



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**UNDERSTANDING AUTOMATIC IDENTIFICATION
SYSTEM DATA AS APPLIED TO SOCIAL NETWORK
RELATIONSHIPS AND ACTIVITIES**

by

Rachel Cline

June 2018

Thesis Advisor:

David L. Alderson Jr.

Co-Advisor:

Ralucca Gera

Second Reader:

Samuel H. Huddleston

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 2018	3. REPORT TYPE AND DATES COVERED Master's thesis		
4. TITLE AND SUBTITLE UNDERSTANDING AUTOMATIC IDENTIFICATION SYSTEM DATA AS APPLIED TO SOCIAL NETWORK RELATIONSHIPS AND ACTIVITIES			5. FUNDING NUMBERS	
6. AUTHOR(S) Rachel Cline				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) This thesis considers the problem of how to infer a social network of entities based solely on periodic location information as they move in space and time. Specifically, we consider social networks implied by vessel traffic within the South China Sea as captured by Automatic Identification System data consisting of vessel identifier, position coordinates (latitude and longitude), heading, speed, and other information about its course. We create customized data structures in the Python programming language and implement an algorithm to quickly and efficiently create a social network of entities based on proximity parameters for both space and time. As we see from our analysis, there is no single network depiction of our data. Rather, the topology is largely influenced by the parameters we use to define a connection between entities. Because we can efficiently query the data, we quickly draw conclusions about the most active time of day and most active location within the region. From this, we are able to look into specific entities and examine the relationships, activities, and behaviors that we wish to understand.				
14. SUBJECT TERMS Automatic Identification System, network science, criminal organization, South China Sea, social network inference			15. NUMBER OF PAGES 79	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**UNDERSTANDING AUTOMATIC IDENTIFICATION SYSTEM DATA AS
APPLIED TO SOCIAL NETWORK RELATIONSHIPS AND ACTIVITIES**

Rachel Cline
Captain, United States Marine Corps
BS, U.S. Naval Academy, 2012

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

and

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

**NAVAL POSTGRADUATE SCHOOL
June 2018**

Approved by: David L. Alderson Jr.
Advisor

Ralucca Gera
Co-Advisor

Samuel H. Huddleston
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

Craig W. Rasmussen
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis considers the problem of how to infer a social network of entities based solely on periodic location information as they move in space and time. Specifically, we consider social networks implied by vessel traffic within the South China Sea as captured by Automatic Identification System data consisting of vessel identifier, position coordinates (latitude and longitude), heading, speed, and other information about its course. We create customized data structures in the Python programming language and implement an algorithm to quickly and efficiently create a social network of entities based on proximity parameters for both space and time. As we see from our analysis, there is no single network depiction of our data. Rather, the topology is largely influenced by the parameters we use to define a connection between entities. Because we can efficiently query the data, we quickly draw conclusions about the most active time of day and most active location within the region. From this, we are able to look into specific entities and examine the relationships, activities, and behaviors that we wish to understand.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Objectives	2
1.3	Thesis Structure	2
2	Background	3
2.1	AIS Data	3
2.2	AIS Focused Studies	4
2.3	Mobility Prediction	6
2.4	Network Focused Studies	7
3	Model and Methodology	13
3.1	Model Primitives	13
3.2	Network Representation	15
3.3	Data Structures	15
3.4	Proof of Concept	22
4	Results and Analysis	29
4.1	Data Processing	29
4.2	Network Exploration	30
4.3	Network Activity	37
4.4	Application: The Spratly Islands	44
4.5	Computational Efficiency	49
5	Conclusion	51
5.1	Summary of Analysis	51
5.2	Future Work	52

List of References	55
Initial Distribution List	59

List of Figures

Figure 3.1	Vessel Activity Spectrum.	14
Figure 3.2	Class Attribute Associations	17
Figure 3.3	Geohash Precision Example	18
Figure 3.4	Example Trie Configuration	21
Figure 3.5	Example Connections Networks	25
Figure 3.6	Results of Time Query	26
Figure 3.7	Map Plot of Entity Track Query	27
Figure 4.1	South China Sea Region of Interest for AIS Data.	29
Figure 4.2	Networks with Timehash Precision = 10	31
Figure 4.3	Networks with Timehash Precision = 9	32
Figure 4.4	Networks with Timehash Precision = 8	33
Figure 4.5	Networks with Geohash Precision = 8	34
Figure 4.6	Networks with Geohash Precision = 7	35
Figure 4.7	Networks with Geohash Precision = 6	36
Figure 4.8	Visited Locations for Central Nodes	39
Figure 4.9	Network Heatmaps Showing Regional Activity	40
Figure 4.10	Port Locations.	41
Figure 4.11	Histogram of Network Activity.	42
Figure 4.12	Network Heatmaps Showing Active Locations for a Given Time .	43
Figure 4.13	Heatmap of Reef Activity.	45

Figure 4.14	Network of Entities Visiting Point of Interest (POI) in the South China Sea.	48
Figure 4.15	Network of 2 Entities of Interest	48

List of Tables

Table 2.1	Preview of AIS Data	4
Table 3.1	Subset of AIS Data	23
Table 3.2	Results of Location Query	23
Table 3.3	Results of Connections Query with Geohash Precision = 7	24
Table 3.4	Results of Connections Query with Geohash Precision = 6	24
Table 3.5	Results of Time Query with Timehash Precision = 6	25
Table 3.6	Data Results of Entity Track Query	27
Table 4.1	Network Characteristics for Timehash Precision = 10	31
Table 4.2	Network Characteristics for Timehash Precision = 9	32
Table 4.3	Network Characteristics for Timehash Precision = 8	33
Table 4.4	Network Characteristics for Geohash Precision = 8	34
Table 4.5	Network Characteristics for Geohash Precision = 7	35
Table 4.6	Network Characteristics for Geohash Precision = 6	36
Table 4.7	Breadth-First Search (BFS) Networks Starting with Maritime Mobile Service Identity (MMSI) Number 600010885	37
Table 4.8	Geohash and Timehash Precision Breakdown	38
Table 4.9	Degree Centrality for BFS Networks	38
Table 4.10	Time Query for Most Active Hour	42
Table 4.11	Average Hourly Activity throughout the Day	44
Table 4.12	Unique Entities Visited at each Reef	45
Table 4.13	Total Times each Reef Visited	46

Table 4.14	The Most Active Entities at Mischief Reef	46
Table 4.15	Entities that have Visited 6 Reefs	47
Table 4.16	Entities that have Visited 5 Reefs	47
Table 4.17	Network Characteristics of Reef Visitors	47
Table 4.18	Computational Efficiency of Two Different Techniques	50

List of Acronyms and Abbreviations

AIS	Automatic Identification System
BFS	Breadth-First Search
COG	Course Over Ground
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DOD	Department of Defense
ETA	Estimated Time of Arrival
FCM	Fuzzy C-Means
FIFO	First In First Out
GPS	Global Positioning System
HMM	Hidden Markov Model
HON	Higher Order Network
IC	Intelligence Community
IMO	International Maritime Organization
k-NN	k-Nearest-Neighbor
MID	Maritime Identification Digits
MMC	Mobility Markov Chain
MMSI	Maritime Mobile Service Identity
NPS	Naval Postgraduate School
PFS	Priority-First Search

PMM	Periodic Mobility Model
POI	Point of Interest
pRNN	Partition-wise Recurrent Neural Network
RNN	Recurrent Neural Network
SOG	Speed Over Ground
SOLAS	Safety of Life at Sea
STF-RNN	Space Time Features-based Recurrent Neural Network
TREAD	Traffic Route Extraction and Anomaly Detection
UTC	Coordinated Universal Time
VCRO	Vessel Conflict Ranking Operator

Executive Summary

The intelligence community is interested in developing quantitative techniques to uncover social networks and understand their relationships, activities, and behaviors. This is of particular interest when studying criminal organizations such as terrorist groups or drug cartels. In order to effectively combat these groups, the intelligence community must understand how these networks are organized and how they operate. The prevalence of digital devices used in today's society, along with the capability to capture and store these signals, allows us to examine interactions within these entities as indicated by their digital signature.

This thesis focuses on social networks implied by vessel traffic within the South China Sea as captured by Automatic Identification System data—consisting of vessel identifier, position coordinates (latitude and longitude), heading, speed, and other information about its course. More generally, we can think of a vessel as an *entity* that moves in both space and time, and that periodically emits a signal as it moves. Because these signals (and their recorded positions) are discrete, the actual continuous movement can only be inferred. Among other things, our interest is understanding where an entity travels, what locations it visits, and with what other entities it interacts. The ability to quickly infer interactions among entities makes it possible to rapidly generate social networks that further lend themselves to various forms of intelligence analysis.

The technical contributions in this thesis are primarily computational. We create customized data structures in the Python programming language and implement an algorithm to quickly and efficiently create a social network of entities based on proximity parameters for both space and time. This organization significantly decreases the computational time required to build and explore a social network. For example, on a subset of 343K observations, our technique is more than 500 times faster than a naive brute-force technique. On a subset of 1.2 million observations, our technique is more than 1000 times faster (i.e., requiring less than a minute vice more than 20 hours).

The data set we study contains nearly 6 million observations of vessel traffic within the South China Sea. Due to the speed at which we are able to query the data, we quickly

draw conclusions about the most active time of day and most active location within the region. From this, we are able to look into specific entities and examine their network to see which entity is potentially most influential. In addition, we infer social networks based on visitors to specified points-of-interest. These queries allow us to gain insight into some of the relationships, activities, and behaviors that we wish to understand.

By varying the space and time parameters to our algorithm, we infer ten different social networks and analyze the connectivity between different entities. Because Automatic Identification System data is used for collision avoidance, very few entities are seen close together. Vessels, unlike human interactions where individuals are often seen within feet of each other, have a much larger threshold for distance. As we see from our analysis, there is no single network depiction of our data. Rather, the topology is largely influenced by the parameters we use to define a connection between entities. Our analysis uncovers differences in topology and activities of the different networks.

We examine nine reefs within the South China Sea as specific points-of-interest and infer a social network based on visitors of those reefs. The results of this query allow us to uncover the most active reef within the region and the most active entities within the network. We are able to uncover the activity levels by both number of observations and number of unique visitors. The results between these two queries match for the most active reef but vary for the other reefs. Since the ranking of most active reef depends on the query used, intelligence analysts must understand which information is most important for answering their question and focus their efforts on the most relevant data.

This thesis creates many opportunities for additional work. In particular, it will be increasingly relevant to understand the data the intelligence analyst requires and provide an efficient way to extract the information. Extending the computational techniques described here to help predict future movements or interactions among entities will be a particularly important topic of future research.

Acknowledgments

First and foremost, I would like to thank my family and friends for all of their love and support throughout my time at the Naval Postgraduate School. In particular, I would like to thank my sister, Christy, for all of the time she spent tutoring me and editing my essays. I could not have done this without your help.

To the faculty and staff, I want to thank you for your dedication and willingness to mentor both inside and outside of the classroom. It was an honor learning from you.

Finally, to my advisors, I cannot thank you enough for your mentorship and guidance. I truly appreciate all the hours you dedicated to helping me throughout my research. Your insights and knowledge were invaluable and I learned a lot from working with you. Thank you.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:

Introduction

1.1 Motivation

Recent advances in information technologies and communication systems have enabled the collection and broadcast of real-time positional data for a variety of devices, ranging from mobile phones, to cameras, to vehicles, to weapon systems. Even personal fitness trackers designed to log individual exercise can trace out the locations and movement of entire populations (Hern 2018).

The Intelligence Community (IC) is keenly interested in tracking the movement of goods and people over space and time, as well as understanding the social network relationships among individuals in terrorist groups, criminal organizations, and other entities who are adversarial to the United States. To this end, the IC has ongoing need to develop quantitative techniques that can uncover social networks and understand interactions between entities within the network. For example, the IC looks to gain an understanding of how the network evolves over time and to uncover which entities within the network are the key players. One way to combat these groups is to target key individuals or assets, and this requires an understanding of how they are organized and how they function.

Creating a social network representation of an illicit organization is a challenging task. Because the entities involved tend to be secretive about their relationships and their activities, it is often useful to infer relationships and activities based on their locations. The prevalence of digital devices that record or broadcast location information creates the possibility of new data-driven approaches to social network inference.

Within the maritime domain, sea vessels broadcast location data through their use of the Automatic Identification System (AIS), which tracks vessel location for collision avoidance and passenger safety. The use of AIS is required by the International Maritime Organization (IMO) as set forth by Regulation 19 of the Safety of Life at Sea (SOLAS) Chapter V (SOLAS 2018).

1.2 Research Objectives

In this thesis, we study the South China Sea region and look at the social network of organizations implied by vessel traffic. We represent connections between vessels based on both space and time. This information helps us gain a better understanding of network behavior within the region and focus on specific areas of interest to the military.

