Automated Risk Management Framework with Application to Big Maritime Data

by

Alexander Teske

A thesis submitted in partial fulfillment of the requirements for the

Master's Degree

in

Computer Science

Ottawa-Carleton Institute of Computer Science
School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Risk management is an essential tool for ensuring the safety and timeliness of maritime operations and transportation. Some of the many risk factors that can compromise the smooth operation of maritime activities include harsh weather and pirate activity. However, identifying and quantifying the extent of these risk factors for a particular vessel is not a trivial process. One challenge is that processing the vast amounts of automatic identification system (AIS) messages generated by the ships requires significant computational resources. Another is that the risk management process partially relies on human expertise, which can be time-consuming and error-prone.

In this thesis, an existing Risk Management Framework (RMF) is augmented to address these issues. A parallel/distributed version of the RMF is developed to efficiently process large volumes of AIS data and assess the risk levels of the corresponding vessels in near-real-time. A genetic fuzzy system is added to the RMF's Risk Assessment module in order to automatically learn the fuzzy rule base governing the risk assessment process, thereby reducing the reliance on human domain experts. A new *weather risk* feature is proposed, and an existing *regional hostility* feature is extended to automatically learn about pirate activity by ingesting unstructured news articles and incident reports. Finally, a geovisualization tool is developed to display the position and risk levels of ships at sea. Together, these contributions pave the way towards truly automatic risk management, a crucial component of modern maritime solutions. The outcomes of this thesis will contribute to enhance Larus Technologies' Total::Insight, a risk-aware decision support system successfully deployed in maritime scenarios.

Acknowledgements

My deepest gratitude goes to my thesis supervisor, Dr. Emil Petriu and my thesis co-supervisor, Dr. Rafael Falcon. Dr. Petriu's support has made this research possible and has enabled me to come this far. Dr. Falcon introduced me to my current research interests as well as the academic realm in general. He is a brilliant scientist and has been an incredible mentor to me over the past 6+ years.

Special thanks go to Dr. Rami Abielmona, whose invaluable guidance and advice have made me a better researcher and professional.

I also thank Ashwin Panchapakesan. His advice, humor, and conversation kept me sane during the last two years.

I would like to thank my parents and family for their unconditional support and encouragement.

Finally, I would like to acknowledge the financial support of the Ontario Centres of Excellence (OCE) and the National Sciences and Engineering Research Council of Canada (NSERC) for the project entitled "Big Data Analytics for the Maritime Internet of Things".

Preface

Two of this thesis's contributions appeared in other publications over the last two years. The contributions of Chapter 3 will be published in the upcoming book chapter "*Genetic Fuzzy Systems for Automating Maritime Risk Assessment*" [1]. The contributions of Chapter 4 were published in "*Automatic Identification of Maritime Incidents from Unstructured Articles*" [2] and have been expanded for this work. The author of this thesis was the lead researcher and lead writer for both of these publications. Content from these publications are reused in this thesis with permission.

Contents

Abstract ii

Acknowledgements iii

Preface iv

List of Figures viii

List of Tables ix

Abbreviations x

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Contributions . . . 3
- 1.3 Outline . . . 4

2 Technical Background and Related Works 5
- 2.1 Maritime Domain Awareness . . . 5
- 2.2 Risk Management Framework . . . 8
  - 2.2.1 Risk Management Framework for Maritime Domain Awareness 10
- 2.3 Genetic Fuzzy Systems . . . 12
- 2.4 Natural Language Processing . . . 15
  - 2.4.1 Document Classification . . . 15
  - 2.4.2 Information Extraction . . . 16
- 2.5 Parallel and Distributed Computing . . . 17
- 2.6 Maritime Visualization . . . 18
- 2.7 Chapter Summary . . . 19

3 Automatic Maritime Risk Assessment 20
- 3.1 Introduction . . . 20

3.2 Proposed Approach 21
3.2.1 Data Sources 21
3.2.2 Risk Features 23
3.2.3 Ground Truth 23
3.2.4 Genetic Fuzzy Systems 24
3.2.5 Performance Metrics 24
3.2.5.1 Statistical Analysis 25
3.3 Experiments 26
3.3.1 Case Studies 26
3.3.2 Experimental Setup 29
3.3.3 Experimental Results 30
3.3.4 Characterization of Fuzzy Rule Base Per AOI 33
3.3.5 Accuracy vs. Interpretability 34
3.4 Chapter Summary 36

4 Automatic Identification of Maritime Incidents from Unstructured Articles 37
4.1 Introduction 37
4.2 Proposed Approach 40
4.2.1 Data Sources 40
4.2.2 Article Classification 42
4.2.3 Maritime Incident Information Extraction 44
4.2.3.1 Extracting the Incident Location 44
4.2.3.2 Extracting the Incident Type 47
4.2.3.3 Extracting the Victim's Vessel Type 49
4.3 Experimental Results 52
4.3.1 Document Classification Results 52
4.3.2 Information Extraction Results 53
4.4 Chapter Summary 53

5 Parallel Maritime Risk Assessment 55
5.1 Parallel Risk Management Framework 55
5.1.1 Example: Parallelizing Collision Factor Calculations 56
5.1.2 Data Sources 57
5.1.3 Benchmark 59
5.1.4 Spark vs. MPI 61
5.2 Weather Risk Factor 62
5.3 Chapter Summary 63

6 Maritime Risk Visualization 65
6.1 Proposed Visualization System 65
6.2 Chapter Summary 67

7 Conclusion **70**
7.1 Summary of Contributions 70
7.2 Future Research Directions 71

A Regular Expressions for Detecting Locations **73**

B Regular Expressions for Detecting Incident Types **75**

C Regular Expressions for Detecting Vessel Types **77**

Bibliography **79**

List of Figures

2.1 Risk Management Framework architecture 9
2.2 Typical genetic fuzzy system 13

3.1 Proposed architecture to automate MRA with GFSs 22
3.2 Maritime incidents, Gulf of Guinea, 2017 27
3.3 Maritime incidents, Strait of Malacca, 2017 28
3.4 Wave height, North Atlantic Ocean, January 1 2018 0:00:00GMT . 28
3.5 Wave height, North Atlantic Ocean, January 13 2018 0:00:00GMT . 29

4.1 Number of articles from each data source 41
4.2 Natural language processing architecture 43

5.1 Example of a KD-Tree 57
5.2 Proposed design of the Parallel Risk Management Framework for Maritime Risk Assessment 58
5.3 Sample incident from IMB's Piracy and Armed Robbery Against Ships report 59
5.4 MPI-PRMF benchmark 60
5.5 Spark-PRFM benchmark 60
5.6 Spark-PRFM vs. MPI-PRMF benchmark 61
5.7 Trapezoidal membership function to map wave height to risk 63

6.1 Visualization layers panel 67
6.2 Vessel information panel 67
6.3 Fleet monitoring panel 68
6.4 Visualization settings panel 68
6.5 Response selection panel 69

List of Tables

3.1 GFS algorithms compared 25
3.2 Accuracy results for the Guinea scenario 30
3.3 Accuracy results for the Malacca scenario 30
3.4 Accuracy results for the Atlantic Storm scenario 31
3.5 Accuracy results for the Atlantic No Storm scenario 31
3.6 Interpretability results for the Guinea scenario 31
3.7 Interpretability results for the Malacca scenario 32
3.8 Interpretability results for the Atlantic Storm scenario 32
3.9 Interpretability results for the Atlantic No Storm scenario 32
3.10 Average distribution of risk features in rule conditions 34

4.1 Weka 3.8 Classifiers 44
4.2 Classification accuracy by feature selection method, bag-of-words type, and article content 51
4.3 Performance of maritime article classifiers 51
4.4 Performance metrics for Logistic Regression classifier 51
4.5 Confusion matrix of the Logistic Regression classifier 51

Abbreviations

AIS	Automatic Identification System
CCP	Cooperation vs. Competition Problem
FIS	Fuzzy Inference System
FRB	Fuzzy Rule Base
GA	Genetic Algorithm
GFS	Genetic Fuzzy System
GRIB	Gridded Binary
IMB	International Maritime Bureau
IMO	International Maritime Organization
IRL	Iterative Rule Learning
KEEL	Knowledge Extraction based on Evolutionary Learning
MCDA	Multi-Criteria Decision Analysis
MDA	Maritime Domain Awareness
MPI	Message Passing Interface
MRA	Maritime Risk Assessment
NER	Named Entity Recognition
NGA	National Geospatial-Intelligence Agency
NLP	Natural Language Processing
NLTK	Natural Language ToolKit
NOAA	National Oceanic and Atmospheric Administration
PRMF	Parallel Risk Management Framework
RMF	Risk Management Framework

SAR	Search-and-Rescue
VID	Vessel in Distress

Chapter 1

Introduction

This chapter introduces maritime risk management. The reliance on human input is highlighted, which motivates the strive for the automation of future maritime solutions. Additionally, the scale of the relevant data is described. Following these introductory passages is the list of contributions contained in this work. The chapter ends with an overview of the manuscript structure.

