributions of fishing and non-fishing trajectories of Longliners

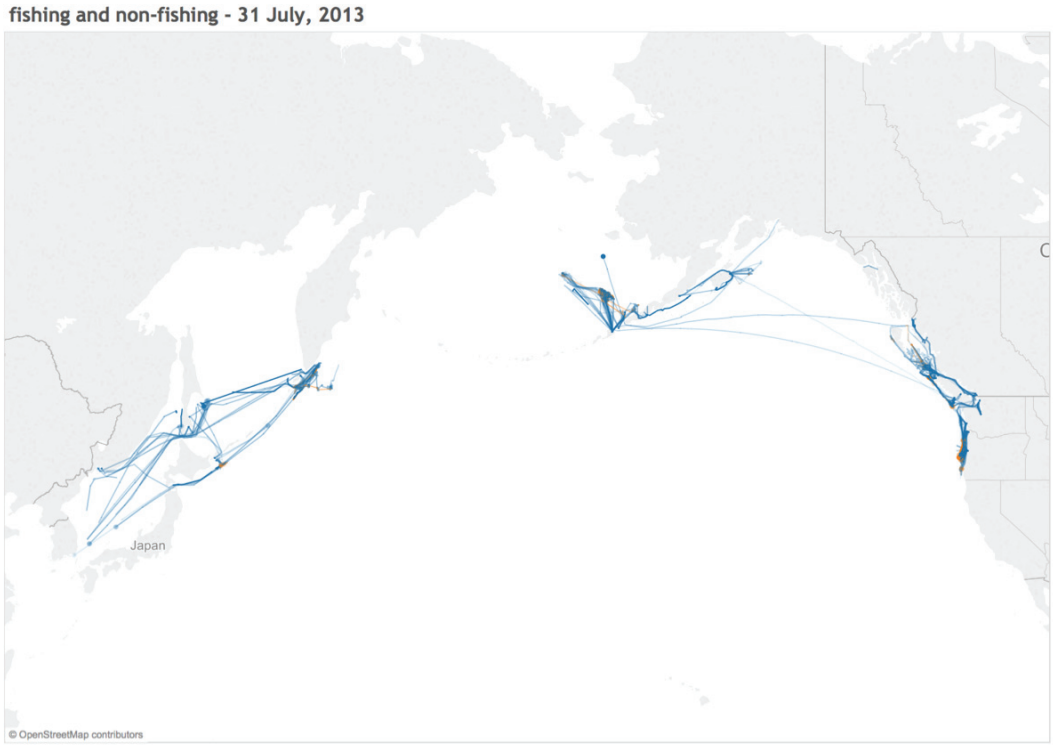


Figure 5.5: Distributions of fishing and non-fishing trajectories of trawlers

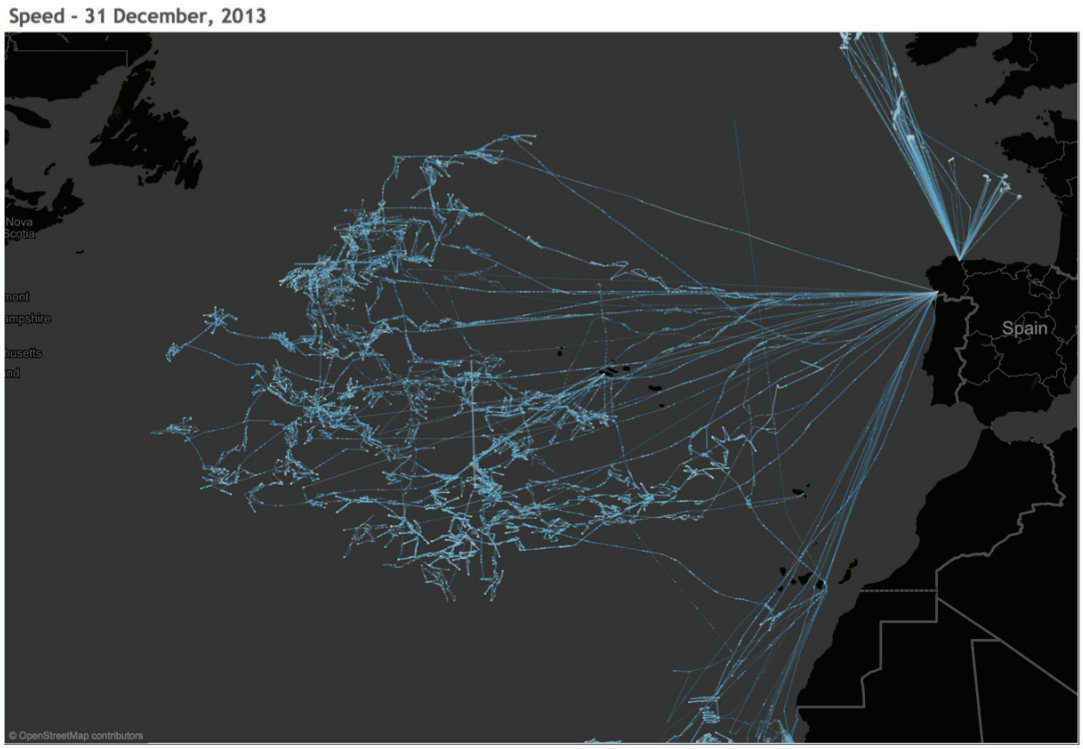


Figure 5.6: Speed distributions of longliners in selected area

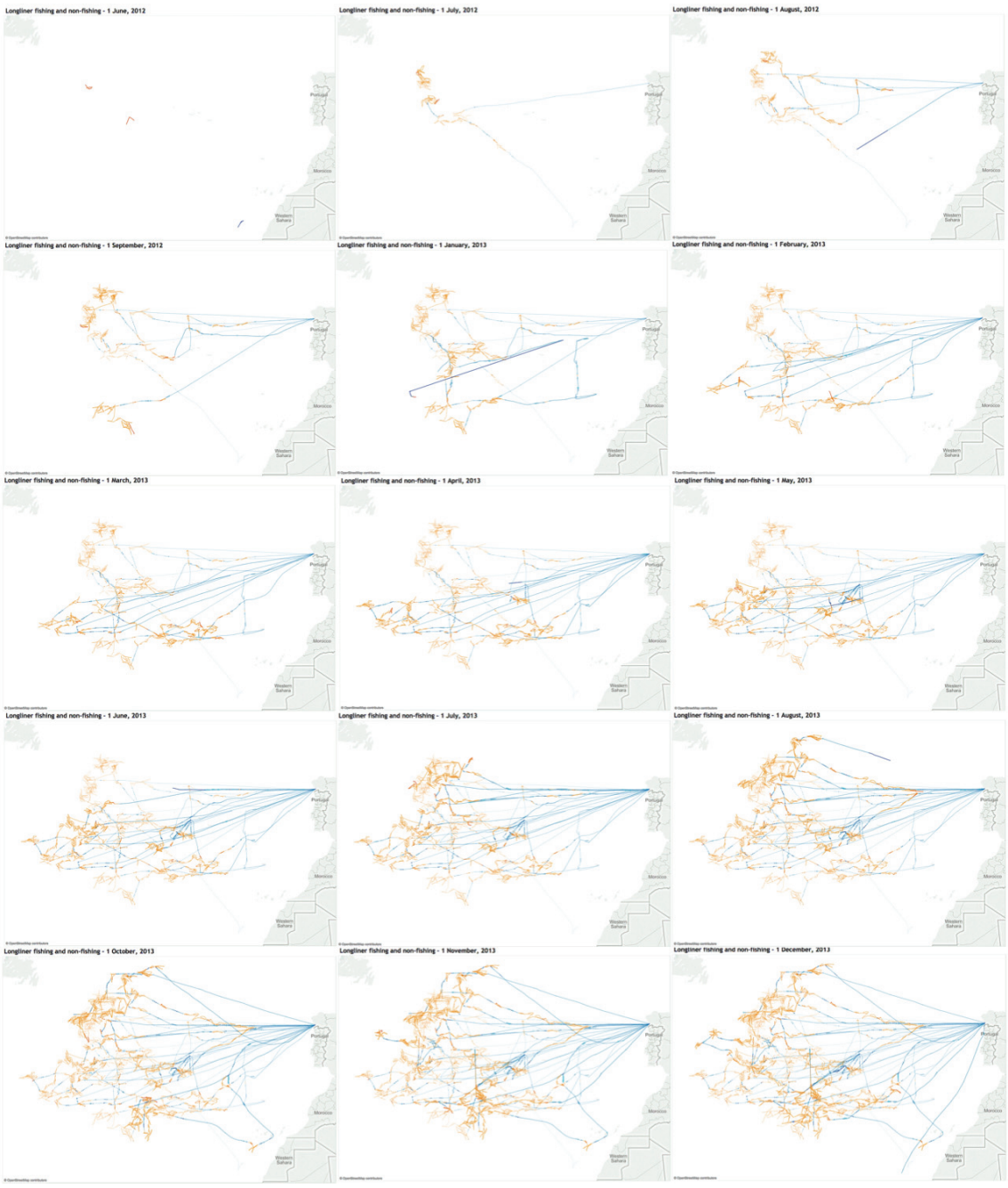


Figure 5.7: Animation of fishing and non-fishing status

5.3.2 Animated Map

We made all three kinds of maps above with animations for both trawlers and longliners. We made the historical points using a little transparency in order to highlight the newest appearing trajectories among trajectories of history. Users can specify a time period and different vessels for the animation. Fig. 5.7 shows an example of fishing and non-fishing animation of longliners with a selected time period and vessels.

5.3.3 Experimental results map

We put the results of the longliner experiments in this thesis into experimental results map. This enables users to explore the details of the results, such as in which part the model gives wrong predictions, with wrong predictions what would the trajectory then look like and in which speed of each point in the trajectory, etc. With such information, we could better analyse the model.

The classification results contains 4 parts including true positive, true negative, false positive and false negative are displayed in 4 colors. The positive class is also non-fishing the same as above experiments. True positive in blue where both predictions and labels are non-fishing, true negative in green where both predictions and labels are fishing, false positive in red where predictions are non-fishing while labels are fishing, false negative in yellow where predictions are fishing while labels is non-fishing. Users can select one or more parts to highlight and hover over the points to obtain detailed information.

Fig. 5.8 gives an example of an experimental results map in a selected area and how we analyse the results using this map. First, we are interested in where the model made incorrect classification, thus we can highlight both the false positive and false negative parts. Then we can hover the mouse over the highlighted points to explore the details of each point.

5.4 Analysis and Discussion

From exploring the experimental results maps, we have the following discoveries: the model tends to predict a sequence of fishing trajectory as a non-fishing trajectory in many cases if a number of consecutive points have speed over around 9 knots; the