With the increase in tensions in the South China Sea over China's buildup of military bases, we examine which vessels are potential key players within the region. This problem is important the U.S. Navy and Department of Defense (DOD) specifically. However, our study of interactions between vessels is also representative of a broader class of problems using data-driven techniques for inferring social network structure and activity.

1.3 Thesis Structure

This thesis proceeds as follows. In Chapter 2 we review terminology and the current literature related to pattern prediction and anomaly detection for both maritime traffic and social networks. In Chapter 3 we describe the AIS data of the South China Sea, the process we use to prepare the data, the data structures we use, and the algorithms we create. In Chapter 4 we present the results of our analysis and our inferences. Finally, in Chapter 5 we present our conclusions and research areas for future work.

CHAPTER 2: Background

This chapter provides basic background information on AIS data and reviews the current literature for both data-specific studies as well as network-related studies. At the end of this chapter we introduce terms relating to this thesis.

2.1 AIS Data

AIS data consists of two different data types: static and dynamic. Dynamic data consists of the Maritime Mobile Service Identity (MMSI) number, AIS navigational status, rate of turn, speed over ground Speed Over Ground (SOG), position coordinates (latitude and longitude), course over ground Course Over Ground (COG), heading, bearing at own position, and coordinated universal time Coordinated Universal Time (UTC) seconds. Dynamic data is transmitted every 2 to 10 seconds when a vessel is underway and every 6 minutes when a vessel is anchored Marine Traffic (2018).

Static AIS data consists of the IMO identification number, call sign, vessel name, vessel type, dimensions, location of the positioning system's antenna, type of positioning system, draught, destination, and Estimated Time of Arrival (ETA) (see Marine Traffic 2018). Static data is transmitted every 6 minutes and is updated by the vessel's crew. In this thesis, we use vessel name, MMSI, IMO identification number, vessel type, dimensions, position coordinates, SOG, and UTC.

The next few paragraphs describe in more detail the data of particular interest in this thesis.

A vessel's MMSI is a unique, nine digit number that is given to each vessel, and its owner, required to broadcast AIS messages. According to the IMO, "the regulation requires AIS to be fitted aboard all ships of 300 gross tonnage and upwards engaged on international voyages, cargo ships of 500 gross tonnage and upwards not engaged on international voyages and all passenger ships irrespective of size" (IMO 2018a). The first three digits of the MMSI number are the Maritime Identification Digits (MID). These digits identify the vessel's country of origin and can range from 201 to 775. If a vessel changes owners, the MMSI

number changes to reflect this new, unique vessel-owner relationship.

The IMO identification number is a seven digit number assigned to a vessel. Unlike the MMSI, the IMO identification number is a permanent number assigned to a vessel (IMO 2018b), regardless if the vessel exchanges owners.

Vessels are classified by their type, which indicates the function it performs. It is usually a two digit number where the first digit indicates the general category of the vessel – tanker, cargo, passenger. The second digit, in certain circumstances, is used to indicate the cargo carried by the vessel: hazard, major hazard, minor hazard, and recognizable hazard (Marine Traffic 2018).

A vessel's SOG is the speed of a vessel relative to the surface of the earth. It indicates how fast the vessel travels, taking into consideration the effects of both wind and current. If a vessel is traveling at a speed of 15 knots, but the current is moving in the opposite direction at 2 knots, then the vessel's SOG is 13 knots, not 15 knots.

Finally, the UTC describes the year-month-day and the hour-minutes-seconds for each observation.

Table 2.1 shows a small sample of the data we use in this thesis, where Epoch Time is the UTC, or the number of seconds that have elapsed since the epoch as defined by the time elapsed since 1 January 1970.

Table 2.1. Preview of AIS Data

Vessel Name	MMSI	IMO	Call Sign	Type	Length	Lat	Long	SOG	COG	Heading	Epoch Time
Swissco Summit	566077000	9616462	9V8428	52	41	5.108167	114.341833	0.9	86	282	1415036156
Stellar Express	353787000	8915249	3EGZ8	7	209	8.464513	108.237748	12.9	232.4	235	1415036166
Neopetro 7	533000746	9495882	9WHW8	9	34	4.69817	112.481272	0.2	247.6	245	1415036232
Wg Vespucci	209128000	9452971	5BXF2	38	90	6.139515	109.862027	3.8	156.2	162	1415036278
Bahtera Hadar	533130248	9307449	9WNH5	7	73	5.583088	114.927192	0.2	280.5	111	1415036353
Densa Shark	215250000	9607681	9HA2687	7	292	7.921812	108.455528	12.4	211.3	214	1415036363
Jin Hai Ou	413911000	9125578	BVKX	7	190	9.634355	109.92851	11.8	209.5	200	1415036369
Rui Da Star	354645000	8407759	3EEJ3	7	97	10.598895	107.022935	0.5	14.1	338	1415036395
Maritime Century	357361000	9574133	3FWO9	7	235	8.364228	109.820552	14.2	30.7	37	1415036414
Oocl Charleston	477655800	9461790	VRFX2	7	260	9.148467	109.455767	12.1	35	41	1415036503

2.2 AIS Focused Studies

Due to the frequency that AIS data is transmitted, it is estimated that there are over 520,000,000 observations world-wide per day (Marine Traffic Blog 2018). As a result

of this quantity of data, an extensive amount of research has been conducted on AIS data sets.

One area of research involving AIS data is anomaly detection in movement patterns. Ristic et al. (2008) introduce adaptive kernel density estimation as a means to predict movement of maritime traffic within ports and waterways. The authors specifically look at motion anomaly detection and motion prediction using AIS data. In another study, Pallotta et al. (2013) adopt a methodology called Traffic Route Extraction and Anomaly Detection (TREAD) to classify routes and detect both “rule-based” and “low-likelihood” anomalies. One of the benefits of TREAD is that it stores the detected historical routes in an atlas, making the data easily accessible. This methodology improves on previous work by allowing analysis of maritime traffic trajectories of unequal lengths. In addition to anomaly detection, the authors use what is known as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to efficiently differentiate “waypoints,” such as entry and exit points, that are too close for normal algorithms to distinguish. DBSCAN is a more effective clustering algorithm as it does not require prior information on the number of clusters in which to group the data.

Since DBSCAN is an efficient clustering algorithm, Mazzearella et al. (2014) make use of DBSCAN to cluster fishing locations. The authors use unsupervised learning to extract maritime traffic routes and stop points, where fishing locations are the stop points of interest. Although the authors focus on discovering fishing locations, their methodology can be modified to illuminate other maritime activities. While DBSCAN is versatile and adaptable, other density-based clustering algorithms exist. Shao et al. (2014) introduce a clustering technique called Fuzzy C-Means (FCM). The authors use what is known as “Fuzzy methods” to correlate (with the k-Nearest-Neighbor (k-NN) algorithm) and associate (with the FCM clustering algorithm) maritime traffic routes. These methods provide accurate results for both correlation and association with Global Positioning System (GPS) and Radar contacts.

In addition to anomaly detection and clustering, AIS data is used to analyze maritime traffic density and identify hazardous regions. Silveira et al. (2013) develop an algorithm to classify which ports and waterways are prone to accidents. They implement the model by Pedersen et al. (1995) as a means to quantify the number of high-risk vessels within a region

and compare their output to historical data. Zhang et al. (2015) introduce the Vessel Conflict Ranking Operator (VCRO), which is the maritime version of the “traffic conflict technique” used to classify accidents or near-accidents in road transportation. It ranks risk levels of maritime traffic encounters as a means to highlight near-miss events. The VCRO proves promising as a real-time detector of potentially catastrophic events. However, one shortfall of this model is that it does not take into account the size of a vessel or its maneuverability. This hinders the VCRO from providing the most accurate ranking results.

Finally, Aarsæther and Moan (2009) use computer vision techniques to explore patterns in maritime traffic. They represent the data based on three conceptual types: times-series, traffic-group, and area-traffic. They consider time-series data to be simply a single vessel’s data in chronological order. The last two data types are similar, but on different scales. Traffic-group data is considered to be a group of vessels traveling along similar routes in the same direction, whereas area-traffic is specific to a region and consists of sea-lanes. Computer vision techniques work well in finding maritime patterns in open-waterways but prove inconsistent when in congested ports or sea-lanes.

2.3 Mobility Prediction

With improved technology comes the increased ability to study how individuals or groups move and behave. According to Cho et al. (2011), “accurate models of human mobility are essential for urban planning, understanding human migration patterns, and spread of diseases.” As such, we introduce a few studies on mobility prediction in the following paragraphs.

In his thesis, Young (2017) uses statistical techniques to predict a vessel’s future location. Vessel routes are clustered as a means to group similar routes into a single trajectory. The results show high predictive accuracy when using a random forest technique, but prove less accurate when using neural networks for classification.

Cho et al. (2011) use check-in data from websites such as Gowalla (2010) and Brightkite (2010) to understand human mobility based off of ties to social networks. Their model consists of spatial locations that members commonly visit, the time it takes to traverse between locations, and a model of the movement. The model, known as the Periodic

Mobility Model (PMM), significantly improves on the Gaussian Model.

Another method to predict where an individual is most likely to visit is introduced by Ye et al. (2013). The authors introduce a Hidden Markov Model (HMM) as a means to model movement patterns. The HMM takes into account not only the states and transition probabilities of a typical Markov model, but also makes use of a “state-dependent output probability.” The authors build upon the HMM with a Mixed HMM that learns through what is known as “Markov Chain Monte Carlo Bayes Estimation.”

In addition to the HMM, Gambs et al. (2012) use what is called a Mobility Markov Chain (MMC) to model the mobility behavior of an individual. The MMC follows closely to a Markov model with a set of states and a set of transition probabilities. For the MMC, the set of states are labeled as specific locations such as “work” or “home.” The transitions indicate the probability of going from one state to the next, such as from home to work or vice versa. By utilizing the MMC, the authors note they can predict the next state with an accuracy of 70% to 95%.

2.4 Network Focused Studies

Networks are all around us. Everyday we use some form of transportation network, communications network, or computer network. We form social networks with our family, friends, co-workers, and peers. Nations build defense networks, society creates financial networks, and scientists study disease networks. This is hardly an exhaustive list of networks, but should serve to introduce the importance networks play in our lives.

For completeness, we provide basic definitions and introduce some of the current literature on networks and the different forms of networks implemented in the studies.

2.4.1 Definitions

Networks (graphs) consist of nodes (or vertices) and arcs (or edges), where nodes are the individual points within the network and the arcs are the connections between the nodes in the network. Specifically, Newman (2010) defines a network as follows.

Definition 2.4.1 *Network*

A network is, in its simplest form, a collection of points joined together in pairs by lines ... A network is a simplified representation that reduces a system to an abstract structure capturing only the basics of connection patterns and little else.

On the other hand, Bollobás (1998) defines a graph as follows.

Definition 2.4.2 *Graph*

A graph is an ordered pair of finite disjoint sets (V, E) such that E is a subset of the set V^2 of unordered pairs of V . The set V is the set of vertices and E is the set of edges. If G is a graph, then $V = V(G)$ is the vertex set of G , and $E = E(G)$ is the edge set. An edge $\{x, y\}$ is said to join the vertices x and y and is denoted by xy . Thus xy and yx means exactly the same edge; the vertices x and y are the end vertices of this edge.

In the context of maritime traffic and AIS data, we are interested in finding connections between vessels and understanding how the network evolves over time. Newman (2003) defines a social network as follows.

Definition 2.4.3 *Social Network*

A social network is a set of people or groups of people with some pattern of contacts or interactions between them.

We look at connections in the network by examining which entities are connected to each other. Ahuja et al. (1993) defines network connectivity as follows.

Definition 2.4.4 *Connectivity*

Two nodes i and j are connected if the graph contains at least one path from node i to node j . A graph is connected if every pair of its nodes is connected.

In order to understand how to create a network, we must understand what a connection is between entities. Ahuja et al. (1993) describe these connections as adjacency's. We think of adjacent nodes as **neighbors**.

Definition 2.4.5 *Adjacency*

Let N be the set of nodes and A the set of arcs. The node adjacency list $A(i)$ is the set of nodes adjacent to that node; in this case, $A(i) = \{j \in N: (i, j) \in A\}$.

Hagberg et al. (2008) and Bastian et al. (2009) describe several network characteristics that we use to compare different networks. These characteristics describe the topology of the network and help us understand its composition.

Definition 2.4.6 *Average Path Length*

A network's average path length is the average shortest path between all pairs of nodes in a connected network. Mathematically, it is defined as follows:

$$a = \sum_{s,t \in V} \frac{d(s,t)}{n(n-1)}$$

where V is the set of nodes in the network, $d(s,t)$ is the shortest path from s to t , and n is the number of nodes in the network.