1.1 Motivation

Risk management is an essential tool for ensuring the safety and timeliness of maritime operations. The real-time risk assessment of maritime vessels is largely enabled by the widespread adoption of automatic identification system (AIS). As of the year 2002, the International Maritime Organization (IMO), which is a United Nations (UN) agency responsible for regulating the shipping industry of its 174 member states, has required most commercial vessels to be fitted with an AIS transponder for safety purposes (e.g. collision avoidance)[1]. These devices periodically transmit updates about the vessel's status including its position, speed,

[1] `http://www.imo.org/en/OurWork/safety/navigation/pages/ais.aspx`

heading, etc. MarineTraffic, one provider of AIS data, receives 520 million AIS messages from 180,000 unique vessels each day[2] – and this number is increasing year by year. This massive quantity of AIS data can be fused with other information such as weather and piracy reports to provide real-time risk analysis for maritime vessels. Arguably, this makes real-time maritime risk management a big data problem. In terms of the so-called "big Vs" of big data: AIS messages certainly have high *velocity* and *volume*, and the additional data sources introduce *variety*. This implies that a risk management solution should be fast and efficient in order to meet current and future maritime data processing needs. Therefore, one objective of this thesis is to enable an existing Risk Management Framework (RMF) [3][4] to scale horizontally (i.e. by adding more processors to an existing system) in order to process large volumes of data (such as AIS) in real-time.

Performance concerns aside, the RMF relies on human operators and/or domain experts to configure various aspects of the risk management process. For example, a domain expert is required to specify the rules governing a fuzzy inference system in the RMF's Risk Assessment module. These rules are used to combine the individual risk levels coming from all risk features being monitored in the system into an overall risk value. However, relying on human input in this manner is generally time-consuming and error-prone. Therefore, the second goal of this research is to make the system as automated as possible, by learning from data instead of human input.

[2]https://www.marinetraffic.com/blog/a-day-in-numbers/

1.2 Contributions

The main contribution of this thesis is automating two key aspects of the Risk Management Framework:

- Genetic fuzzy systems (GFSs) have been integrated with the RMF's Risk Assessment module. This module, which combines individual risk values to overall risk levels, previously relied on domain experts to specify the fuzzy rule base (FRB) of a fuzzy inference system (FIS). GFSs allow the RMF to automatically adapt the FRB to the incoming data instead of requiting a domain expert to manually specify the rules.
- A suite of natural language processing (NLP) techniques has been developed to extract details about maritime incidents (i.e. piracy) from unstructured articles published by newspapers or magazines. This allows the RMF to passively learn about maritime incidents as they are reported, and adjust the *regional hostility* risk factor accordingly. By doing so, the RMF is endowed with an automated learning component from textual data sources.

To support these contributions, the following improvements to the Risk Management Framework have also been made:

- The design and implementation of a parallel version of the RMF's Risk Feature Extraction and Risk Assessment modules. This parallel RMF (PRMF) has been implemented in MPI (C++) and Apache Spark (Java), and has been shown to scale linearly with the number of processors allocated to it. This enables the RMF to quickly process large volumes of incoming data.
- The development of a new *weather risk* feature for the RMF. This risk feature uses weather reports to determine the risk incurred by a vessel's local weather conditions.

- A maritime visualization system built with Cesium[3]. This provides a visual representation of all of the above contributions and improvements. The horizontal scalability of the PRMF is demonstrated by visualizing large amounts of calculated risk values. The automated aspect of the system is shown both by visualizing the adaptive risk features as well as by visualizing the other adaptive aspects of the system (e.g. maritime incidents). Furthermore, the visualization system serves as a prototype of a decision support system with the RMF at its core. Future version of Larus Technologies' Total::Insight could use this visualizer (or something similar) to support decision makers.

1.3 Outline

The rest of this thesis is structured as follows. Chapter 2 covers the relevant background material and briefly reviews the related literature. Chapter 3 describes how GFSs are used to automate the maritime risk assessment process. Chapter 4 reveals a suite of NLP techniques to extract details about maritime incidents from unstructured articles. Chapter 5 introduces the parallel RMF and the new *weather risk* feature. Chapter 6 unveils a maritime visualization system. Finally, Chapter 7 concludes the thesis and discusses potential future research directions.

[3] `https://cesiumjs.org/`

Chapter 2

Technical Background and Related Works

This chapter explores the technical background necessary for the following chapters. Section 2.1 introduces maritime domain awareness (MDA). Section 2.2 unveils the Risk Management Framework (RMF) which is the starting point of this work. Section 2.3 deals with genetic fuzzy systems (GFSs) which are used in Chapter 3. Section 2.4 introduces the natural language processing (NLP) techniques that are relevant for Chapter 4. Section 2.5 reviews the concepts of parallel and distributed computing: both of which are featured in Chapter 5. Finally, Section 2.6 briefly reviews several relevant visualization techniques which will be expanded upon in Chapter 6.

2.1 Maritime Domain Awareness

Maritime domain awareness (MDA) is defined as the situational understanding of activities that impact maritime security, safety, economy, or the environment.

One goal of MDA is to effectively coordinate assets to respond to illegal activities, disaster situations, and rescue scenarios in the maritime domain [5].

One important component of MDA is risk management. ISO 8402:1995/BS 4778 defines risk as: "*A combination of the probability, or frequency, of occurrence of a defined hazard and the magnitude of the consequences of the occurrence*". In the maritime domain, risk is broadly understood as any factor that can hinder the timeliness of maritime operations or threaten the safety of crew, passengers, equipment, or cargo. A complete risk management strategy typically involves the following actions: (1) *identifying risk factors* (to be aware of the present hazards), (2) *assessing risk factors* (often as a combination of probability and impact), (3) *controling* (to mitigate the risks that are not tolerable), and (4) *reviewing* (to monitor the effectiveness of the controls) [6].

An effective risk management system uses a variety of data sources to refine the situational picture for an operator and/or decision maker. A key example of such data is automatic identification system (AIS) messages. Most merchant ships are required to be equipped with an AIS transponder, which periodically transmits key information about the vessel's status such as position, heading, and speed. This information can be complemented with additional data such as weather and piracy reports to provide a rich view of any AIS-transmitting vessel's risk status. However, an effective risk-centric MDA solution requires large volumes of data and significant computational resources.

Many techniques have been put forward to handle the big data problem (Section 1.1) introduced by MDA. Hidden Markov models [7] [8] have been proposed to monitor risk in networks. However, the authors of [9] point out that hidden Markov models can be unstable in dynamic situations such as MDA. In particular, if several interconnected hidden Markov models are used in maritime monitoring, a single change in the environment would require all models to be updated. This quickly becomes prohibitively expensive.

The study in [10] applied Bayesian belief networks to assess risk for vessels in the approach channel of the Tianjin port. This work identified areas where traffic was being managed inefficiently. However, the approach was tested on a dataset of only 234 collision reports. Furthermore, the timeliness of the calculation was not reported.

Recently, the authors of [11] used genetic programming and linear scaling along with AIS data to perform vessel path prediction. This approach was shown to outperform two different versions of genetic programming as well as three non-evolutionary algorithms.

Other projects that deal with MDA, tracking, risk detection, risk analysis, and risk management include:

- A Risk Management Framework (RMF) for the risk-driven multi-criteria decision analysis (MCDA) of various maritime situations, including the automatic generation of responses to incidents such as a vessel in distress (VID) [12] [13].
- Raytheon's ATHENA Integrated Defense System (IDS), [14] which merges data sources such as AIS and ship databases to perform threat evaluation, search for suspicious behaviours, support search-and-rescue operations, etc.
- The Predictive Analysis for Naval Deployment Activities (PANDA) [15] case-based reasoning system that uses contextual-based risk assessment that relies on a human-generated risk ontology.
- Sealink Advanced Analysis (S2A), previously known as the Maritime Automated Super Track Enhanced Reporting (MASTER) integrative reporting project based on the Joint Capability Technology Demonstration (JCTD) and the Comprehensive Maritime Awareness (CMA) [16]. S2A primarily provides vessel track correlation, cargo information, and anomaly detection.