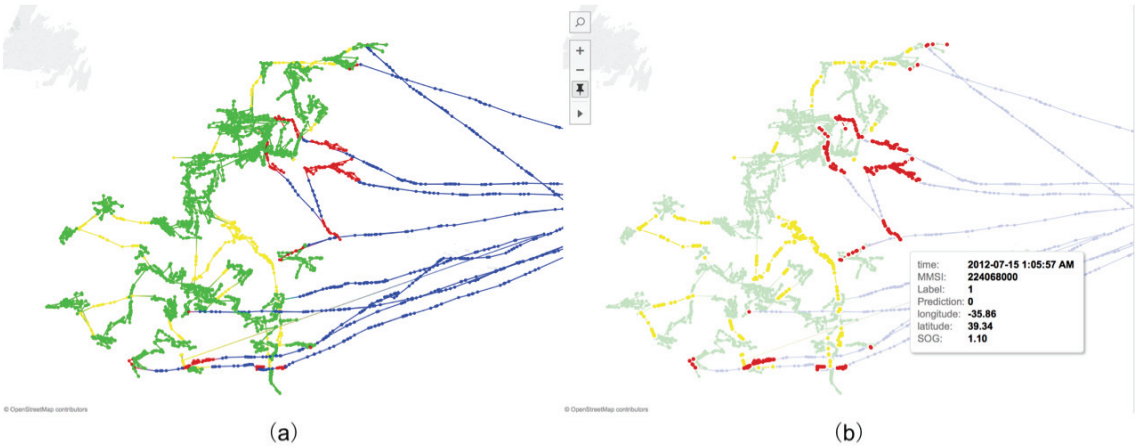


Figure 5.8: Experimental result map in a selected area. (a) is the original experimental result map, (b) is the map after highlighting false positive and false negative parts with detailed information of a point by hovering over the point.

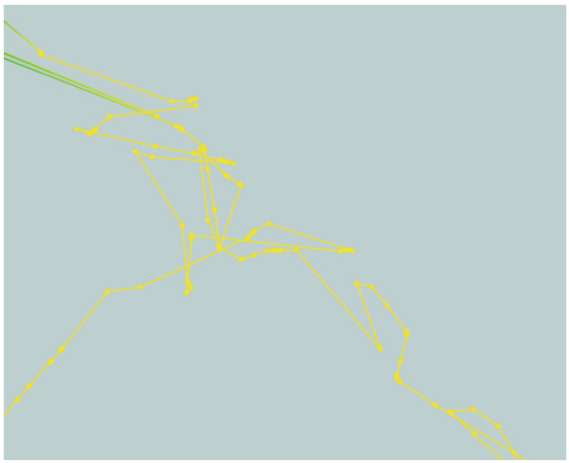


Figure 5.9: Example of non-fishing track predicted by the model as fishing track

model is not good at differentiating fishing and non-fishing parts when lots of short fishing tracks gather in a densely populated area as shown in Fig. 5.8 (a) in the left parts of the trajectories; the model could easily make wrong classification around the area where fishing tracks and non-fishing tracks have a common boundary; the model is good at differentiating non-fishing tracks with long length as shown in Fig. 5.8 (b) in the right parts of the trajectories; in some parts of the non-fishing tracks where the model gives prediction as fishing which on visual inspection clearly seems to be fishing, shown in Fig. 5.9, the yellow trajectory represents a track where the prediction is fishing while the label is non-fishing, which has a similar pattern to fishing tracks.

As shown above, this visualization could provide exploration of AIS data, however it also has some limitations. The display speed is not fast as it might take seconds to update the screen. And the animation is divided by day instead of a smaller unit because smaller units require the system to render the data more times which takes more time. Finally, the on-line version is not accessible so that it could not be published on-line without sharing the data.

Chapter 6

Conclusions and Future Work

In this thesis, we described how to detect fishing activities from historical AIS data. The proposed approach successfully applies Conditional Random Fields to the task of fishing activity detection, as a corresponding task of Part-Of-Speech tagging in natural language processing. The main contribution of our approach is that it incorporated the contextual information of both geo-location and vessel speed into features to build our model. We used differential longitude and latitude to represent geo-location. We further used distance and angle as an alternative representation of geo-location. We performed feature engineering to generate features by incorporating the spatial-temporal domain knowledge about fishing activities and developing features that can be easily classified by CRFs. This included features of the current point as well as its relationships to the neighboring points. By making use of such contextual information, our model could help provide better classification results. Since Conditional Random Fields are capable of developing robust features, our models are applicable to both trawlers and longliners.