Definition 2.4.7 *K-Core*

A K-Core is a subset of the network where all nodes are connected to at least k other nodes within the network.

Definition 2.4.8 *Community*

A community is a cluster of nodes within a network.

Definition 2.4.9 *Central*

A central node is a node within a network that is most often on the average shortest path, as indicated by the average path length.

Definition 2.4.10 *Degree Centrality*

The fraction of nodes an entity is connected to.

Our specific interest is understanding how the network structure inherent in AIS data reveals insight into the organization and behavior of the system. We build upon the following prior work.

2.4.2 Network Studies

Xu et al. (2016) use shipping traffic networks to analyze what is called a Higher Order Network (HON). A HON allows the authors to understand movement through the network. Like Markov models, a first-order network forgets the past. However, a HON retains previous history, which allows for higher accuracy when predicting next port of visit. The authors introduce two steps in constructing a HON, rule-extraction and network wiring. The role rule-extraction plays in the model is to identify the level of dependencies in the network. Network wiring then converts these rules to a network representation. The results show that compared to a first-order network, a HON doubles the accuracy in predicting a ship’s next port of visit (Xu et al. 2016).

A more abstract use of networks is introduced in the study by Jiang et al. (2017), who use a more sophisticated version of the Recurrent Neural Network (RNN), “a family of neural networks for processing sequential data” (Goodfellow et al. 2016), seen in typical machine-learning applications. In their study, Jiang et al. (2017) focus on point-based AIS

trajectory classification involving vessel trajectories (i.e., a sequence of locations over time) to predict fishing patterns. Due to the low-dimensionality and diversity of AIS data, the authors note that creating deep neural networks on the data proves cumbersome. As a result, they introduce the Partition-wise Recurrent Neural Network (pRNN), which partitions the data into a hierarchical and multi-layer structure. Their research shows an improvement at the 95% confidence level as compared to typical RNNs.

In addition to the pRNN, Al-Molegi et al. (2016) introduce another improvement of the RNN, using what they call a Space Time Features-based Recurrent Neural Network (STF-RNN) for predicting an individual's next location. The authors use data obtained from GPS-enabled mobile devices to analyze mobility patterns. The STF-RNN is a four-layer model. The first layer is an input layer that consists of two, 1-of- P encoded vectors, where P is the number of elements in the vector. The second layer is a lookup layer, which converts the two 1-of- P vectors into a real value vector for analyzing meaningful interactions within the data. The third layer is a hidden layer that holds the history of an individual's movement. Finally, the last layer in the model is the output layer. This layer outputs the probability distribution for the points of interest and is used to predict an individual's next location. The results of the STF-RNN indicate that it outperforms a RNN by nearly 10%.

As opposed to using a network to analyze the data, Xu and Chen (2004) use link analysis on criminal networks to identify individuals of interest within the network. Link analysis implements two separate shortest-path algorithms: a Priority-First Search (PFS) and a two-tree PFS. The first algorithm is a modification of Dijkstra's algorithm (Dijkstra 1959) that improves performance by implementing an efficient priority queue. The two-tree PFS is the same as the two-tree Dijkstra's algorithm (Dijkstra 1959). One tree grows by looking at its outgoing nodes while the other tree looks at its incoming nodes. Using link analysis on criminal networks, the authors identify the most-notable associations between individuals within the network.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:

Model and Methodology

This chapter introduces our lexicon, describes key entities of interest, and then details our methodology for transforming AIS data into a network-based representation suitable for analysis. Finally, we describe our network analysis techniques and visualization of results.

3.1 Model Primitives

In general, we can think of a vessel as an *entity* that moves in both space and time. Periodically, an entity emits a signal, termed here a *ping*, announcing its position in space at a given moment in time. The interval between successive pings can be regular or irregular.

A *track* is a sequence of pings that connects the positions of an entity over time; this specifies the movement of the entity. We assume the movement of a vessel is continuous in space and time. However, because the recorded positions in a track are discrete, the actual continuous movement can only be inferred.

A port, fishing location, or other position or area of interest is considered a *Point of Interest (POI)*. By our definition, a POI is stationary (i.e., it does not move over time) and is described by a designated coordinate or set of coordinates (e.g., an area).

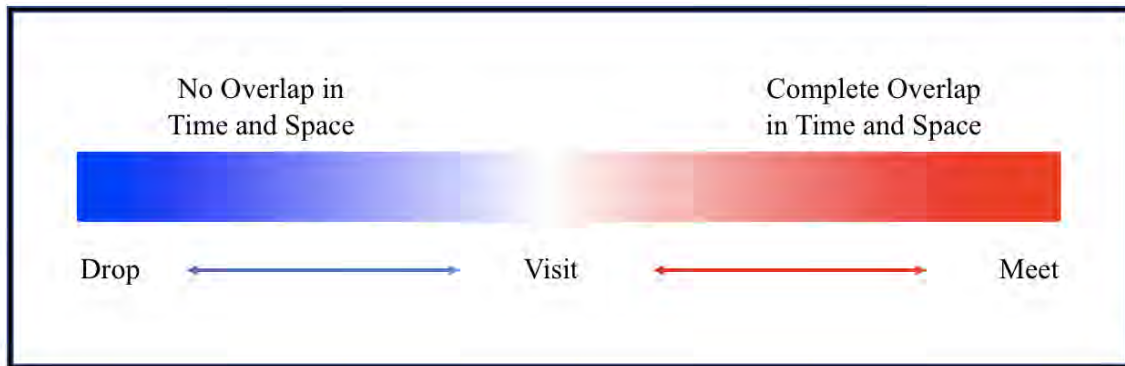
Our overall interest is in understanding the behavior of entities and the potential relationships between them, as evidenced by their tracks. To this end, we define the following additional model primitives.

- If an entity remains in the same position for an extended period of time, we refer to this as a *stop*.
- If two entities are at the same position at the same time, we refer to this as a potential *meet*, suggesting that they might have directly engaged with one another.
- If two entities are at the same position, but at different times (within some designated time window), we refer to this as a potential *drop*, suggesting that they might have indirectly engaged with one another (e.g., by “dropping” material that could later be picked up).

- If an entity passes through a POI, or position, than we refer to this as a *visit*. A visit does not account for the length of time an entity is at a POI, but merely accounts for whether an entity visits or not.

Due to the ambiguous nature of a drop, visit, and meet, we create Figure 3.1, which summarizes the characteristics associated with a drop, visit, and meet. It addresses the ambiguities associated with these terms and provides a visual representation of these abstract concepts.

Figure 3.1. Vessel Activity Spectrum.



On the far left is a *drop*, where there is no overlap between vessels in a given location. In the middle is a *visit*, where there is little to differentiate it from a drop. On the right is a *meet*, where there is overlap between vessels in a given location. The difference between a visit and a meet is also ambiguous and dictated by circumstances.

A natural way to represent the relationships between entities is to infer a social network based on their movement, stops, and potential meets or drops. We describe our approach to constructing these networks in the next section.

3.2 Network Representation

Given our model primitives, we seek to answer four broad categories of questions.

1. *Given an entity, which locations does it visit and at what times?* This question is largely answered by the recorded track for the entity, subject to ambiguities in the discrete nature of recorded pings.
2. *Given a location, which entities have visited?* The answer to this question depends on the given window of time that we are interested in allowing a visit to occur.
3. *Given a window of time, which locations are visited?* The answer to this question depends on the proximity to the location, that is, how close an entity must come in order to consider it a visit. In particular, we are interested in understanding which locations are most active during the time-period.
4. *Given an entity, with whom is it “connected”?* We define *connection* in terms of a drop, meet, or visit.

This final question, in particular, suggests the utility of organizing our AIS data as a *network* in order to gain insights into the behaviors and relationships of vessels.

There is no single, absolute network of relationships within our AIS data. Rather, there are many different network representations that can be extracted and studied. In particular, the vertices of the network can denote an entity, a ping, or a POI. Similarly, the edges, or connections, can represent a drop, meet, or visit.

We are interested in finding connections between vessels and understanding how the network evolves over time. Specifically, we are interested in finding vessels that have visited a given POI and finding the *drops* and *meets* of that POI.

In the next section, we describe the data structures we use in order to organize, manipulate and explore the data as a network to address our questions of interest.

3.3 Data Structures

In order to explore the data, we first create several data structures to organize the data in an efficient and useful manner. We begin by creating three different class objects in the Python programming language (Python Software Foundation 2018). A *class* combines both data

and functionality together, and a class object is the means through which the functionality is obtained. Figure 3.2 provides an illustration of these data structures.

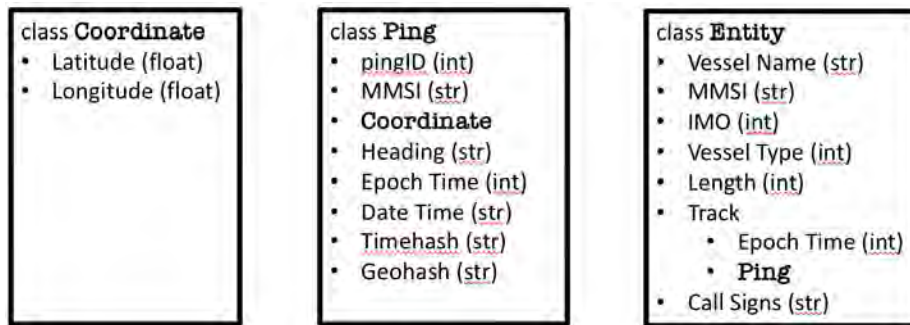
- The first class we define is the `Coordinate` class. This class has latitude and longitude as attributes. An *attribute* is a characteristic associated to all objects of the same class. The `Coordinate` class contains basic functionality for computing distances. We use the `Coordinate` class as our primary means of representing location information, particularly associated with a ping.
- The next class we create is the `Ping` class. The primary data associated with the `Ping` class is a location (in the form of a `Coordinate`) and a timestamp (typically in UTC), but in the context of AIS data it also contains heading and speed information, as well as information about the vessel (e.g., the MMSI) that generated the ping. Because each row in our AIS data corresponds to a single ping event, we customize the `Ping` class so that its attributes correspond directly to the fields in each row. As discussed below, for each row of AIS data we read, we create a corresponding `Ping` object.
- The last class we create is the `Entity` class. This class contains static information about the entity; in the context of AIS data this includes vessel name, MMSI, IMO, vessel type, and length. In addition, the `Entity` class contains a “track”, which is essentially a collection of `Ping` objects associated with that entity. More specifically, the track is a dictionary of key-value pairs, where each key is the timestamp of a ping, and the value is the corresponding `Ping` object.

These classes allow us to manipulate and store the data in a way that is both flexible and efficient for further analysis. Specifically, we create a dictionary of vessels (`Entity` objects) where the key is a MMSI and the value is the `Entity`. The algorithm for populating the dictionary is as follows.

```
create empty vessel dictionary
for each line of AIS data:
    parse the data and create a Ping object
    if the corresponding vessel is not in vessel dictionary:
        create a new Entity and add it to the dictionary
    add the Ping to the track of its corresponding Entity
```

With a single pass through our AIS data, this algorithm populates a dictionary that can be searched easily for a vessel by its MMSI, and for each vessel we can easily obtain information about its movement from its track. However, in order to answer our questions of interest in an efficient manner, we need to encode location and time information in a manner that makes it easy to search for them.

Figure 3.2. Class Attribute Associations



The attributes of the `Coordinate` class feed into the `Ping` class, and the attributes of the `Ping` class feed into the `Entity` class. In parenthesis are the data types: str = string, int = integer, and float = decimal number.