With the exception of the RMF, these systems are commercial or classified projects, therefore the details of the underlying algorithms are not disclosed to the public. Furthermore, the RMF features a strong emphasis on risk management. For these reasons, the RMF was selected as the central framework for this thesis.

2.2 Risk Management Framework

The Risk Management Framework (RMF), introduced by Larus Technologies, is a modular system that enables real-time risk management in distributed systems of system units. The RMF was first proposed by Falcon et al. in [3] and was later expanded in [4] [9]. An example of a relevant domain for the RMF is a system of cameras surveying a secure perimeter: here, the RMF monitors the risk induced by the battery levels of the cameras and the proximity of an intruder. Based on the risk levels assigned to the cameras, a mobile robot is tasked to replenish the battery levels of each camera or to reposition itself to better detect the intruder.

The architecture of the RMF is shown in Figure 2.1. The main components of the RMF are the following:

Risk Feature Extraction Takes the raw data collected by the system units and assigns each unit one or more risk values (i.e. numbers in [0,1] that measure the extent to which a particular risk factor applies to a unit). The raw data can be in an arbitrary format such as textual, numerical, or video as long as a mapping between the given format and a numerical risk value can be defined. For example, numerical inputs are typically mapped to risk values by modeling them as fuzzy sets and properly defining their corresponding fuzzy membership functions. Therefore, this module is mainly driven by a domain expert, who defines the fuzzy sets and fuzzy membership functions.

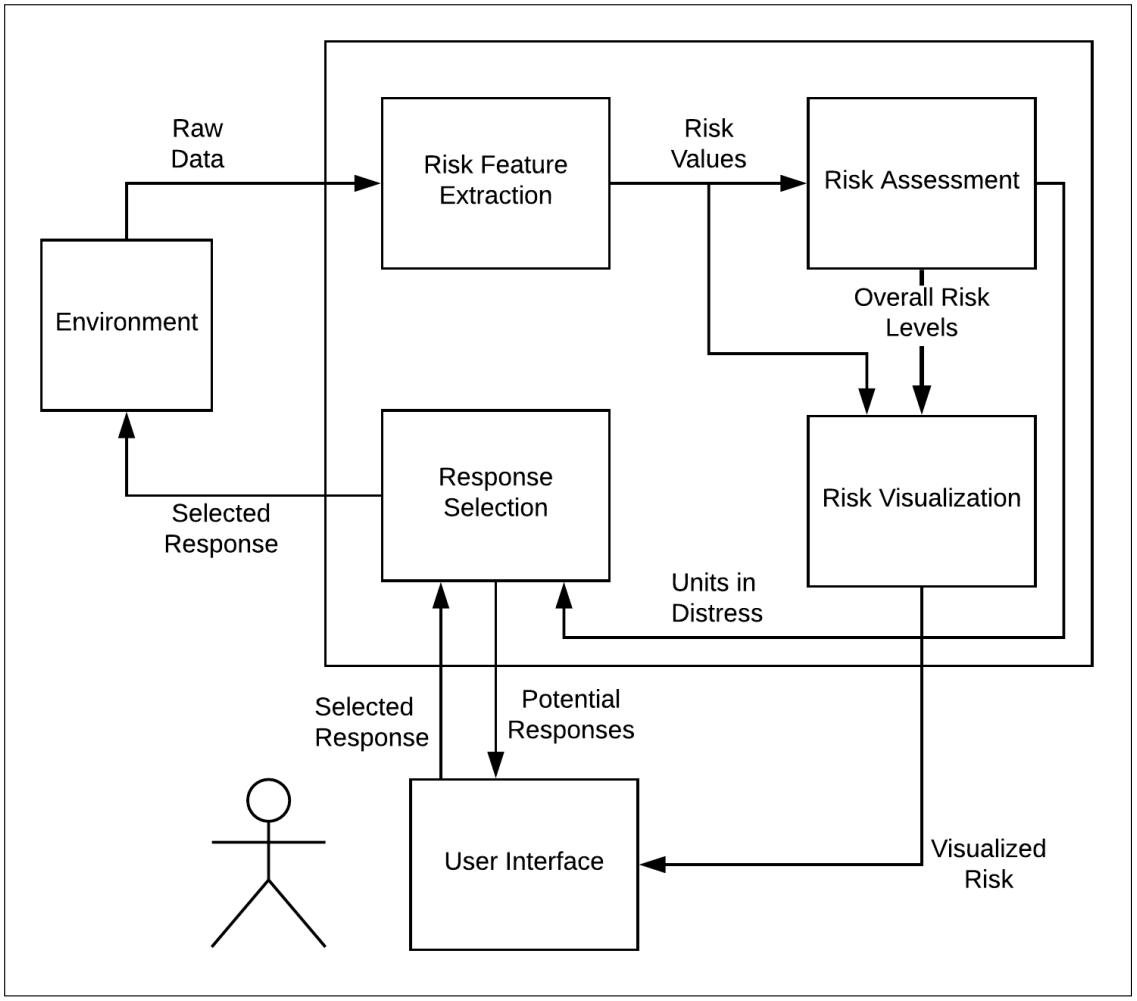

FIGURE 2.1: Risk Management Framework architecture

Risk Assessment Aggregates the risk values assigned to each unit, and assigns an overall risk level to each unit. This is accomplished with a fuzzy inference system (FIS) whose fuzzy rule base (FRB) is defined by a domain expert. In Chapter 3 of this thesis, the reliance on a domain expert is alleviated with the help of GFSs.

Risk Visualization Provides a visual representation of the risk landscape as defined by the previous modules. This module could directly produce graphs or charts to represent risk, or it could produce output to be read by other visualization tools. In some domains this module may also provide insight

about the relationship of the units, for example by clustering units which face similar risk conditions.

Response Selection Generates potential responses to mitigate risk in the environment. These responses could involve changing the state of the system units or deploying assets that have the ability to reduce risk. Since responses will often have several conflicting qualities such as effectiveness, timeliness and cost, they are typically generated through multi-criteria decision analysis (MCDA) algorithms. These algorithms generate sets of responses which offer a tradeoff between the relevant qualities, but do not offer any response that is inferior to any other across all of the qualities. The potential responses are presented to a human operator who makes the final decision about which response to enact, if any.

As human operators enact responses from the Response Selection module, the risk landscape of the environment changes accordingly. The system units detect the changes in the environment resulting in a changed set of inputs to the Risk Extraction module. This means that the RMF is a closed loop system: the decisions made in the Response Selection module affect the subsequent data received by the Risk Feature Extraction module.

2.2.1 Risk Management Framework for Maritime Domain Awareness

The RMF was first applied to the maritime realm in [4]. In this domain, the system units are vessels at sea, which transmit their status (e.g position, heading, speed) via AIS transponders. These transmission are collected by satellites and onshore radar, and are complemented by additional data sources such as weather and piracy reports. On the basis of these data, the following risk features are evaluated for each vessel:

Weather Risk Harsh weather conditions can pose a serious risk to the safety of a ship's passengers, crew, and cargo. However, in previous works, no source of weather data was identified. Instead, the values of the weather risk (then called "*sea state*") feature were randomly generated for illustrative purposes. In Chapter 5 of this thesis, a suitable source for weather data has been identified and integrated with the RMF.

Regional Hostility Certain regions of the world tend to see hostility activity by bad actors such as pirates. Navigating in the areas where these maritime incidents occur is inherently risky to the vessel's safety. In [17], the authors propose a methodology to learn about the occurrence of maritime incidents by applying NLP techniques to the *National Geospatial-Intelligence Agency*'s (NGA) *Worldwide Threats to Shipping* reports. In [9], these techniques are integrated with the RMF's *regional hostility* risk factor. This thesis extends these efforts in Chapter 3 by proposing a method to learn about the maritime incidents by ingesting unstructured articles from magazines and newspapers.

Collision Factor Vessels navigating near one another run the risk of colliding. The authors of [4] proposed using the self-reported position of all the vessels to determine how close each vessel is to the nearest other ship. This is used as the basis to calculate the risk of collision for each vessel.

Degree of Distress Indicates the severity of the potential sinking of the vessel. In [4], it was suggested that this should be a function of the vessel's fuel level, the number of people onboard, as well as the potential impact to the environment (e.g. an oil spill caused by the sinking of an oil tanker). One issue with this formulation is that there is no way to know a vessel's fuel level or passenger count since this information is not transmitted over AIS (or any other known data source). The work of [9] extended this risk feature by adding another component to the *degree of distress*: the probability of the vessel being targeted by pirates according to its type.