To preprocess the data, we performed data cleansing, discretization and transformation, followed by feature selection. We then combined features that are closely related to each other by specifying proper state-observation functions. For longliners, we used feature functions that incorporated contextual information of both geoinformation and speed. For trawlers, we only used feature functions for speed. We then generated features from both state-observation and transition functions to train CRF models for longliners and trawlers, respectively.

We used three methods to evaluate the proposed model. The first evaluation metric is Modified Monte Carlo experiment which aims at obtaining an evaluation of the model on future trajectories based on historical data. This can prevent overfitting introduced by random permutation in cross-validation methods. The second evaluation metric is Iterative Leave One Batch Out which regards every vessel as a

different batch, each time the model is trained and tested with different vessels. The third evaluation metric is Stratified Leave One Batch Out which builds one model and tests it on a set of different vessels. Based on the experimental results, we made the following comparisons: first, we compared CRF models built with different sets of features; second, we compared CRF models with autoencoder and a data mining method using a segmentation algorithm for longliners; third, we compared CRF models with Hidden Markov Models for trawlers. From the experimental results, we conclude that CRFs could provide more stable and better results than autoencoder, the data mining method and HMM; our two different sets of features with different discretization strategies work identically, which prove the robustness of our discretization and feature generation methods; in terms of efficiency, we find CRFs can be trained more efficiently than complex models such as deep learning; in terms of effectiveness, the three evaluation experiments suggest the model can generalize well in future fishing activity detection problems.

To better explore the data and results, we developed an interactive visualization to help better understand the nature of fishing and non-fishing patterns and analyse the experimental results.

As for future work, we will investigate better ways of developing additional features, such as acute angle density within a certain distance. We will also consider systematic approaches to incorporate additional density information into feature functions to aid the development of the model. We will try different discretization methods. Further, we are going to apply our model to different types of fishing vessels.

Bibliography

[1] Aaron Koblin - Flight Patterns .

[2] Automatic Identification Systems (AIS).

[3] Business Intelligence and Analytics - Tableau Software .

[4] CRF++: Yet Another Toolkit .

[5] ExactEarth Looks to Government AIS, Satellite IoT to Boost Growth.

[6] Improving International Fisheries Management Report, NOAA Fisheries.

[7] Nationwide Automatic Identification System.

[8] Ship Tracking Hack Makes Tankers Vanish from View.

[9] Karl Gunnar Aarsæther and Torgeir Moan. Estimating navigation patterns from ais. *Journal of Navigation*, 62(04):587–607, 2009.

[10] David J Agnew, John Pearce, Ganapathiraju Pramod, Tom Peatman, Reg Watson, John R Beddington, and Tony J Pitcher. Estimating the worldwide extent of illegal fishing. *PloS one*, 4(2):e4570, 2009.

[11] Heather Ball. Satellite ais for dummies. *Mississauga, ON: Wiley*, 2013.

[12] Selma Belgacem, Clement Chatelain, and Thierry Paquet. A hybrid crf/hmm for one-shot gesture learning. In *Adaptive Biometric Systems*, pages 51–72. Springer, 2015.

[13] Christopher M Bishop, Julia Lasserre, et al. Generative or discriminative? getting the best of both worlds. *Bayesian statistics*, 8(8):3–24, 2007.

[14] Shuwei Che, Chunxia Meng, Jin Bai, and Wensi Wu. Mapping underwater sound noise and assessing its characteristic based on ais. In *2016 IEEE/OES China Ocean Acoustics (COA)*, pages 1–4. IEEE, 2016.

[15] Aron Culotta, David Kulp, and Andrew McCallum. Gene prediction with conditional random fields. 2005.

[16] Erico N de Souza, Kristina Boerder, Stan Matwin, and Boris Worm. Improving fishing pattern detection from satellite ais using data mining and machine learning. *PLOS ONE*, 11(7):e0158248, 2016.