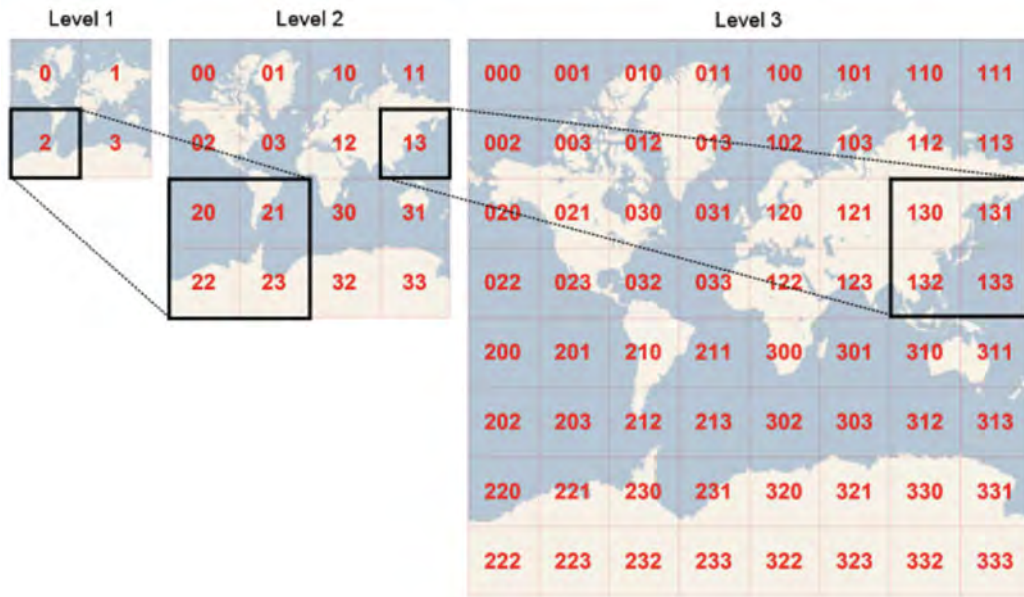
3.3.1 Hashing

In computer science, *hashing* is a fundamental concept in which one stores data in an associative array called a *hash table* that maps arbitrary keys to values. The keys are passed to a *hash function* that maps data of arbitrary size (the key) to a fixed size (the location in the array). A hash table is an abstract data type whose primary feature allows arbitrary additions and deletions that are efficient (i.e., in constant time on average, independent of the size of the array being managed).

We use two specific hash functions to support our analysis. The first of these is a *geohash*, which is a pre-defined function that encodes a latitude and longitude coordinate pair into a single, alphanumeric string. The underlying idea is to partition the surface of the globe into a grid of squares (or boxes) of fixed size. Each alphanumeric string represents a bounding box instead of a single point (Liu et al. 2014). The geohash function returns a string for the bounding box associated with the given coordinate.

A key feature of this encoding is that the number of characters in the geohash indicates how large this bounding is. For a visual representation, Figure 3.3 shows the recursive partitioning scheme used in geohash: at each level the existing boxes are subdivided and labelled. Thus, as the boxes get smaller in size, the corresponding geohash string gets longer, and vice versa. In general, we use term *precision* to indicate the number of characters included in the geohash.

Figure 3.3. Geohash Precision Example



Example of geohash precision (GitHub 2018).

We use the `pygeohash` Python module (McGinnis 2015), in which a geohash with five characters indicates a bounding box of length 3803 meters, or a box with an area of $3803 \text{ m} \times 3803 \text{ m}$. With additional characters in the geohash, the smaller the bounding box becomes, and the more precise the location. At a precision of 8 characters, the geohash represents a $19 \text{ m} \times 19 \text{ m}$ box.

A geohash is useful not only for encoding coordinates but also for quickly calculating distances. Two coordinates that are in the same geohash box, termed here a *geobox*, will have the same geohash. Moreover, because the encoding is “nested”, the closer two coordinates are to one another, the smaller the geobox they will occupy together, and the

more characters of their geohash they will share. Thus, the “nearness” of two coordinates can quickly be approximated based on the shared prefix of their geohash. For this reason, we often also refer to geohash precision as a *proximity*.

However, it is important to note the presence of boundary effects when employing geohash. Due to the discretization of geography, two entities that are close to each other may be placed into separate geoboxes if a geobox boundary falls in between them. For example, assume we want to find all entities within a $19\text{ m} \times 19\text{ m}$ box of an entity of interest. All entities in the same 8-character geobox as our entity of interest are within a $19\text{ m} \times 19\text{ m}$ box of our entity of interest, but not all entities within a $19\text{ m} \times 19\text{ m}$ box of our entity of interest are in the same 8-character geobox as our entity of interest.

In order to identify all entities that are within a $19\text{ m} \times 19\text{ m}$ box of our entity of interest, we need to implement a neighbor search (i.e., find all entities in the 8 surrounding geoboxes) and individually calculate their distances to our entity of interest. For the purposes of our analysis, we ignore the neighboring geoboxes and focus our efforts on entities that are within the same geobox as our entity of interest.

The second hash function that we implement is a *timehash*. We create a timehash by converting time into a standard twelve-digit representation of time: two digits each for year, month, day, hour, minute, and second (i.e., YYMMDDHHMMSS). Using the same concept of precision as indicated in the geohash function, the more precision we include in the timehash, the smaller the timebox is. For example, if we match on a timehash with a precision of 10 (i.e., YYMMDDHHMM), we include two digits for year, month, day, hour, and minute. Our timebox is now a 60-second window, where only the seconds will differ. If, however, we have a precision of 8 (i.e., YYMMDDHH), our timebox is an hour long, where minutes and seconds may differ, but the hour remains unchanged.

Given that we want to find entities that are “at the same position at the same time”, we can now say that “same location” means same geohash, and “same time” means same timehash for a specified level of precision. We embed both a geohash and timehash attribute in each Ping object we create. What remains is a need to quickly search for ping data with the same geohash and/or timehash values.

3.3.2 Searching

Answering our questions of interest at scale requires the ability to efficiently search for different data elements. To this end, we create four dictionaries.

- We create a vessel dictionary, as discussed above. The vessel dictionary maps a MMSI key to an `Entity` object.
- We create a geohash dictionary, which uses the geohash string as the key and a list of `Ping` objects (matching that complete geohash) as the value.
- We create a timehash dictionary, which uses a timehash as the key and a list of the `Ping` objects (matching that complete timehash) as the value.
- We create a ping dictionary, which uses a unique pingID number as the key and a `Ping` object as the value.

These dictionaries provide an efficient way to cross-check information. However, we also need the ability to find all geohashes that have the same prefix. To support this, we implement another data structure called a *trie*. A trie is a tree-like dictionary structure for storing information.

To understand how a trie works, consider a simple example originally provided by Flajolet (1983), suppose we have five items: 00110, 010, 101011, 1011, and 1100. A trie is organized such that elements sharing a common prefix are in the same part of the tree. For this example, we have 0 and 1 as common starting elements. At each level of the tree, the elements are broken down in an easily accessible manner. Figure 3.4 shows how this process works.

Figure 3.4. Example Trie Configuration

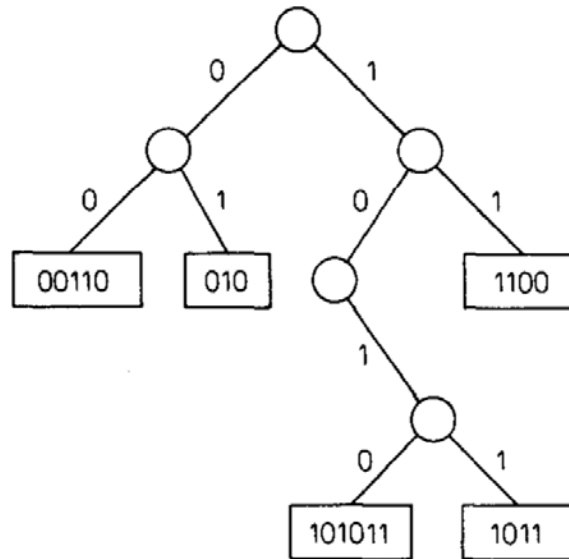


Figure 1 from On the Performance Evaluation of Extendible Hashing and Trie Searching (Flajolet 1983).

Each node on the tree is a key and the values are the branches below it. In this example, if you give a key of 101 (the node above the bottom two branches), you receive the values of 101011 and 1011. In the same way, if you give a key of 10, you receive the same values of 101011 and 1011 since these are all the branches of the given node (10). Since a trie is a dictionary, retrieving the values of a given key is a constant time operation, regardless of size.

We create two tries for our AIS data, one for geohashes and one for timehashes. Thus, for any given precision, we can immediately find all pings whose prefix matches either a given geohash (at the same location) or a given timehash (at the same time). To find ping data at the same location *and* the same time, we simply need to take the intersection of the Ping objects returned from both tries; for this we use the `set` data structure in Python, which does not allow duplicate elements in the identified intersection.

3.3.3 Building a Network

Once we have this basic foundation for our model, we can create networks at different granularity levels depending on the “nearness” of pings in space and time. Depending on our question of interest, two entities are connected if they occupy the same geobox and timebox (with specified precision for each). In the context of a network, we refer to two connected nodes as “neighbors”.

We construct the network by implementing what is known as the Breadth-First Search (BFS) algorithm. The BFS algorithm is a technique used to search a graph (Ahuja et al. 1993), but we use it here to explore all connections between entities and build the network as we do so. The basic concept of BFS is that it takes an initial node and searches all nodes directly connected to the initial node, putting them into a First In First Out (FIFO) queue to search later. Once a node has been searched, it is considered marked. The algorithm then reaches out from the next unmarked node in the queue, finding new connections and new nodes. This process repeats until there are no more unmarked nodes in the queue. At this point, all nodes and connections that are reachable from the initial node have been identified (see Ahuja et al. 1993, for additional details).

We use BFS as a means to illuminate the connections within our network, where connections are entities within the same geohash and timehash boxes. However, because BFS is only able to illuminate the parts of the network that are connected to the start node, we may have to execute this algorithm numerous times, once for each connected component of the network. In order to do this, we choose an entity from the set of entities within our data that are not part of a network. We use this entity as the starting entity and build out the network. This process continues until all entities are part of a network.

In the next section we walk through a few examples on a small subset of the AIS data to see how our model works and the output it produces.

3.4 Proof of Concept

In order to gain an understanding of how our model works, we first consider a small example. From the AIS data, we have a small subset of the data that consists of 22 observations from 6 entities. The data is seen in Table 3.1. For ease of viewing, we remove the vessel type,

length, IMO number, SOG, COG, and heading from this table as these attributes are not considered in our algorithms. The timehash and geohash fields are not part of the original raw AIS data (instead, we generate them as we read each line), but we include them here for clarity of exposition.

Table 3.1. Subset of AIS Data

Vessel Name	MMSI	CallSign	Lat	Long	Epoch Time	Time	timehash	geohash
667003226	667003226	NaN	10.15204	125.6705	1435984497	2015-07-04 04:34:00	150704043457	wcc3g8n7
667003226	667003226	NaN	10.13629	125.678	1436109620	2015-07-05 15:20:00	150705152020	wcc3ev7b
667003226	667003226	NaN	10.14316	125.6737	1436676539	2015-07-12 04:48:00	150712044859	wcc3ey8b
667003226	667003226	NaN	10.15254	125.6713	1437008133	2015-07-16 00:55:00	150716005533	wcc3g8pn
Golden Star5	600010885	JVKT5	10.18759	125.6693	1435970690	2015-07-04 00:44:00	150704004450	wcc3gwte
Golden Star5	600010885	JVKT5	10.14239	125.6751	1436973040	2015-07-15 15:10:00	150715151040	wcc3ey3g
Golden Star5	600010885	JVKT5	10.15204	125.6701	1437008075	2015-07-16 00:54:00	150716005435	wcc3g8n5
Golden Star5	600010885	JVKT5	10.15201	125.67	1437275380	2015-07-19 03:09:00	150719030940	wcc3g8n5
Golden Star5	600010885	JVKT5	10.15211	125.6702	1437653454	2015-07-23 12:10:00	150723121054	wcc3g8nh
Mei Ling5	667003373	DUB2032	10.18512	125.6668	1435970643	2015-07-04 00:44:00	150704004403	wcc3gw5u
Mei Ling5	667003373	DUB2032	10.15204	125.6707	1437499464	2015-07-21 17:24:00	150721172424	wcc3g8ne
Meiling 10	548015500	4DEW2	10.14415	125.6781	1436676589	2015-07-12 04:49:00	150712044949	wcc3eyev
Meiling 10	548015500	4DEW2	10.13789	125.6832	1438145849	2015-07-29 04:57:00	150729045729	wcc3evx9
Meiling 11	548016500	4DEW-3	10.15214	125.6709	1435984368	2015-07-04 04:32:00	150704043248	wcc3g8nu
Meiling 11	548016500	4DEW-3	10.13669	125.6776	1436330511	2015-07-08 04:41:00	150708044151	wcc3ev7d
Meiling 11	548016500	4DEW-3	10.14559	125.6806	1436972941	2015-07-15 15:09:00	150715150901	wcc3eyvy
Meiling 11	548016500	4DEW-3	10.15231	125.6711	1437277875	2015-07-19 03:51:00	150719035115	wcc3g8nv
Meiling 11	548016500	4DEW-3	10.15146	125.6725	1437499465	2015-07-21 17:24:00	150721172425	wcc3g8pb
Meiling 11	548016500	4DEW-3	10.15226	125.6711	1437653011	2015-07-23 12:03:00	150723120331	wcc3g8nv
Meiling 11	548016500	4DEW-3	10.13651	125.6757	1438145859	2015-07-29 04:57:00	150729045739	wcc3ev63
Yuedian 52	412607000	BYDA	10.137	125.6783	1436109452	2015-07-05 15:17:00	150705151732	wcc3evkh
Yuedian 52	412607000	BYDA	10.137	125.6783	1436330492	2015-07-08 04:41:00	150708044132	wcc3evkh

Since this is a small data set, we can verify the results of our algorithms by inspection. To begin, consider a POI with latitude and longitude coordinates of (10.13629, 125.6780), and let us examine which entities visit near that location. We use a precision of 7, which results in a geohash of wcc3ev7b and indicates that an entity must be within 118 meters of that point in order to be considered visiting. The results of this query return the data shown in Table 3.2.