In [4], the RMF's response selection module is used to generate search-and-rescue (SAR) plans to mitigate risk. When a vessel's overall risk level exceeds an acceptable threshold, the vessel is deemed a *vessel in distress* (VID). Such vessels must be assisted in a SAR. This mission involves selecting one or more SAR assets which come to the aid of the distressed ship. The RMF generates plans by selecting the assets that will be part of the mission and planning the route that they will use to reach the VID. The potential plans are evaluated according to their timeliness, cost, and casualty probability. The best plans are presented to the user, who selects a plan to be enacted.

The RMF has been shown to be a suitable framework for managing risk in the maritime domain. It can model the relevant risk features and assess the risk levels of any AIS-reporting vessel. Additionally, it can generate responses (e.g. SAR plans) to mitigate risk for particular vessels. However, there are still many improvements that can be made to the RMF's ability to manage maritime risk, which will be explored throughout the rest of this thesis.

2.3 Genetic Fuzzy Systems

Fuzzy inference systems (FISs) use fuzzy logic to map input features to class outputs. Typically, FISs rely on fuzzy membership functions to map numerical inputs to degrees of membership to linguistic variables modelled as fuzzy sets along with FRBs to accomplish this. The two most common types of FISs are Mamdani [18] and Sugeno [19]. The main difference between these is that the consequent of Mamdani FIS rules are fuzzy sets, whereas in Sugeno FISs the rule consequents are polynomial expressions. Both types of FISs will provide a numerical output back to the user, which reflects the decision variable of interest in the problem under consideration. Figure 2.2 shows the architecture of a typical GFS.

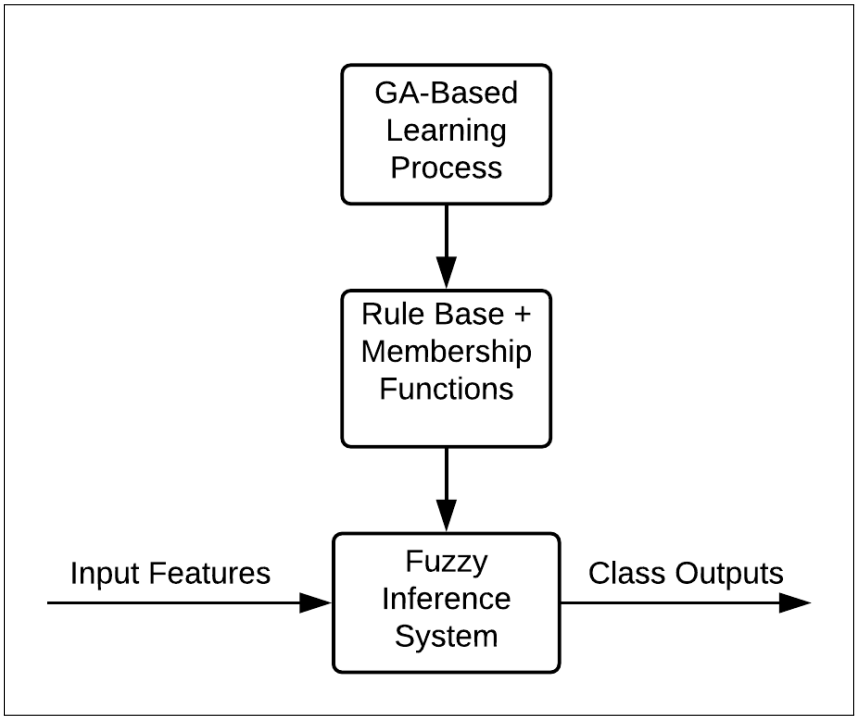

FIGURE 2.2: Typical genetic fuzzy system

Introduced in 1992 with the publication of [20], genetic fuzzy systems (GFSs) are computational models for automatically learning the FIS membership functions' parameters directly from data. In this work, a genetic algorithm (GA) is used to optimize the parameters of the FIS, with the objective of finding membership function parameters that emulate a known fuzzy logic controller. This first version of GFS is technically considered an example of reinforcement learning.

The same year (i.e. 1992) saw the introduction of the Michigan approach for GFSs [21]. The Michigan approach typically optimizes the FIS rule base. Each individual in the genetic population represents a single rule, and the entire population represents the rule base. This creates a fascinating contradiction. In GA terms, the individuals in the population are competing with each other to survive based on the natural selection principles that GAs are built upon. Yet from the FIS perspective, the individuals in the population are cooperating together to collectively form a good rule base. Therefore the individuals are both competing with and

cooperating with each other, a contradiction that is referred to as the "*cooperation vs. competition problem*" (CCP) [22].

The Pittsburgh approach of GFSs was introduced with [23]. This approach is suitable for optimizing the FRB and/or membership functions. Each individual in the population encodes the entire set of rules and/or membership functions, and the population is a set of candidate rule base/membership functions. This scheme implies that the individuals in the population are competing against each other and not cooperating with each other, which resolves the CCP seen with the Michigan approach. The drawback of this method is that the individuals contain much more information, which drastically increases the size of the search space. This can make it difficult to obtain optimal solutions.

Another common family of GFSs is known as iterative rule learning (IRL) approaches. As with the Michigan approach, the IRL approach models each individual as a single rule. As implied by the name, the IRL is an iterative approach. In each iteration, a set of rules are generated with a GA but only the best one (i.e. the one covering the most training set examples) is added to the final population. Between each iteration, the training set examples already covered by a rule are removed so that subsequent rules are more likely to cover the remaining samples. This process repeats until all the samples are covered. Since the rules generated in later stages are unaware of the previously removed samples, it is possible that they will contradict the earlier rules. This is a variation of the CCP.

Similar to the IRL approach is the boosting approach. Generally, boosting refers to constructing a good classifier as a weighted combination of many weak classifiers. This concept can be applied to GFSs by considering the fuzzy rules themselves as weak classifiers. A boosting GFS then determines the weight to assign to each rule. The boosting approach addresses the CCP by never discarding any samples with the intention that pairs of contradicting rules will not both be assigned large weights.

Since their inception, GFSs have been extensively studied [24] and applied to a wide variety of domains including medicine [25][26], finance [27] [28], industrial/-manufacturing [29][30], and many others. For further reading, [31] is a recent survey on the state-of-the-art of GFSs.

KEEL (Knowledge Extraction based on Evolutionary Learning) [32] [33] is a GUI-based software tool that contains a large range of computational intelligence algorithms. In particular, it contains several implemented GFSs. These GFSs are used for the experiments of Chapter 3.

2.4 Natural Language Processing

This section reviews the NLP techniques that are employed in Chapter 4 to automatically learn about maritime incidents from unstructured articles in magazines and newspapers.

2.4.1 Document Classification

Document classification entails assigning documents to one or more pre-defined categories. The authors of [34] provide an overview of automated document classification. They note that documents must be mapped to compact representations in order to be interpreted by a classification algorithm. The most common technique is the bag-of-words approach which converts documents to vectors of term weights, representing the set of words across all documents. Two variation of the bag-of-words approach are mentioned: in the *binary* version, each term is a binary value indicating the presence or absence of a word in the document; in the *frequency* version, each term is an integer representing the number of occurrences of a word in the document. The authors also mention that dimensionality reduction can increase accuracy and reduce overfitting.

2.4.2 Information Extraction

Information extraction refers to automatically extracting structured information from unstructured or semi-structured machine readable text [35]. Current information extraction techniques include rule-based methods [36] such as regular expressions, statistical methods such as conditional random fields (CRF) [37], and NLP techniques such as relationship extraction [38] and named entity recognition (NER). Surveys of information extraction techniques include [39] and [40].

In [41], the authors describe a NLP approach for extracting the location of crimes from newspaper articles. The approach uses NER to identify the names of cities, suburbs, or streets where crimes occur. Each sentence is tagged with ten features (e.g. whether the sentence has a crime-related keyword, whether the sentence contains the name of a city, etc.) as well as a binary class label indicating if the sentence mentions a crime location. A CRF was then trained to learn the relationship between the features and the class label. Despite its high accuracy, it is unclear how the location of the crime is ultimately determined when an article's crime location sentences mention multiple locations.

In [42], the authors extracted information about natural disasters reported in online newspapers. The proposed system classifies the articles according to the type of incident (e.g. hurricane, forest fire) using standard document classification techniques. It also uses regular expressions to identify proper names, quantities, and dates then matches these details to categories (e.g. location of incident, number of casualties) with classification algorithms.

The NLP approach used in Chapter 4 of this thesis uses several of the document classification and information extraction notions mentioned above. The proposed approach uses the standard document classification technique of converting the documents to bags-of-words, selecting a subset of the features, and training a classifier to assign the document to a category. The information extraction techniques

uses regular expressions and NER to detect location mentions and contextual cues to select the final answer among all extracted facts.