[17] Bertrand Douillard, Dieter Fox, and Fabio Ramos. A spatio-temporal probabilistic model for multi-sensor object recognition. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2402–2408. IEEE, 2007.

[18] FAO Fisheries. Aquaculture department (2014) the state of world fisheries and aquaculture. *Food and Agriculture Organization of the United Nations, Rome*.

[19] Reinhard Furrer, Douglas Nychka, Stephen Sain, and Maintainer Doug Nychka. Package fields. *R Foundation for Statistical Computing, Vienna, Austria. http://www. idg. pl/mirrors/CRAN/web/packages/fields/fields. pdf (last accessed 22 December 2012)*, 2009.

[20] Laurie Goldsworthy and Brett Goldsworthy. Modelling of ship engine exhaust emissions in ports and extensive coastal waters based on terrestrial ais data–an australian case study. *Environmental Modelling & Software*, 63:45–60, 2015.

[21] Marco Guerriero, Peter Willett, Stefano Coraluppi, and Craig Carthel. Radar/ais data fusion and sar tasking for maritime surveillance. In *Information Fusion, 2008 11th International Conference on*, pages 1–5. IEEE, 2008.

[22] Asela Gunawardana, Milind Mahajan, Alex Acero, and John C Platt. Hidden conditional random fields for phone classification. In *INTERSPEECH*, pages 1117–1120, 2005.

[23] Xuming He, Richard S Zemel, and Miguel Á Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on*, volume 2, pages II–695. IEEE, 2004.

[24] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[25] Yasser Hifny and Steve Renals. Speech recognition using augmented conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(2):354–365, 2009.

[26] Xiang Jiang, Daniel L Silver, Baifan Hu, Erico N de Souza, and Stan Matwin. Fishing activity detection from ais data using autoencoders. In *Canadian Conference on Artificial Intelligence*, pages 33–39. Springer, 2016.

[27] A Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841, 2002.

[28] Daphne Koller and Nir Friedman. Probabilistic graphical models: Principles and techniques (adaptive computation and machine learning series). 2009.

[29] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

[30] Janette Lee, Andy B South, and Simon Jennings. Developing reliable, repeatable, and accessible methods to provide high-resolution estimates of fishing-effort distributions from vessel monitoring system (vms) data. *ICES Journal of Marine Science: Journal du Conseil*, page fsq010, 2010.

[31] Nicholas J Lewin-Koh, Roger Bivand, J Pebesma, Eric Archer, Adrian Baddeley, Duncan Golicher Giraudoux, Virgilio GÃ3mez Rubio, Patrick Hausmann, Karl Ove Hufthammer, Thomas Jagger, et al. Package maptools. *Internet: http://cran. r-project. org/web/packages/maptools/maptools. pdf (30.1. 2012)*, 2012.

[32] Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research*, 26(1):119–134, 2007.

[33] Bo Liu, Erico N de Souza, Cassey Hilliard, and Stan Matwin. Ship movement anomaly detection using specialized distance measures. In *Information Fusion (Fusion), 2015 18th International Conference on*, pages 1113–1120. IEEE, 2015.

[34] Bo Liu, Erico N de Souza, Stan Matwin, and Marcin Sydow. Knowledge-based clustering of ship trajectories using density-based approach. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 603–608. IEEE, 2014.

[35] Fabio Mazzarella, Michele Vespe, Dimitrios Damalas, and Giacomo Osio. Discovering vessel activities at sea using ais data: mapping of fishing footprints. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–7. IEEE, 2014.

[36] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598, 2000.

[37] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics, 2003.

[38] Ryan McDonald and Fernando Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC bioinformatics*, 6(1):1, 2005.

[39] Fabrizio Natale, Maurizio Gibin, Alfredo Alessandrini, Michele Vespe, and Anton Paulrud. Mapping fishing effort through ais data. *PloS one*, 10(6):e0130746, 2015.