Table 3.2. Results of Location Query

Vessel Name	MMSI	Lat	Long	Time	geohash
667003226	667003226	10.13629	125.678	2015-07-05 15:20:20	wcc3ev7b
Meiling 11	548016500	10.13669	125.6776	2015-07-08 04:41:51	wcc3ev7d

3.4.1 Searching an Entity of Interest

Next, suppose that we are interested in the entity with MMSI number 600010885 and we want to know with whom this entity is connected. As indicated, we need to specify the

precision in space (for the geobox) and time (for the timebox). If we choose a timehash with precision 8 (meaning that the year, month, day, and hour must match) and a geohash precision of 7 (indicating that connected entities must pass within 118 meters of each other), then we get the results in Table 3.3.

Table 3.3. Results of Connections Query with Geohash Precision = 7

Vesle IName	MMSI	Vessel Name	MMSI	timehash	geohash
Golden Star5	600010885	Meiling 11	548016500	15071903	wcc3g8n
Golden Star5	600010885	Meiling 11	548016500	15072312	wcc3g8n
Meiling 11	548016500	Mei Ling5	667003226	15070404	wcc3g8n

However, if we change the proximity parameters for our algorithm, we get a different network. In this example, if we change our geohash precision from 7 to 6 (meaning entities must pass within 610 meters of each other to be considered connected) and maintain our timehash precision at 8, then we get the results in Table 3.4.

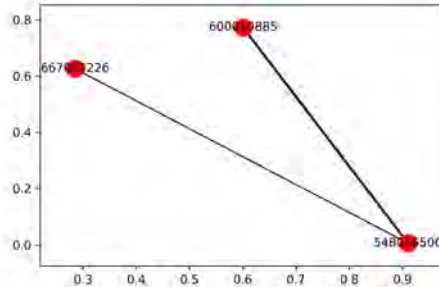
Table 3.4. Results of Connections Query with Geohash Precision = 6

Vessel Name	MMSI	Vessel Name	MMSI	timehash	geohash
Golden Star5	600010885	Mei Ling5	667003373	15070400	wcc3gw
Golden Star5	600010885	Meiling 11	548016500	15071515	wcc3ey
Golden Star5	600010885	Meiling 11	548016500	15071903	wcc3g8
Golden Star5	600010885	Meiling 11	548016500	15072312	wcc3g8
Golden Star5	600010885	667003226	667003226	15071600	wcc3g8
Mei Ling5	667003373	Meiling 11	548016500	15072117	wcc3g8
Meiling 11	548016500	667003226	667003226	15070404	wcc3g8
Meiling 11	548016500	Yuedian 52	412607000	15070804	wcc3ev
Meiling 11	548016500	Meiling 10	548015500	15072904	wcc3ev
667003226	667003226	Yuedian 52	412607000	15070515	wcc3ev
667003226	667003226	Meiling 10	548015500	15071204	wcc3ey

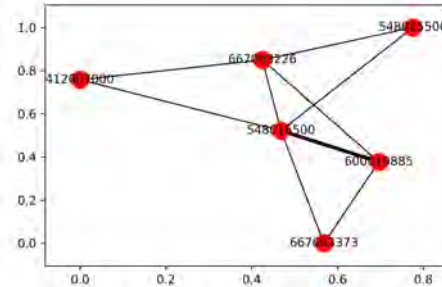
The corresponding networks are shown in Figure 3.5. When representing network connections, we use a weighted graph to indicate the number of times entities are “near” to each other. As seen in Figure 3.5a, the edge connecting entities 548016500 and 600010885 is bold. This indicates that the entities are nearby multiple times, which we see from the data in Table 3.3.

Figure 3.5. Example Connections Networks

(a) Geohash with Precision = 7



(b) Geohash with Precision = 6



These figures are rendered using the Matplotlib package in Python (Python Software Foundation 2018).

In Figure 3.5b, we see a different network representation when we decrease our level of precision in terms of location. As seen from Table 3.4, there are several connections between MMSI number 600010885 and MMSI number 548016500. This is also indicated by the bold line connecting these two nodes in Figure 3.5b.

3.4.2 Searching a Time of Interest

Now if we examine a specific time instead of a specific entity, we find out which locations have a higher concentration of activity. For example, suppose we wish to see the level of activity for July 4, 2015. Following along with our example, we see from inspection that the entities in Table 3.5 are active throughout the day.

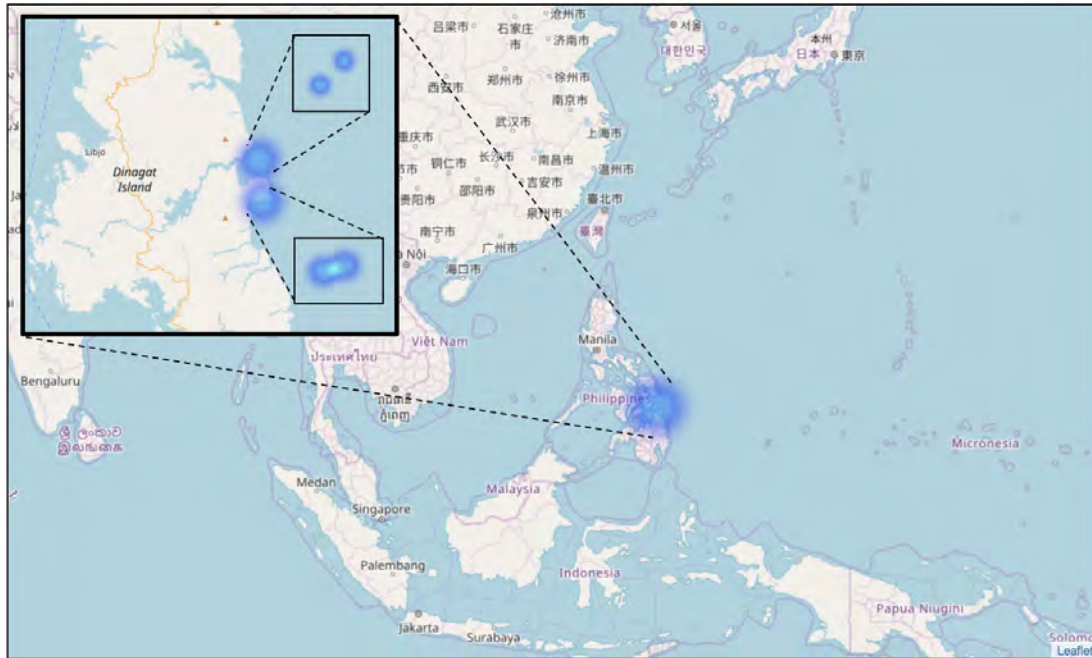
Table 3.5. Results of Time Query with Timehash Precision = 6

Vessel Name	MMSI	Lat	Long	Time	geohash
Golden Star5	600010885	10.18759	125.6693	2015-07-04 00:44:50	wcc3gwte
Meiling 11	548016500	10.15214	125.6709	2015-07-04 04:32:48	wcc3g8nu
667003226	667003226	10.15204	125.6705	2015-07-04 04:34:57	wcc3g8n7
Mei Ling5	667003373	10.18512	125.6668	2015-07-04 00:44:03	wcc3gw5u

We use a heatmap to represent the activity throughout the day. In Figure 3.6, the map highlights the region that has observations throughout the day. In the top left of the figure, we zoom in to get a better view of the activity. From this, we see there are two distinct areas

that have activity. Upon further inspection, we see that there are two observations in both the upper and lower regions throughout the day.

Figure 3.6. Results of Time Query



This figure is rendered using the Folium package in Python (Python Software Foundation 2018).

3.4.3 Tracking an Entity

In order to answer which locations an entity visits and at which times, we plot the entity's track that is created when we import the data. For this example, we use the entity with MMSI number 548016500. We see from Table 3.6 that this entity has 7 observations.

Table 3.6. Data Results of Entity Track Query

Vessel Name	MMSI	Lat	Long	Time	geohash
Meiling 11	548016500	10.15214	125.6709	2015-07-04 04:32:48	wcc3g8nu
Meiling 11	548016500	10.13669	125.6776	2015-07-08 04:41:51	wcc3ev7d
Meiling 11	548016500	10.14559	125.6806	2015-07-15 15:09:01	wcc3eyvy
Meiling 11	548016500	10.15231	125.6711	2015-07-19 03:51:15	wcc3g8nv
Meiling 11	548016500	10.15146	125.6725	2015-07-21 17:24:25	wcc3g8pb
Meiling 11	548016500	10.15226	125.6711	2015-07-23 12:03:31	wcc3g8nv
Meiling 11	548016500	10.13651	125.6757	2015-07-29 04:57:39	wcc3ev63

With this data, we create the track map in Figure 3.7. The red dots in the figure correspond to the entity's ping data. We can click on any dot to find the latitude, longitude, and time information associated with the entity's ping data at that location. In the top left of the figure we zoom in to get a better view of the overlapping observations. This figure shows 7 observations, which we expect to see from the data in Table 3.6.

Figure 3.7. Map Plot of Entity Track Query



This figure is rendered using the Folium package in Python (Python Software Foundation 2018).

In the next section, we implement our algorithms on the full data and investigate the networks and activities throughout the region. After we gather our preliminary results, we proceed to analyze the data and draw conclusions on the changes of the network over time.

CHAPTER 4:

Results and Analysis

4.1 Data Processing

The AIS data used in this thesis is a subset of a larger, global data base collected by receiving stations and satellites. The data is obtained from the Naval Postgraduate School (NPS) and contains about six million observations with over 37,000 unique vessels. We focus on the South China Sea because of the potential intelligence interest contained within the region.

4.1.1 Geographic focus: South China Sea

We filter the data to contain only those observations with positional coordinates within the South China Sea. For our purposes, we define the South China Sea to be the area between latitudes 4 and 40 degrees north and longitudes between 107 and 126 degrees east as shown in Figure 4.1.

Figure 4.1. South China Sea Region of Interest for AIS Data.



The blue box indicates the latitudes between 4 and 40 degrees north and the longitudes between 107 and 126 degrees east. This plot is rendered using the Folium package in Python (Python Software Foundation 2018).

4.1.2 Data Validation

After importing and examining the data, we notice some discrepancies within the data. In particular, when we examine the MMSIs, we notice that there are several occurrences with fewer than, or greater than, nine digits. In addition, there are numerous MMSIs that still contain the default MMSI. Default MMSIs are pre-loaded onto the transponder before installation and require the owner to update them with the current information. Once we remove all discrepancies within the MMSI column, we ensure that we have enough observations per vessel within our data, using a threshold of ten observations. If an MMSI does not have at least ten observations, we remove it from our data.

Next, we test each set of observations per MMSI to see if the observations are feasible. For example, we calculate an inferred SOG from raw position data as a means to ensure vessels are not teleporting, or moving faster than is mechanically possible, from one location to the next. Specifically, we calculate the inferred SOG based on the UTC and the positional coordinates. If the vessel exceeds 35 knots throughout the course of its route, we consider that segment to be infeasible and remove it from our data. We check the last known position the vessel travels below 35 knots and proceed to check the rest of the route. If more than ten percent of the route is erroneous, we completely remove that MMSI from our data.

Once the data is cleaned and processed, we read in the remaining observations into our data structures. The remaining data contains over 4.5 million geohashes in the geohash trie and nearly 3 million timehashes in the timehash trie.

4.2 Network Exploration

In this section we analyze and model different networks obtained from our methodology, in order to see how the geohash precision and timehash precision affect the network. In particular, we look at the topology of the obtained networks and their characteristics.

4.2.1 Networks with Varied Geohash Precision

We begin our analysis by varying the geohash precision while keeping the timehash precision constant to see how the network changes. In Figure 4.2, we see the networks created with a timehash precision of 10 and geohash precisions of 8, 7, and 6.

Figure 4.2. Networks with Timehash Precision = 10



These figures are rendered using Gephi (Bastian et al. 2009). The networks are created (from left to right) with geohash precision 8, 7, and 6.