2.5 Parallel and Distributed Computing

Parallel and distributed computing [43][44][45] are techniques for dividing a computational workload into smaller tasks that can be processed concurrently. The goal of this is to reduce the overall amount of time to complete the workload.

Parallel computing systems utilize multiple processors (or multicore processors) that typically share a pool of memory. This shared memory enables rapid communication between the processors. However, there is a practical limitation to how many processors can share a pool of memory in that adding more processors increases traffic on the communication channel. Care must be taken to ensure that the processors do not interfere with each other's memory.

Distributed computing involves multiple processors that communicate primarily by message passing (e.g. over ethernet). The processors can be distributed geographically and do not physically share memory. This enables the system to scale to arbitrarily large numbers of processors. Furthermore, each processor can access its own memory without interfering with another processor.

In Chapter 5, this work makes use of two popular parallel and distributed frameworks: Apache Spark [46] and MPI [47]. MPI (Message Passing Interface) is a low level library for parallel and distributed computing. In MPI, all parallel algorithms must be explicitly implemented by the programmer down to the communication level: each send and receive must be manually specified. Apache Spark is a data processing engine for distributed and parallel computing. Spark is higher level

than MPI, requiring the programmer to specify operations on datasets but handling the parallelization of the operations under the hood. It also has an emphasis on fault tolerance.

2.6 Maritime Visualization

The goal of a maritime visualization system is to efficiently represent a maritime situation in an intuitive and informative way. The system should display enough information to sufficiently inform the operators and/or decision makers about the maritime situation without overwhelming them. Such systems will likely display a map of the world with the positions of relevant vessels indicated. Additional information about the vessels such as their name, type, heading, or speed might also be relevant, although it is less clear how these data should be displayed without cluttering the display. Furthermore, it might be desirable to display the locations of other maritime points of interest such as ports, exclusive economic zones, weather patterns, etc.

There are many general visualization tools which have the potential to be configured to display maritime data. However, these may require significant amounts of effort to configure and the final result may not be as effective as a more specialized tool. Examples of these include D3[1], Tableau[2], and Omiscope[3].

Some visualization tools are explicitly designed for geovisualization. Such tools can be configured to display maritime data with minimal effort and the result can be quite effective. These include Google Earth[4], Cesium[5], Geotime[6], and ArcGIS[7].

[1] `https://d3js.org/`
[2] `https://www.tableau.com/`
[3] `http://www.visokio.com/`
[4] `https://www.google.com/earth/`
[5] `https://cesiumjs.org`
[6] `https://geotime.com/`
[7] `https://www.arcgis.com/index.html`

Finally, there are several commercial products that are explicitly designed to display maritime data. These include MarineTraffic[8] and ExactEarth's ShipView[9]. The main interfaces of these products are dominated by large interactive maps of the world. The position, heading, and type of all known vessels are represented as colored arrows on the map. Side panels allow data to be filtered and additional information to be displayed. Additional pages give more details about specific ships, ports, etc.

These visualization tools set the stage for the maritime risk visualization system presented in Chapter 6. Cesium was selected as the system's underlying platform since it is intended for geospatial mapping and also has great flexibility and performance. The risk visualization system features many of the elements seen in the commercial maritime visualization tools such as a large interactive maps displaying the positions of vessels additional pages/panels with additional information. However, the proposed system has a focus on displaying risk which is not seen in any of the previously mentioned products.

2.7 Chapter Summary

This chapter reviewed the background and key concepts that are relevant for this thesis. Maritime domain awareness (MDA) was introduced, setting the state for the rest of the thesis. The next section discussed the Risk Management Framework (RMF). The next few sections reviewed the concepts used to improve the RMF: namely, genetic fuzzy systems, natural language processing, and parallel and distributed computing. Finally, a brief overview of maritime visualization techniques was presented.

[8] `https://www.marinetraffic.com/`
[9] `https://shipview.exactearth.com/`

Chapter 3

Automatic Maritime Risk Assessment

This chapter uses genetic fuzzy systems (GFSs) to assess the risk level of maritime vessels that transmit automatic identification system (AIS) messages. In previous versions of the Risk Management Framework (RMF), the Risk Assessment module's fuzzy inference system (FIS) relied on domain experts to specify the FIS membership functions as well as the fuzzy rule base (FRB), a burdensome and time-consuming process. This chapter aims to alleviate this burden by learning the membership functions and FRB directly from data.

3.1 Introduction

The RMF put forth in [3] [12] makes use of a FIS to combine the value of several risk factors into a single overall risk level. The FRB powering the FIS is directly acquired from domain experts. The process of consulting with domain experts is time consuming and error prone. It would be desirable to avoid this step.

In this chapter, the reliance on domain experts is alleviated with the help of a GFS, which learns the rule base directly from data. The proposed methodology is illustrated with four case studies in maritime risk analysis. In each of these scenarios, fourteen GFS algorithms available in KEEL [32] [33] have been tested. The performance of each GFS was evaluated according to their accuracy and interpretability. The experimental results indicate that IVTURS, LogitBoost, and NSLV generate the most accurate rule bases while SGERD, GCCL, NSLV, and GBML each generate highly interpretable rule bases. Meanwhile, IVTURS, NSLV, and GBML algorithms offer a reasonable compromise between accuracy and interpretability.

The rest of this chapter is structured as follows: Section 3.2 unveils the proposed methodology to automate maritime risk assessment (MRA). Section 3.3 outlines the experimental procedure and reveals the experimental results. Finally, Section 3.4 concludes the chapter.

3.2 Proposed Approach

In order to apply GFSs to MRA, the latter is modeled as a classification problem. The input features describing each AIS message transmitted by a vessel are a set of risk features, i.e. numerical attributes in the range [0,1] that quantify the extent of a particular risk for the vessel. The decision classes represent the overall risk assessment which can be *low*, *medium*, or *high* risk.

The overall architecture of the proposed methodology is shown in Figure 3.1. The remainder of this section explores the components of the experiment's architecture.

3.2.1 Data Sources

The data for the experiments originates from the following sources:

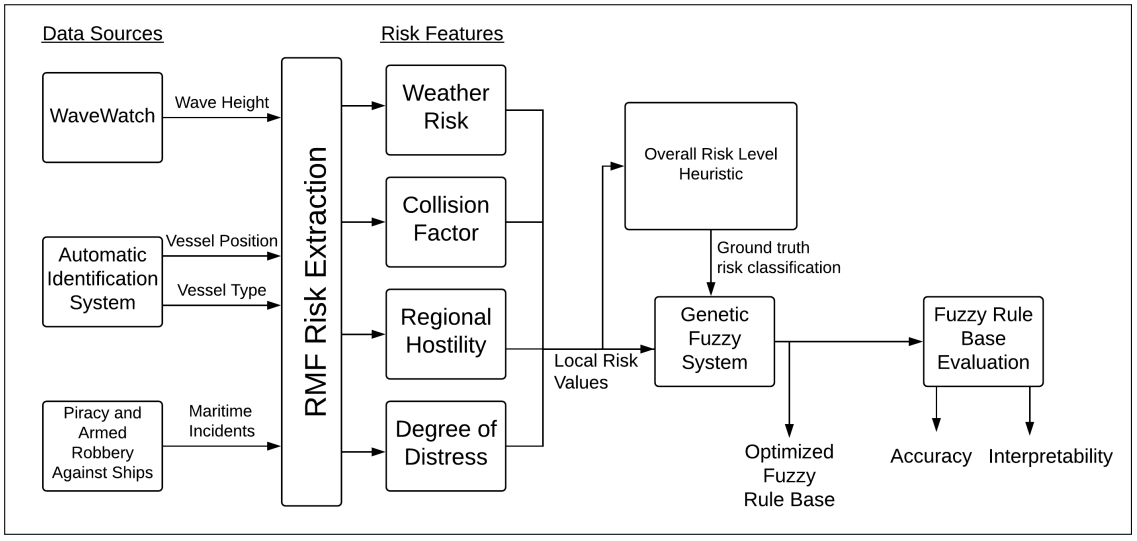

FIGURE 3.1: Proposed architecture to automate MRA with GFSs

AIS Data The data includes two full days of AIS messages from Orbcomm[1] (i.e. January 1 2018 and January 13 2018). This provides more than enough data to train the genetic algorithms, therefore the messages are sampled as specified in Section 3.3.2. Among the fields available in the AIS messages, only the *latitude*, *longitude*, and *ship type* are used.

Weather Data As specified in Section 5.2, this experiment uses weather data from WaveWatch III.