[40] Giuliana Pallotta, Michele Vespe, and Karna Bryan. Traffic knowledge discovery from ais data. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1996–2003. IEEE, 2013.

[41] David Peel, Norman M Good, and Terrance Quinn II. A hidden markov model approach for determining vessel activity from vessel monitoring system data. *Canadian Journal of Fisheries and Aquatic Sciences*, 68(7):1252–1264, 2011.

[42] Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics*, page 562. Association for Computational Linguistics, 2004.

[43] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *Information processing & management*, 42(4):963–979, 2006.

[44] Nils Plath, Marc Toussaint, and Shinichi Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 817–824. ACM, 2009.

[45] Avinesh PVS and G Karthik. Part-of-speech tagging and chunking using conditional random fields and transformation based learning. *Shallow Parsing for South Asian Languages*, 21, 2007.

[46] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[47] Branko Ristic, Barbara La Scala, Mark Morelande, and Neil Gordon. Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction. In *information fusion, 2008 11th international conference on*, pages 1–7. IEEE, 2008.

[48] Tommaso Russo, Antonio Parisi, Marina Prorgi, Fabrizio Boccoli, Innocenzo Cignini, Maurizio Tordoni, and Stefano Cataudella. When behaviour reveals activity: Assigning fishing effort to métiers based on vms data using artificial neural networks. *Fisheries Research*, 111(1):53–64, 2011.

[49] Sunita Sarawagi and William W Cohen. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems*, pages 1185–1192, 2004.

[50] Kengo Sato and Yasubumi Sakakibara. Rna secondary structural alignment with conditional random fields. *Bioinformatics*, 21(suppl 2):ii237–ii242, 2005.

[51] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. Association for Computational Linguistics, 2004.

[52] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics, 2003.

[53] PAM Silveira, AP Teixeira, and C Guedes Soares. Use of ais data to characterise marine traffic patterns and ship collision risk off the coast of portugal. *The Journal of Navigation*, 66(6):879, 2013.

[54] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373, 2011.

[55] Jie Tang, Mingcai Hong, Juanzi Li, and Bangyong Liang. Tree-structured conditional random fields for semantic annotation. In *International Semantic Web Conference*, pages 640–653. Springer, 2006.

[56] Enmei Tu, Guanghao Zhang, Lily Rachmawati, Eshan Rajabally, and Guang-Bin Huang. Exploiting ais data for intelligent maritime navigation: A comprehensive survey. *arXiv preprint arXiv:1606.00981*, 2016.

[57] Youen Vermard, Etienne Rivot, Stéphanie Mahévas, Paul Marchal, and Didier Gascuel. Identifying fishing trip behaviour and estimating fishing effort from vms data using bayesian hidden markov models. *Ecological Modelling*, 221(15):1757–1769, 2010.

[58] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[59] Hanna Wallach. *Efficient training of conditional random fields*. PhD thesis, Masters thesis, University of Edinburgh, 2002.

[60] Sy Bor Wang, Ariadna Quattoni, L-P Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1521–1527. IEEE, 2006.

[61] YB Wang and Y Wang. Estimating catches with automatic identification system (ais) data: a case study of single otter trawl in zhoushan fishing ground, china. *Iranian Journal of Fisheries Sciences*, 15(1):75–90, 2016.

[62] Boris Worm, Edward B Barbier, Nicola Beaumont, J Emmett Duffy, Carl Folke, Benjamin S Halpern, Jeremy BC Jackson, Heike K Lotze, Fiorenza Micheli, Stephen R Palumbi, et al. Impacts of biodiversity loss on ocean ecosystem services. *science*, 314(5800):787–790, 2006.

[63] Feixiang Zhu. Mining ship spatial trajectory patterns from ais database for maritime surveillance. In *Emergency Management and Management Sciences (ICEMMS), 2011 2nd IEEE International Conference on*, pages 772–775. IEEE, 2011.