The attributes of the networks in Figure 4.2 are displayed in Table 4.1. The columns indicate the geohash precision used to create the network, the average degree of the network, the average path length of the network, the K-Core of the network, the number of edges visible for the given K-Core, the number of nodes visible for the given K-Core, the number of communities within the network, and the node that is most central to the network.

Table 4.1. Network Characteristics for Timehash Precision = 10

Geohash	Nodes	Edges	Degree	Path Length	K-Core	Edges Visible	Nodes Visible	Communities	Central Node
8	4	3	1.5	1.5	1	4 (100%)	3 (100%)	1	600010885
7	1121	1936	3.5	7.9	9	18 (1.61%)	111 (5.73%)	23	457900024
6	4880	17272	7.1	5.9	23	53 (1.09%)	846 (4.9%)	40	412018240

The networks seen in Figure 4.2 are drastically different from the networks we create with a timehash precision of 9. Figure 4.3 shows the networks created with timehash precision of 9 and varied geohash precisions.

Figure 4.3. Networks with Timehash Precision = 9



These figures are rendered using Gephi (Bastian et al. 2009). The networks are created (from left to right) with geohash precision 8, 7, and 6.

While the network with geohash precision of 8 is the same for timehash precisions of 9 and 10, the networks created with geohash precision of 7 and 6 are significantly larger. The network characteristics change with this new level of granularity, as indicated by Table 4.2.

Table 4.2. Network Characteristics for Timehash Precision = 9

Geohash	Nodes	Edges	Degree	Path Length	K-Core	Edges Visible	Nodes Visible	Communities	Central Node
8	4	3	1.5	1.5	1	4 (100%)	3 (100%)	1	600010885
7	2011	4397	4.4	8	16	23 (1.14%)	219 (4.98%)	32	413018020
6	6481	28025	8.6	5.4	36	59 (0.91%)	1319 (4.71%)	36	413018020

With a timehash precision of 9, we start to see that the most central node changes from MMSI number 600010885 at geohash precision of 8 to MMSI number 413018020 with geohash precisions 7 and 6.

Finally, we look at the networks created with a timehash precision of 8 and geohash precisions of 8, 7, and 6. Figure 4.4 shows these networks and Table 4.3 displays their characteristics.

Figure 4.4. Networks with Timehash Precision = 8



These figures are rendered using Gephi (Bastian et al. 2009). The networks are created (from left to right) with geohash precision 8, 7, and 6.

Table 4.3. Network Characteristics for Timehash Precision = 8

Geohash	Nodes	Edges	Degree	Path Length	K-Core	Edges Visible	Nodes Visible	Communities	Central Node
8	4	3	1.5	1.5	1	4 (100%)	3 (100%)	1	600010885
7	2427	5833	4.8	8	18	25 (1.03%)	266 (4.56%)	40	413018020
6	7678	34454	9	5.3	40	79 (1.03%)	2194 (6.37%)	40	413018020

From Figures 4.2, 4.3, and 4.4, we see that there is a significant difference between the sizes and topology of the obtained networks based on the same timehash precision, yet different geohash precisions. From the network characteristic tables, we see that the central node within the network changes as the size of the network increases. In addition, we see that the size of the strongly connected components of the K-Core increase as the network increases, and we begin to see more communities within our networks as we grow the network. However, the average degree and average path length tend to remain unchanged across the different timehash precisions.

4.2.2 Networks with Varied Timehash Precision

After we examine the different networks through changing geohash precision, we now look at how the timehash precision affects the obtained network. In Figure 4.5, we see the networks created with a geohash precision of 8 and timehash precisions of 10, 9, 8, and 7. The timehash precision of 10, 9 and 8 produces the same network, and thus we present them only once on the left of the figure.

Figure 4.5. Networks with Geohash Precision = 8



These figures are rendered using Gephi (Bastian et al. 2009). The network on the left shows the same network obtained for a timehash precisions 10, 9, and 8. The network on the right shows timehash precision 7.

Figure 4.5 shows that as we change our level of connection, from entities that are connected through a meet to entities that are connected through a drop or a visit, the graph grows from 4 entities to 230 entities. Table 4.4 shows the change in network characteristics occurs from timehash 8 to timehash 7. The network increases in size and the new central node becomes MMSI number 667003215. We also see a significant jump in the number of communities in the network with timehash precision of 7, which provides more meaningful topology.

Table 4.4. Network Characteristics for Geohash Precision = 8

Timehash	Nodes	Edges	Degree	Path Length	K-Core	Edges Visible	Nodes Visible	Communities	Central Node
10	4	3	1.5	1.5	1	4 (100%)	3 (100%)	1	600010885
9	4	3	1.5	1.5	1	4 (100%)	3 (100%)	1	600010885
8	4	3	1.5	1.5	1	4 (100%)	3 (100%)	1	600010885
7	230	285	2.5	8	3	18 (7.83%)	35 (12.28%)	16	667003215

Next, we examine the networks with a geohash precision of 7. Figure 4.6 shows the different networks with timehash precision of 10, 9, and 8.

Figure 4.6. Networks with Geohash Precision = 7



These figures are rendered using Gephi (Bastian et al. 2009). The networks are created (from left to right) with timehash precision 10, 9, and 8.

At first glance, the network on the right may appear smaller than the center network, however, it has more communities, slightly more nodes, and more edges than the center network. In addition, the central node shifts from timehash precision 10 to timehash precision 9, but remains unchanged with timehash precision 8. Table 4.5 summarizes the sizes and shows the characteristics of these networks. Note that the number of communities increases as the number of entities within the network increases. The average path stays the same and there is only a slight change in the average degree between the networks.

Table 4.5. Network Characteristics for Geohash Precision = 7

Timehash	Nodes	Edges	Degree	Path Length	K-Core	Edges Visible	Nodes Visible	Communities	Central Node
10	1121	1936	3.5	7.9	9	18 (1.61%)	111 (5.73%)	23	457900024
9	2011	4397	4.4	8	16	23 (1.14%)	219 (4.98%)	32	413018020
8	2427	5833	4.8	8	18	25 (1.03%)	266 (4.56%)	40	413018020

Finally, we examine the networks with a geohash precision of 6. The networks range from 4880 nodes to 7678 nodes, and 17272 edges to 34454 edges. The number of entities and the connections between entities is significantly more for the networks with geohash precision of 6 compared to the networks with geohash precisions of 8 and 7. Figure 4.7 shows the graphs of these networks.

Figure 4.7. Networks with Geohash Precision = 6



These figures are rendered using Gephi (Bastian et al. 2009). The networks are created (from left to right) with timehash precision 10, 9, and 8.

The networks with geohash precision 6 have similar path lengths and communities, but vary greatly in the number of entities that are in the K-Core. As with the networks with geohash precision 7, the networks with geohash precision 6 change central nodes from timehash precision 10 (MMSI number 412018240) to timehash precision 9 (MMSI number 413018020). This is summarized in Table 4.6.

Table 4.6. Network Characteristics for Geohash Precision = 6

Timehash	Nodes	Edges	Degree	Path Length	K-Core	Edges Visible	Nodes Visible	Communities	Central Node
10	4880	17272	7.1	5.9	23	53 (1.09%)	846 (4.9%)	40	412018240
9	6481	28025	8.6	5.4	36	59 (0.91%)	1319 (4.71%)	36	413018020
8	7678	34454	9	5.3	40	79 (1.03%)	2194 (6.37%)	40	413018020

From this analysis, we note that there is a more significant change between the network structure when holding the timehash precision constant and varying the geohash precision than there is with the network structure when holding the geohash precision constant and varying the timehash precision. While there are topological differences in both cases, the differences are more apparent with changes in geohash precision than with timehash precision.

This result follows from the fact that we use AIS data. AIS is primarily used for collision avoidance, and as such, prevents vessels from being too close together. Therefore, we see that more observations occur within a larger region of an entity's position than within a larger time-window of the entity. In order for the time-window to affect the network, the time-window needs to be significantly larger. We specifically examine the cases where the

time-window is less than a day and would assume the same results of an increasing geo-box once the time-window becomes large enough.

4.3 Network Activity

In this section we examine ten different networks created through our BFS algorithm. We start each of our BFS algorithms with MMSI number 600010885 as our entity of interest. Table 4.7 shows a breakdown of the levels of precision used for the geohash and timehash in order to create these networks.

Table 4.7. BFS Networks Starting with MMSI Number 600010885

Geohash	Timehash	Nodes	Edges
8	10	4	3
8	9	4	3
8	8	4	3
8	7	230	285
7	10	1121	1936
7	9	2011	4397
7	8	2427	5833
6	10	4880	17272
6	9	6481	28025
6	8	7678	34454

Table 4.7 indicates that as the geohash precision and timehash precision decrease, the network's size increases. This is true for both the number of entities within the network as well as for the number of connections between entities. Due to the faulty use and capture of AIS data, we expect to see fewer connections between entities when the geo-box and time-box are small. With nearly 6 million observations and over 37,000 unique entities, we only see a network of 7678 entities when starting our BFS algorithm with MMSI number 600010885, geohash precision 6, and timehash precision 8. The geohash precision and timehash precision boxes are broken down in Table 4.8.

Table 4.8. Geohash and Timehash Precision Breakdown

Geohash Precision		Timehash Precision		
8	19 m $\times$ 19 m box	10	YYMMDDHHMM	60-second time window
7	118 m $\times$ 118 m box	9	YYMMDDHHM	10-minute time window
6	610 m $\times$ 610 m box	8	YYMMDDHH	1-hour time window
		7	YYMMDDH	10-hour time window

In order to see how the network changes over space and time, we examine the degree centrality of the network. The entity with the highest degree centrality is the entity that is most connected to the rest of the network. While we start with MMSI number 600010885, we see that this entity is not the most central, or most connected, to the network as the network grows. Table 4.9 indicates that while the network is very small, the starting entity (MMSI number 600010885) is the most central node in the network. However, as the network increases, due to decreasing geohash precision and timehash precision, the most central node changes.

Table 4.9. Degree Centrality for BFS Networks

Geohash	Timehash	Node	Value
8	10	600010885	1
8	9	600010885	1
8	8	600010885	1
8	7	667003215	0.069868996
7	10	457900024	0.036607143
7	9	413018020	0.023383085
7	8	413018020	0.021846661
6	10	412018240	0.031153925
6	9	413018020	0.031790123
6	8	413018020	0.031001693

To gain insight into why an entity is a central node within the network, we examine the locations the entity visits. We check to see how many observations each entity has within the AIS data to determine if the occurrences factor into determining the connections the entity makes within the network. Figure 4.8 shows the plots of the different entities and lists the number of observations each entity has within the data set. From this, we see that there are two main regions where the central node is located. The first region surrounds the Philippines, as seen from Figure 4.8 (a), (b), and (c). The other location is seen near the Port of Tianjin, shown in Figure 4.8 (d), and (e).

Figure 4.8. Visited Locations for Central Nodes

(a) Entity 600010885 with 12079 Observations



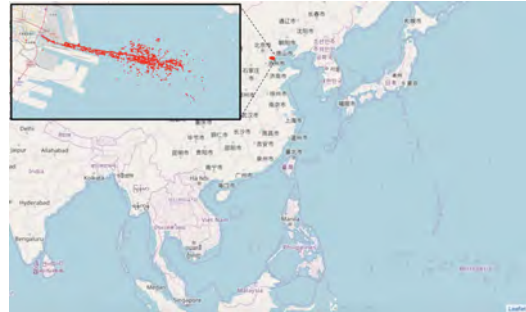
(b) Entity 667003215 with 8574 Observations



(c) Entity 457900024 with 13342 Observations



(d) Entity 413018020 with 2763 Observations



(e) Entity 412018240 with 2320 Observations



These figures are rendered using the Folium package in Python (Python Software Foundation 2018).

In addition to these two regions containing the central nodes, we note that there are significantly more observations for the entities surrounding the Philippines than there are for the entities with activity near the Port of Tianjin. However, since the use of AIS is not enforced, we assume that there are several vessels coming in and out of the Port of Tianjin area, but are not broadcasting their positional data after departing the congested sea-lanes.

From these observations, we expect that there is more vessel activity originating from China than from the Philippines. In order to check this, we make a heatmap of our networks to see which areas are traversed more frequently.

Figure 4.9. Network Heatmaps Showing Regional Activity

- (a) Geohash Precision = 8, Timehash Precision = 8 (b) Geohash Precision = 8, Timehash Precision = 7



- (c) Geohash Precision = 7, Timehash Precision = 8 (d) Geohash Precision = 5, Timehash Precision = 8



These figures are rendered using the Leaflet package in R (Cheng et al. 2018).