Maritime Incident Reports From the International Maritime Bureau's (IMB) 2017 Piracy and Armed Robbery Against Ships report[2]. This report lists maritime incidents that occur throughout the world in a semi-structured format from which the date/time, location, type of vessel attacked, and type of incident are extracted.

[1] `https://www.orbcomm.com/`
[2] `https://www.icc-ccs.org/`

3.2.2 Risk Features

The data from Section 3.2.1 is used as inputs for the RMF's Risk Feature Extraction module. This produces the four risk features described in Section 2.2.1: namely, *weather risk*, *collision factor*, *regional hostility*, and *degree of distress*. These risk features are the input features by which the GFS will classify the vessel as either low risk, medium risk, or high risk.

3.2.3 Ground Truth

For each set of risk values, a ground truth overall risk level is assigned to train the GFS. In this work a simple heuristic is used to generate the ground truth, but in practice the ground truth could be determined by consulting a domain expert or any other method. The simple heuristic assigns a ground truth overall risk level with the following scheme:

$$\text{Ground Truth} = \begin{cases} \text{HIGH-RISK}, & \text{if at least one risk value is in [b,1]} \\ & \text{OR} \\ & \text{at least two risk values are in [a,b)} \\ \text{LOW-RISK}, & \text{if all risk features are in [0,a)} \\ \text{MEDIUM-RISK}, & \text{otherwise} \end{cases}$$

with a=0.4 and b=0.7.

3.2.4 Genetic Fuzzy Systems

GFSs are used to generate FRBs that describe the relationship between the four RMF risk features (Section 3.2.2) and the ground truth overall risk level (Section 3.2.3). In total, fourteen different GFS algorithms are tested. Table 3.1 lists the algorithms and some of their differences.

Regarding the "*GFS Type*" column, the algorithms generally fall into the categories discussed in Section 2.3. However, some differences can be noted. GBML uses a unique hybridization of the Michigan and Pittsburgh approaches: a Pittsburgh-style step generates a population of rule bases, and a Michigan-style step is used to optimize the generated rule bases. SP algorithm utilizes simulated annealing with genetic programming inspired operators; the authors claim that this approach achieves similar results to genetic algorithms but with a lower memory footprint. IVTURS uses interval value fuzzy sets to represent the membership functions of the fuzzy sets.

The "*Interpretable Rule Base*" column indicates whether or not the algorithm generates a human-readable rule base (e.g. this is not the case for ensemble-based methods).

In the "*Number of Linguistic Terms Per Variable*" column, a value of "*User-Specified*" indicates that a parameter is available for the user to tune, a value of "*Learned*" indicates that the parameter's value is learned from data, and a value of "*Fixed*" indicates that the algorithm always uses the same value.

As for the "*Parameters*" column, the default value of each parameter is used unless the algorithm requests more than 1,000 iterations or fitness function evaluations.

3.2.5 Performance Metrics

Each rule base was evaluated by two metrics:

TABLE 3.1: GFS algorithms compared

Algorithm	**GFS Type**	**Interpretable Rule Base**	**Number of Linguistic Terms per Variable**	**Parameters**
COACH [48]	Michigan	Yes	User-Specified	Max Iterations=1000
GCCL [49]	Michigan	Yes	User-Specified	Max Evaluations=1000
SGERD [50]	Michigan	Yes	Learned	Default
GBML [51]	Michigan/Pittsburgh Hybrid	Yes	Learned	Default
GP [52]	Pittsburgh	Yes	User-Specified	Max Iterations=1000
GPG [52]	Pittsburgh	No	User-Specified	Max Iterations=1000
IVTURS [53]	Pittsburgh	Yes	Learned	Max Iterations=1000
SP [52]	Pittsburgh	No	User-Specified	Max Iterations=1000
NSLV [54]	Iterative Rule Learning	Yes	Fixed	Default
Slave2 [54]	Iterative Rule Learning	Yes	Fixed	Default
SlaveV0 [54]	Iterative Rule Learning	Yes	Fixed	Default
LogitBoost [55]	Boosting	No	User-Specified	Default
MaxLogitBoost [56]	Boosting	No	User-Specified	Default
AdaBoost [57]	Boosting	No	User-Specified	Default

F-Measure To evaluate an FRB's ability to correctly determine each contact's risk level, the well-known F-measure metric is employed.

Total Rule Length Is a useful tool for measuring the complexity of a FRB. It is defined as the sum of the number of conditions in each rule [58]. This implicitly takes into account both the number of rules and the number of conditions in each rule.

The ideal FRB should have high F-measure and low total rule length. Note that although two objectives are considered, the GFSs themselves may not be dual-objective optimization algorithms; most have the sole objective of maximizing accuracy.

3.2.5.1 Statistical Analysis

The well-known nonparametric Friedman test was employed to rank the performance of the algorithms. Although the Friedman test can rank the algorithms, it does not guarentee a statistical significance between the groups. Therefore, the Nemenyi post-hoc test was used to test the statistical significance between the groups [59].

The Nemenyi tests allow us to arrange the algorithm into tiered groups, i.e. group "A", group "B", group "C", etc. All of the algorithms in group "A" are statistically better than the algorithms in group "B" and so on. However, an algorithm can be placed in more than one group. For example, a group of "AB" indicates that the statistical test could not confirm that the algorithm is inferior to any of the algorithms in group "A", nor could the test confirm that the algorithm is statistically superior to all of the algorithms in group "B". Therefore, it may belong to group "A" or to group "B".

3.3 Experiments

This section outlines the experiments used to test the proposed approach.

3.3.1 Case Studies

This experiment involves four case studies in maritime surveillance, each concerning a specific area of interest (AOI) and period of interest (POI). Each case study was designed to capture a different risk landscape (i.e. prevalence of certain risk features) in order to test the GFSs abilities to learn the distinctive features in different scenarios.

The first AOI is the Gulf of Guinea (min latitude=-20, max latitude=7, min longitude = -7, max longitude=15) with a POI of January 1 2018 00:00:00-January 1 2018 23:59:59. It is expected that this region's risk will primarily arise from *regional hostility*: thirty-eight maritime incidents were reported in the AOI in 2017 (Figure 3.2). Amongst the victims whose ship type could be determined, 47.3% were cargo ships, 44.7% were tankers, and 7.8% were utility vessels. Weather in this region is typically mild, as was the case for this POI.

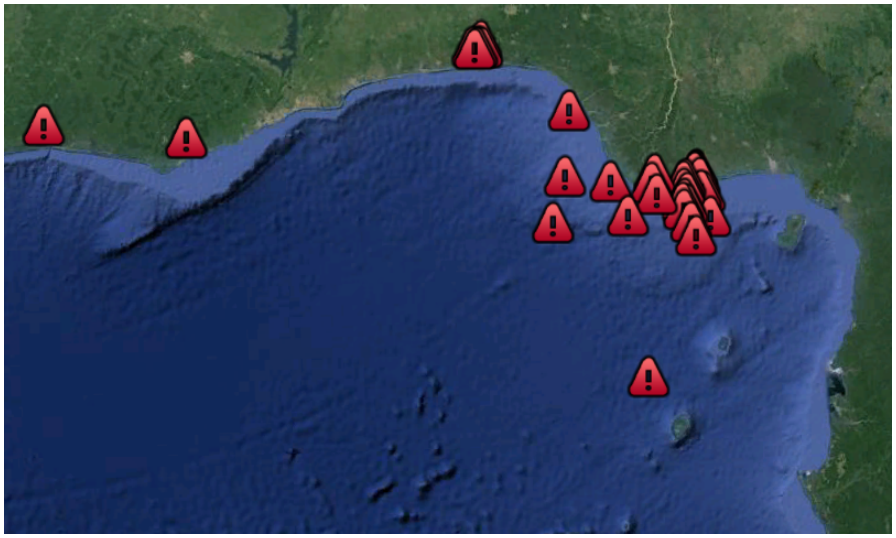

FIGURE 3.2: Maritime incidents, Gulf of Guinea, 2017

The second AOI concerns the Strait of Malacca (min latitude=-4, max latitude=9, min longitude=92, max longitude=110) with POI January 1 2018 00:00:00-January 1 2018 23:59:59. Not only is the Strait of Malacca one of the world's busiest maritime traffic lanes, it is also one of the narrowest: 1.5 nautical miles at its narrowest point. This, combined with the steady growth of traffic within the strait make it a potentially dangerous area to navigate. Indeed, 60 ship accidents were reported to Maritime and Port Authority Singapore in 2015 [60]. Therefore, *collision factor* is expected to be a significant contributor risk in the Strait of Malacca. Additionally, 37 maritime incidents occurred in the AOI in 2017 (Figure 3.3). Finally, weather conditions were mild in this AOI/POI.

The third and fourth scenarios each concern the same AOI: a northern stretch of the Atlantic ocean (min latitude=35, max latitude=60, min longitude=-50, max longitude=0) with two different POIs: January 1 2018 00:00:00-January 1 2018 23:59:59 ("Atlantic Storm" scenario) and January 13 2018 00:00:00-January 13 2018 23:59:59 ("Atlantic No-Storm" scenario). The Atlantic Storm scenario takes place during a harsh weather event in the Atlantic (Figure 3.4), which is expected to result in elevated *weather risk* values. In the Atlantic No-Storm scenario the

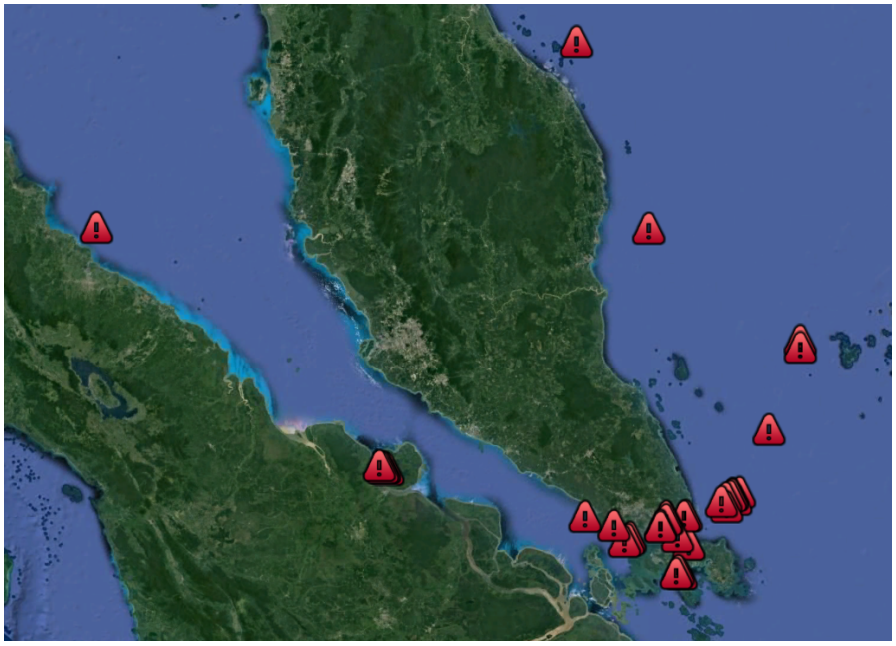

FIGURE 3.3: Maritime incidents, Strait of Malacca, 2017

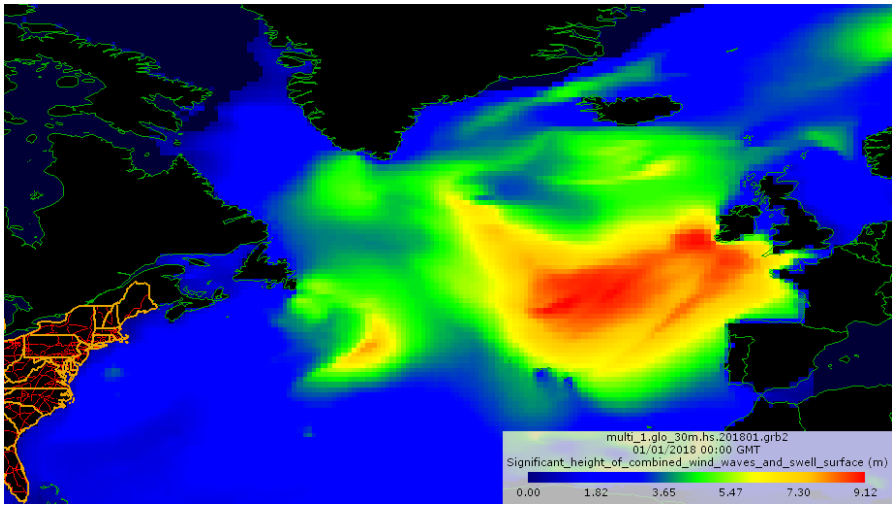

FIGURE 3.4: Wave height, North Atlantic Ocean, January 1 2018 0:00:00GMT

weather is much milder (Figure 3.5). No piracy activity was recorded in this region in 2017.

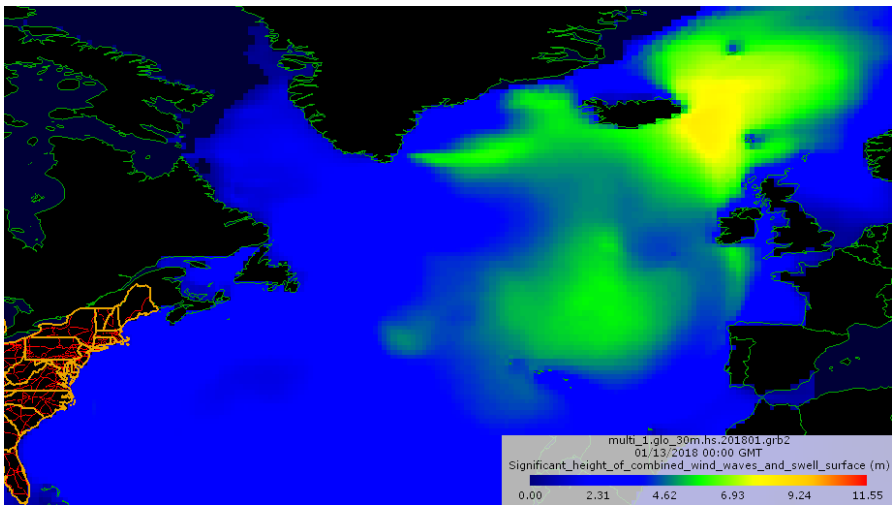

FIGURE 3.5: Wave height, North Atlantic Ocean, January 13 2018 0:00:00GMT

3.3.2 Experimental Setup

For each case study mentioned in Section 3.3.1, 1,000 vessels from the dataset were randomly chosen, and a single AIS message was selected from each of these vessels. This means that each of the four scenario's inputs are made up of 1,000 AIS contacts and the complete dataset contains 4,000 records. For each of these messages, the *latitude*, *longitude*, and *ship type* fields are used. The input features are determined as in Section 3.2.2 and the ground truth is determined as in Section 3.2.3.

The KEEL master branch was checked out from source control[3] to perform the experiments. All of the algorithms listed in Table 3.1 were tested. Each experiment was repeated 30 times using a different random seed to account for the stochastic nature of the algorithms, and the average values are reported.

[3] `https://github.com/SCI2SUGR/KEEL`, checked out on 01/05/2018

TABLE 3.2: Accuracy results for the Guinea scenario

Algorithm	**Accuracy (F-Measure)**	**Friedman Rank**	**Nemenyi Group**
IVTURS	0.98	1.27	A
LogitBoost	0.98	2.4	AB
NSLV	0.98	3.03	ABC
GBML	0.98	3.7	ABCD
SLAVEv0	0.97	5.42	BCDE
SLAVE2	0.97	5.52	BCDEF
MaxLogitBoost	0.97	6.77	DEFG
AdaBoost	0.96	7.9	EFGH
COACH	0.92	9.0	EFGHI
SGERD	0.82	10.6	HIJ
GP	0.81	10.8	HIJK
SP	0.76	12.13	IJKL
GPG	0.75	12.47	IJKLM
GCCL	0.6	14.0	JKLMN

TABLE 3.3: Accuracy results for the Malacca scenario

Algorithm	**Accuracy (F-Measure)**	**Friedman Rank**	**Nemenyi Group**
IVTURS	0.99	1.2	A
LogitBoost	1.0	1.8	AB
NSLV	0.95	3.73	ABC
SLAVE2	0.95	4.43	ABCD
SLAVEv0	0.95	4.57	ABCDE
MaxLogitBoost	0.91	6.0	CDEF
GBML	0.9	6.27	CDEFG
GP	0.75	9.03	FGH
SGERD	0.72	9.7	FGHI
COACH	0.71	9.97	GHIJ
GPG	0.7	10.27	HIJK
SP	0.65	11.67	HIJKL
AdaBoost	0.64	12.37	HIJKLM
GCCL	0.29	14.0	LMN

3.3.3 Experimental Results

The results for accuracy are given in Table 3.2 for the Guinea scenario, Table 3.3 for the Malacca scenario, Table 3.4 for the Atlantic Storm scenario, and Table 3.5 for the Atlantic No-Storm scenario. The results for interpretability are given in Table 3.6 for the Guinea scenario, Table 3.7 for the Malacca scenario, Table 3.8 for the Atlantic Storm scenario, and Table 3.9 for the Atlantic No-Storm scenario.