Figure 4.9 shows four different heatmaps selected from our set of networks. We plot two heatmaps with geohash precision 8, and then one each for geohash precisions 6 and 7. The heatmaps of our networks indicate that as the network grows, more activity is seen further north within the South China Sea. We query our data structures to see how many entities are within the Port of Tianjin, China, as compared to the Port of Davao, Philippines. From our query, we see that there are 364 unique entities that are observed near the Port of Tianjin, whereas there are only 181 unique entities observed near the Port of Davao. We plot the two ports in Figure 4.10

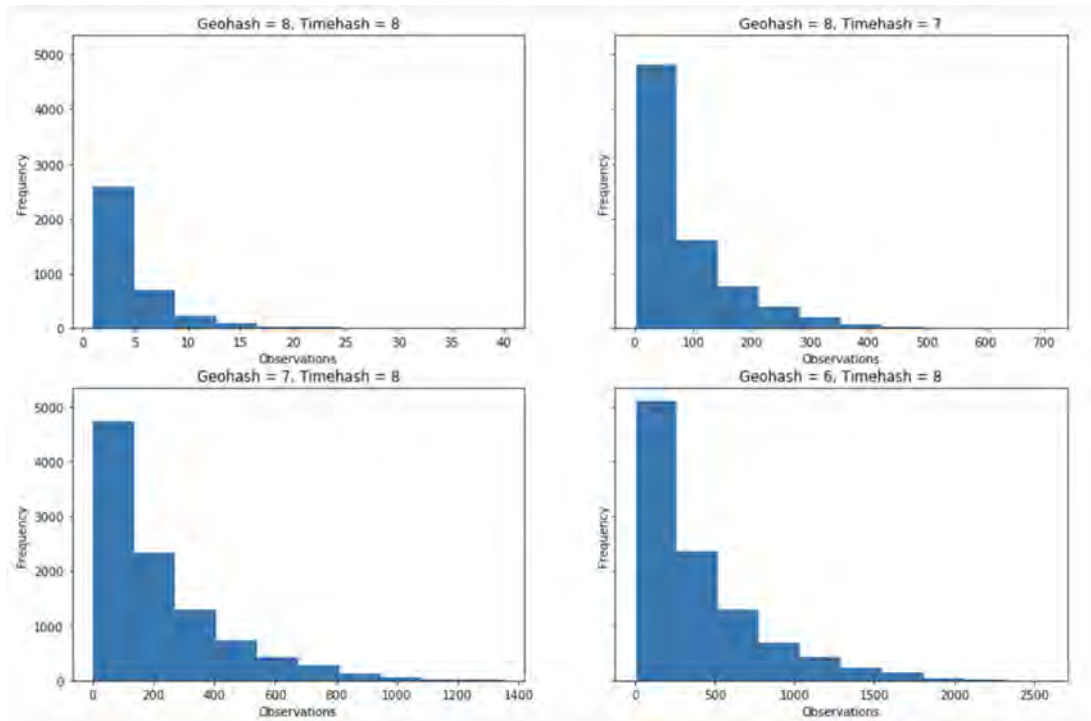
Figure 4.10. Port Locations.



This figure is rendered using the Folium package in Python (Python Software Foundation 2018).

In addition to the heatmaps shown in Figure 4.9, we create histograms to examine the network activity. As the number of entities contained within the network increases, so does the activity seen in a given day. Figure 4.11 shows the activity levels of four of our networks. The top-left histogram in Figure 4.11 shows the activity level of a network with four entities, whereas the bottom-right histogram shows the activity level for a network with 7678 entities.

Figure 4.11. Histogram of Network Activity.



This figure is rendered using the Matplotlib package in Python (Python Software Foundation 2018).

Next, we look at which locations are visited during the most active hour for four of our networks. The breakdown for the most active hour and the number of observations during that time period are shown in Table 4.10

Table 4.10. Time Query for Most Active Hour

Geohash Precision	Observations	Timehash Precision	YYMMDDHH
8	40	8	15080412
8	705	7	15101012
7	1352	8	15101012
6	2581	8	16022317

We plot the heatmaps for these four networks to see which areas are visited during this time period. Figure 4.12 shows the results of this query.

Figure 4.12. Network Heatmaps Showing Active Locations for a Given Time

(a) Geohash Precision = 8, Timehash Precision = 8 (b) Geohash Precision = 8, Timehash Precision = 7



(c) Geohash Precision = 7, Timehash Precision = 8 (d) Geohash Precision = 6, Timehash Precision = 8



These figures are rendered using the Leaflet package in R (Cheng et al. 2018).

From this small sample, it appears that noon local time is the most active hour throughout the day. We further explored this by looking at the entire AIS data set, expanding on the data from our networks. The results of the hourly activity throughout our data is shown in Table 4.11. We see that, on average, noon is indeed the most active hour throughout the day, followed closely by midnight, 4 AM, and 4 PM.

Table 4.11. Average Hourly Activity throughout the Day

Hour	Observations	Percent
00	502740	8.62%
01	304914	5.23%
02	205890	3.53%
03	358166	6.14%
04	500027	8.58%
05	350154	6.01%
06	145554	2.50%
07	77021	1.32%
08	39534	0.68%
09	31648	0.54%
10	34063	0.58%
11	88251	1.51%
12	592637	10.17%
13	463288	7.95%
14	276796	4.75%
15	297020	5.10%
16	500207	8.58%
17	404060	6.93%
18	181230	3.11%
19	135220	2.32%
20	152705	2.62%
21	79605	1.37%
22	38969	0.67%
23	69801	1.20%

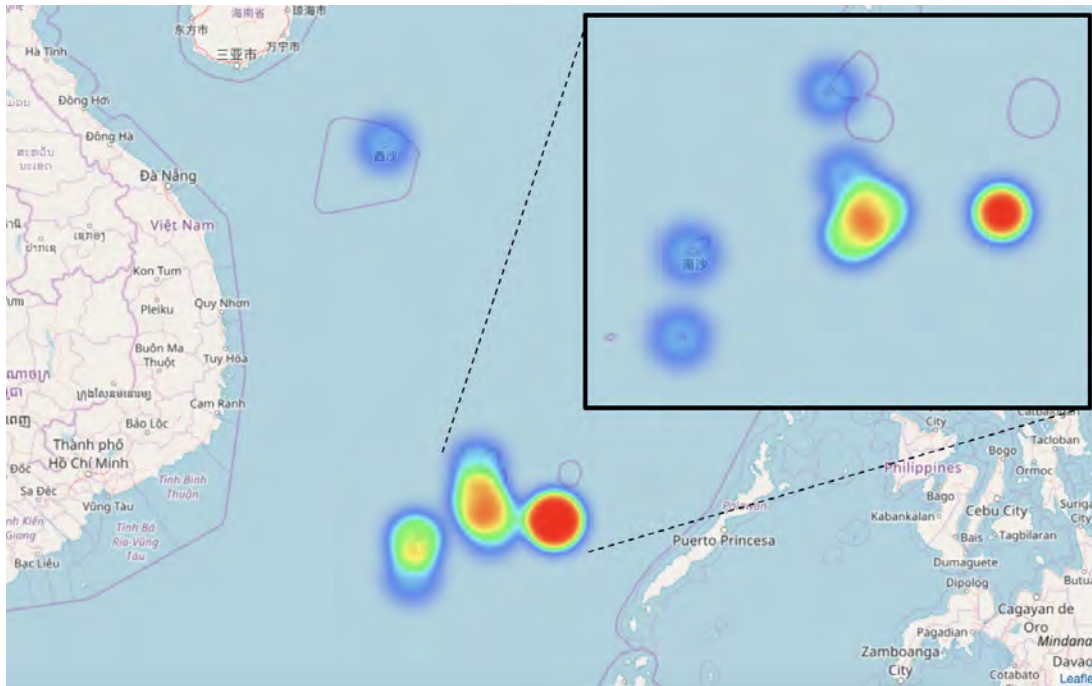
4.4 Application: The Spratly Islands

In this section we examine the activity in and around the Spratly Islands. We specifically look at which entities visit the following POI: Woody Island, Gaven Reef, Fiery Cross Reef, Hughes Reef, Scarborough Reef, Mischief Reef, Johnson South Reef, Cuarteron Reef, and Subi Reef. We are interested in finding out which reef is most active, which entities visit which reefs, and which entities are connected within the reef network.

We start by examining the activity level around the reefs. Figure 4.13 shows the heatmap of activity in our data. We see from the zoomed in figure in the top right that Mischief Reef is the most active. To confirm this, we conduct two queries in order to gain an understanding

of how much activity is seen at Mischief Reef compared to the other reefs.

Figure 4.13. Heatmap of Reef Activity.



This figure is rendered using the Folium package in Python (Python Software Foundation 2018).

The first query we conduct examines how many unique entities visit each reef. Table 4.12 displays the reefs and the number of entities that have visited the reef.

Table 4.12. Unique Entities Visited at each Reef

Reef	Entities Visited
Cuarteron	19
Fiery Cross	35
Gaven	23
Hughes	30
Johnson South	28
Mischief	123
Scarborough	1
Subi	58
Woody Island	8

Next, we examine how many times a reef is visited, including multiple times from a single entity. We see that regardless of how we view the activity, Mischief Reef is by far the most active reef among those that we are examining. Table 4.13 shows the results of the second query we conduct.

Table 4.13. Total Times each Reef Visited

Reef	Times Visited
Cuarteron	173
Fiery Cross	413
Gaven	190
Hughes	342
Johnson South	309
Mischief	4007
Scarborough	2
Subi	994
Woody Island	37

Since Mischief Reef is the most active reef from our list, we look to see which entities tend to visit in and around Mischief Reef. Table 4.14 shows the top ten most active entities around Mischief Reef and the number of times they are observed in the area.

Table 4.14. The Most Active Entities at Mischief Reef

MMSI	Times Visited
413473360	1109
413040040	231
413300300	182
600013090	180
412686000	134
412472710	120
600013058	119
413070000	111
412222222	104
413700860	86

Expanding on our Mischief Reef query, we examine which entities have visited the most reefs in order to see which entities are the most active throughout the region. From this query, we find that there are three entities that visit 6 of the 9 reefs in our list. Table 4.15

lists these entities. We notice that the first entity in this table is also listed as the most active entity at Mischief Reef.

Table 4.15. Entities that have Visited 6 Reefs

MMSI	Reefs Visited					
413473360	Cuarteron Reef	Johnson South Reef	Mischief Reef	Hughes Reef	Subi Reef	Gaven Reef
413374950	Cuarteron Reef	Johnson South Reef	Mischief Reef	Hughes Reef	Subi Reef	Gaven Reef
413304980	Cuarteron Reef	Johnson South Reef	Fiery Cross Reef	Hughes Reef	Subi Reef	Gaven Reef

Our query shows that there are also 4 entities that visit 5 of the 9 reefs. Table 4.16 shows these entities, and we note that the second most active entity from our Mischief Reef search appears in this list.

Table 4.16. Entities that have Visited 5 Reefs

MMSI	Reefs Visited				
413054570	Cuarteron Reef	Johnson South Reef	Mischief Reef	Hughes Reef	Gaven Reef
413448260	Cuarteron Reef	Johnson South Reef	Fiery Cross Reef	Hughes Reef	Gaven Reef
412302150	Cuarteron Reef	Johnson South Reef	Fiery Cross Reef	Hughes Reef	Gaven Reef
413040040	Cuarteron Reef	Mischief Reef	Hughes Reef	Subi Reef	Gaven Reef

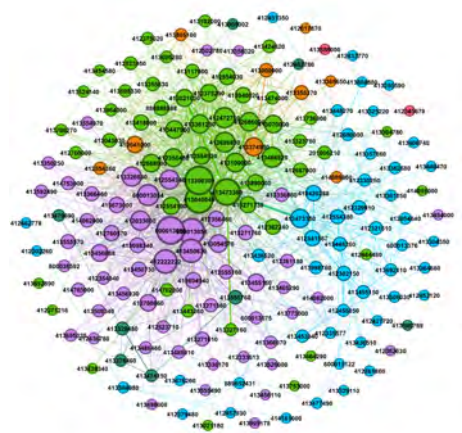
After we examine Mischief Reef and the activity throughout the Spratly Islands, we look to see how these entities are connected. From our previous queries, we see that entities 413473360 and 413040040 are fairly active and expect them to be central to the network. Table 4.17 shows the network characteristics for the reef visitors. We see from this table that our two nodes of interest are within the top three most central nodes within the network.

Table 4.17. Network Characteristics of Reef Visitors

Nodes	Edges	Degree	Path Length	K-Core	Edges Visible	Nodes Visible	Communities	Central Node	Rank
179	1429	16	2.4	18	368 (25.75%)	32 (17.88%)	5	413040040	1
								600013090	2
								413473360	3

Figure 4.14 shows the entire network of reef visitors and their interactions between each other. We see several entities within the center of the network that are connected to several other members of the network.

Figure 4.14. Network of Entities Visiting POI in the South China Sea.



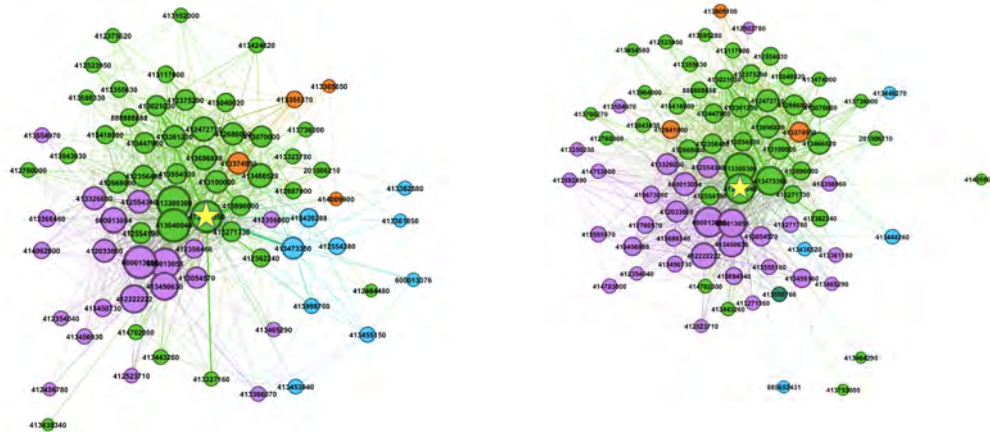
This figure rendered using Gephi (Bastian et al. 2009).

We look for entities 413473360 and 413040040 to see where they fit within the network. Figure 4.15 shows these entities and their connections. The entity of interest is highlighted with a yellow star. From this, we see that our two entities of interest within the region are part of the larger reef visitors network.

Figure 4.15. Network of 2 Entities of Interest

(a) Entity 413473360

(b) Entity 413040040



These figures rendered using Gephi (Bastian et al. 2009).

4.5 Computational Efficiency

Finally, we examine the efficiency of our technique for rapidly constructing social networks from AIS data. We compare the performance of our technique to the performance of a “naive” technique by filtering ping data. This filtering is based on “nearness” in both space and time as indicated by matching the prefix of the geohash and the timehash. The ping data is stored in a dataframe (i.e., a table) with the associated geohash and timehash already computed. Given an entity as a starting point, the pseudo code for this process is as follows.

- create empty temp dataframe
- find all ping observations for this entity; save as a subset
- for each ping observation in subset as ping1:
 - filter all rows in original dataframe that match prefix of
geohash and timehash of ping1
 - add these filtered rows to temp dataframe
- return the temp dataframe

This use of filtering dataframes is significantly slower than our use of dictionary structures and tries for both geohash and timehash. Our technique achieves decreased computational time by implementing a faster look-up and return of matching data.

Table 4.18 summarizes two cases that illustrate the computational efficiency of our technique. In all cases, we perform our computation on a personal MacBook Pro with a 3.1 GHz Intel Core i5 processor and 8 GB of memory. In our first example, we use a month’s subset of the data with 343,290 observations. To build a network with a geohash precision of 6, a timehash precision of 10, and a starting entity with MMSI number 413130000, a sub-setting search takes 423.05 seconds to complete. This is just over 7 minutes. Using the same parameters with our dictionary structures, the algorithm completes the network in 0.746 seconds. The dictionary structures we implement decrease the computational time to search and build the network by a factor of 566. When we increase the amount of data, we see this time-saving effect even more. With a data set of 1,282,953 observations spanning four months, a geohash precision of 7, a timehash precision of 10, and a starting entity with MMSI number 413130000, our sub-setting algorithm takes 71,858.47 seconds to complete. This equates to 19.96 hours. Using the same parameters but implementing the algorithm

with our dictionary structures, we create the same network in 50.13 seconds. This reduces the time by a factor of 1,433. Table 4.18 shows the computational savings for our new algorithm.

Table 4.18. Computational Efficiency of Two Different Techniques

Raw Data Observations	Network Construction Task	Naïve Technique	Our Technique	Savings
343,290	geohash = 6 timehash = 10 start node = 413130000 nodes = 22 edges = 31	423.05 sec	0.746 sec	566x
1,282,953	geohash = 7 timehash = 10 start node = 413130000 nodes = 129 edges = 167	71,858.47 sec	50.13 sec	1433x

In the next section we summarize our results and major findings in this thesis. We introduce areas for future research and further analysis.

CHAPTER 5:

Conclusion

This chapter consists of two sub-sections. In the first, we summarize our findings and our analysis, and in the second, we introduce areas for future work.

5.1 Summary of Analysis

In this thesis we look at an AIS data set consisting of nearly 6 million observations within the South China Sea. We create several data structures to hold and manipulate the data in order to efficiently query the data and return the results. We compare our data structures to a standard sub-setting query and find that our data structures decrease the computational time, with an efficiency of hundreds to thousands of times faster as the data set becomes larger. This efficiency provides us the flexibility to create and examine several networks with varying levels of granularity.

We specifically examine how changing geohash and timehash precisions affect the topological outcome of our network: number of entities (nodes), number of connections (edges), average path length, number of communities, and central node. We see within the AIS data that vessels typically do not come close together except in ports. This prevents us from using a precise geohash precision and we note that a geohash precision of 7 ($118 \text{ m} \times 118 \text{ m}$ box) or a geohash precision of 6 ($610 \text{ m} \times 610 \text{ m}$ box) allow us to quickly create a social network implied by vessel traffic. Additionally, we see that the timehash precision also impacts the network. We vary the timehash precision based on the analysis we wish to conduct.

Due to the ability to easily create networks, vary their geohash and timehash precisions, and efficiently query the data, we are able to gain insights into the data and the network activities within the South China Sea. We examine points-of-interest throughout the Spratly Islands and note that the majority of visitors originate from China. In addition, we see that out of the nine points-of-interest we examine, Mischief Reef receives nearly 40% of all visits. From these queries, we gain insights into the activity levels of all entities within our data

set and find that the most active time for vessel activity, on average, occurs during the noon hour.

Understanding activity levels, both time and location, allows us to focus our analysis on the highlighted entities and POIs. This allows us to focus on creating a network of interest, preventing unnecessary computation, saving time both computationally and analytically.

5.2 Future Work

Our methodology allows us to gain insight into network connections and activities with the use of our data structures, but we are unable to verify the results. We do not have ground truth for these networks, and while all networks vary based on the parameters used to create them, it would be beneficial to examine the results from our algorithm as compared to ground truth. Future work should implement this on known networks and see how accurate it is compared to ground truth.

In addition, our algorithms create fixed searches for location and time based on the length of characters in the geohash or timehash. A flexible search, dictated by user input, should be considered for creating networks based on both space and time proximity. Furthermore, our analysis never examines how the network changes as new information is received. Intelligence communities receive massive amounts of data every day. It is imperative to create and update networks in real-time, but our current algorithms do not account for new data.

Another area to consider in future work is pattern prediction. Several of the papers we examine in this thesis discuss this problem, but we do not account for these patterns or examine them in our research. Particular questions of interest are as follows.

- Given an entity left a POI, where is it most likely to go next?
- Given two entities that interact with each other, who are they most likely to interact with next?
- Are there patterns for an entity of interest over specific time frames (hourly, daily, monthly, yearly)?

These future research directions, along with our current efficiency, will allow for timely analysis and accurate depictions of activities and connections within a network.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

- Aarsæther KG, Moan T (2009) Estimating navigation patterns from ais. *The Journal of Navigation* 62(4):587–607.
- Ahuja RK, Magnanti TL, Orlin JB (1993) *Network Flows: Theory, Algorithms, and Applications* (Prentice Hall).
- Al-Molegi A, Jabreel M, Ghaleb B (2016) Stf-rnn: Space time features-based recurrent neural network for predicting people next location. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–7.
- Bastian M, Heymann S, Jacomy M, et al. (2009) Gephi: An open source software for exploring and manipulating networks. *ICWSM* 8:361–362.
- Bollobás B (1998) *Modern graph theory*, volume 184 (New York, New York: Springer Science & Business Media).
- Brightkite (2010) Brightkite Network Data. <https://snap.stanford.edu/data/loc-brightkite.html>.
- Cheng J, Karambelkar B, Xie Y (2018) *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. R package version 2.0.0.
- Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1082–1090.
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische mathematik* 1(1):269–271.
- Flajolet P (1983) On the performance evaluation of extendible hashing and trie searching. *Acta Informatica* 20(4):345–369.
- Gambs S, Killijian MO, del Prado Cortez MN (2012) Next place prediction using mobility markov chains. *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, 3.
- GitHub (2018) Geohashing Chat by Proximity. <https://github.com/stephenlb/geohash-chat-by-proximity>.
- Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning* (MIT Press), <http://www.deeplearningbook.org>.

- Gowalla (2010) Gowalla Network Data. <https://snap.stanford.edu/data/loc-gowalla.html>.
- Hagberg AA, Schult DA, Swart PJ (2008) Exploring network structure, dynamics, and function using networkx. Varoquaux G, Vaught T, Millman J, eds., *Proceedings of the 7th Python in Science Conference*, 11 – 15 (Pasadena, CA USA).
- Hern A (2018) Fitness tracking app strava gives away location of secret us army bases. <https://www.theguardian.com/world/2017/mar/12/netherlands-will-pay-the-price-for-blocking-turkish-visit-erdogan>, urldate = 2017-03-12.
- IMO (2018a) Automatic identification system regulations. <http://www.imo.org/en/OurWork/Safety/Navigation/Pages/AIS.aspx>.
- IMO (2018b) Automatic identification system regulations. <http://www.imo.org/en/OurWork/MSAS/Pages/IMO-identification-number-scheme.aspx>.
- Jiang X, de Souza EN, Soleimani BH, Wang X, Silver DL, Matwin S (2017) Partition-wise recurrent neural networks for point-based ais trajectory classification .
- Liu J, Li H, Gao Y, Yu H, Jiang D (2014) A geohash-based index for spatial data management in distributed memory. *22nd International Conference on Geoinformatics*, 1–4.
- Marine Traffic (2018) AIS Transmitted Information. <https://help.marinetraffic.com/hc/en-us/articles/205426887-What-kind-of-information-is-AIS-transmitted->.
- Marine Traffic Blog (2018) AIS Transmitted Information. <https://www.marinetraffic.com/blog/a-day-in-numbers/>.
- Mazzarella F, Vespe M, Damalas D, Osio G (2014) Discovering vessel activities at sea using ais data: Mapping of fishing footprints. *2014 17th International Conference on Information Fusion (FUSION)*, 1–7.
- McGinnis W (2015) pygeohash. <https://pypi.org/project/pygeohash/>.
- Newman M (2010) *Networks: An introduction* (Oxford, England: Oxford University Press).
- Newman ME (2003) The structure and function of complex networks. *SIAM review* 45(2):167–256.
- Pallotta G, Vespe M, Bryan K (2013) Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction. *Entropy* 15(6):2218–2245.
- Pedersen PT, et al. (1995) Collision and grounding mechanics. *Proceedings of WEMT* 95(1995):125–157.

- Python Software Foundation (2018) Python Programming Language. <https://www.python.org>.
- Ristic B, La Scala B, Morelande M, Gordon N (2008) Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction. *2008 11th International Conference on Information Fusion (FUSION)*, 1–7.
- Shao H, Japkowicz N, Abielmona R, Falcon R (2014) Vessel track correlation and association using fuzzy logic and echo state networks. *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2322–2329.
- Silveira P, Teixeira A, Soares CG (2013) Use of ais data to characterise marine traffic patterns and ship collision risk off the coast of portugal. *The Journal of Navigation* 66(6):879–898.
- SOLAS (2018) Regulations for Carriage of AIS. <http://www.imo.org/en/OurWork/Safety/Navigation/Pages/AIS.aspx>.
- Xu J, Wickramaratne TL, Chawla NV (2016) Representing higher-order dependencies in networks. *Science Advances* 2(5):e1600028.
- Xu JJ, Chen H (2004) Fighting organized crimes: using shortest-path algorithms to identify associations in criminal networks. *Decision Support Systems* 38(3):473–487.
- Ye J, Zhu Z, Cheng H (2013) What’s your next move: User activity prediction in location-based social networks. *Proceedings of the 2013 SIAM International Conference on Data Mining*, 171–179.
- Young BL (2017) *Predicting vessel trajectories from AIS data using R*. Ph.D. thesis, Monterey, California: Naval Postgraduate School.
- Zhang W, Goerlandt F, Montewka J, Kujala P (2015) A method for detecting possible near miss ship collisions from ais data. *Ocean Engineering* 107:60–69.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California