TABLE 3.4: Accuracy results for the Atlantic Storm scenario

Algorithm	**Accuracy (F-Measure)**	**Friedman Rank**	**Nemenyi Group**
LogitBoost	0.99	1.0	A
MaxLogitBoost	0.94	2.1	AB
IVTURS	0.95	3.03	ABC
NSLV	0.93	4.73	BCD
SLAVEv0	0.93	5.08	BCDE
SLAVE2	0.93	5.12	BCDEF
COACH	0.8	7.2	DEFG
GBML	0.92	7.73	DEFGH
AdaBoost	0.85	9.03	GHI
SP	0.73	10.07	GHIJ
GP	0.76	11.63	IJK
GPG	0.74	11.73	IJKL
SGERD	0.74	12.53	IJKLM
GCCL	0.61	14.0	KLMN

TABLE 3.5: Accuracy results for the Atlantic No Storm scenario

Algorithm	**Accuracy (F-Measure)**	**Friedman Rank**	**Nemenyi Group**
LogitBoost	0.99	1.0	A
GBML	0.95	2.7	AB
IVTURS	0.93	4.0	ABC
SLAVE2	0.92	4.3	ABCD
SLAVEv0	0.92	4.53	ABCDE
NSLV	0.92	5.53	BCDEF
MaxLogitBoost	0.89	6.8	CDEFG
AdaBoost	0.91	7.3	CDEFGH
GP	0.84	9.3	GHI
SGERD	0.82	10.3	GHIJ
GPG	0.81	10.47	GHIJK
SP	0.71	12.33	IJKL
COACH	0.7	12.43	IJKLM
GCCL	0.62	14.0	JKLMN

TABLE 3.6: Interpretability results for the Guinea scenario

Algorithm	**Interpretability (Rule Length)**	**Friedman Rank**	**Nemenyi Group**
SGERD	7.22	1.08	A
GCCL	14.11	2.25	B
NSLV	21.21	3.91	C
GBML	22.88	4.33	CD
IVTURS	24.1	4.71	CDE
COACH	38.9	6.69	F
GP	49.1	6.87	FG
SLAVE2	49.05	8.2	H
SLAVEv0	50.9	8.47	HIJ

TABLE 3.7: Interpretability results for the Malacca scenario

Algorithm	**Interpretability (Rule Length)**	**Friedman Rank**	**Nemenyi Group**
SGERD	8.2	1.34	A
GCCL	10.12	1.93	AB
NSLV	16.07	3.18	C
IVTURS	21.23	4.33	D
COACH	26.8	6.03	E
GBML	27.11	6.04	EF
GP	51.29	7.74	G
SLAVEv0	34.18	8.06	GHI
SLAVE2	35.16	8.3	GHIJ

TABLE 3.8: Interpretability results for the Atlantic Storm scenario

Algorithm	**Interpretability (Rule Length)**	**Friedman Rank**	**Nemenyi Group**
SGERD	5.8	1.08	A
GBML	9.98	2.45	B
GCCL	12.13	3.11	BC
IVTURS	17.79	4.66	D
NSLV	21.82	4.97	DE
COACH	22.42	6.32	F
GP	46.88	7.04	FG
SLAVE2	36.63	8.21	H
SLAVEv0	36.55	8.57	HIJ

TABLE 3.9: Interpretability results for the Atlantic No Storm scenario

Algorithm	**Interpretability (Rule Length)**	**Friedman Rank**	**Nemenyi Group**
SGERD	5.76	1.23	A
GBML	8.85	2.49	B
GCCL	10.14	2.98	BC
COACH	14.2	4.52	D
IVTURS	14.76	4.9	DE
NSLV	21.03	6.26	F
GP	44.16	7.89	G
SLAVEv0	28.55	8.05	GHI
SLAVE2	30.79	8.64	GHIJ

In terms of accuracy, the top performers include IVTURS (A), LogitBoost (AB), and NSLV (ABC) in the Guinea and Malacca scenarios. For the Atlantic Storm scenario, the best results were obtained from LogitBoost (A), MaxLogitBoost (AB), and IVTURS (ABC). Finally, in the Atlantic No-Storm scenario the top algorithms are LogitBoost (A), GBML (AB), and IVTURS (ABC). In all of the scenarios, IVTURS and LogitBoost are each top performers.

In terms of interpretability, SGERD is the clear winner in all of the scenarios (A). GCCL is also a strong contender in the Guinea scenario (B), the Malacca scenario (AB), the Atlantic Storm scenario (BC), and the Atlantic No-Storm scenario (BC). NSLV algorithm performs well in the Guinea (C) and Malacca (C) scenarios but performs slightly worse in the Atlantic Storm (DE) and Atlantic No-Storm (F) scenarios. Finally, GBML has good performance in the Atlantic Storm (B) and Atlantic No-Storm (B) scenarios although its performance is less impressive in the Guinea (CD) and Malacca (EF) scenarios.

In terms of algorithms that achieve good accuracy and interpretability, there is no one clear answer. Although LogitBoost and MaxLogitboost provide top tier accuracy, their rule bases are not at all interpretable. On the other hand, SGERD consistently generates simple rule bases at the cost of low accuracy. The algorithms which seem to have the best compromise between the two objectives include IVTURS, NSLV, and GBML.

3.3.4 Characterization of Fuzzy Rule Base Per AOI

In section 3.3.1, it was anticipated that the FRBs generated for each scenario would differ significantly, corresponding to the unique risk landscape of each case study. To test this, consider how frequently each risk feature appeared as an antecedent of a fuzzy rule.

TABLE 3.10: Average distribution of risk features in rule conditions

AOI	**Weather Risk**	**Collision Factor**	**Regional Hostility**	**Degree of Distress**
North Atlantic No-Storm	0.21	0.4	0.09	0.30
North Atlantic Storm	0.27	0.34	0.09	0.30
Gulf of Guinea	0.17	0.21	0.29	0.33
Strait of Malacca	0.16	0.22	0.32	0.31

Table 3.10 shows the average probability that an antecedent will correspond to a particular risk feature. Across all of the case studies, the *degree of distress* risk factor consistently appears in roughly 30% of all conditions. In the Gulf of Guinea scenario, *regional hostility* (29%) and *collision factor* (21%) are both important risk features, whereas *weather risk* (17%) plays a slightly lesser role. The risk landscape in the Strait of Malacca is revealed to be similar to the Gulf of Guinea, although *weather risk* (16%) is slightly less important while *collision factor* (22%) and *regional hostility* (32%) are slightly more important. It is surprising that *collision factor* is not much more important in the Strait of Malacca given the vessel congestion in the AOI. This could be because the congestion of the Strait of Malacca and Gulf of Guinea were similar enough that the algorithms assigned a similar importance to *collision factor*. For the two Atlantic scenarios, *regional hostility* (9%) almost never appears in the rule base. As expected, *weather risk* is more important in the Atlantic Storm (27%) than in the Atlantic No-Storm (21%) and *collision factor* is more important in the Atlantic No-Storm (40%) than in the Atlantic Storm (34%).

3.3.5 Accuracy vs. Interpretability

In order to illustrate the difference between a highly accurate and a highly interpretable FRB, consider a rule base generated by IVTURS as well as one generated by SGERD. SGERD generated the following FRB:

1. *IF collisionFactor IS LOW AND regionalHostility IS LOW-MEDIUM THEN OVERALLRISK IS LOW*
2. *IF collisionFactor IS LOW AND degreeOfDistress IS LOW-MEDIUM THEN OVERALLRISK IS MEDIUM*
3. *IF collisionFactor IS MEDIUM-HIGH AND regionalHostility IS LOW-MEDIUM THEN OVERALLRISK IS HIGH*

IVTURS generated the following RB:

1. *IF weatherRisk IS LOW THEN OVERALL RISK IS LOW*
2. *IF weatherRisk IS VERY LOW THEN OVERALL RISK IS LOW*
3. *IF collisionFactor IS VERY LOW THEN OVERALL RISK IS LOW*
4. *IF collisionFactor IS LOW AND degreeOfDistress IS LOW THEN OVERALL RISK IS MEDIUM*
5. *IF collisionFactor IS MEDIUM AND degreeOfDistress IS LOW AND weatherRisk IS VERY LOW THEN OVERALL RISK IS MEDIUM*
6. *IF degreeOfDistress IS HIGH THEN OVERALL RISK IS HIGH*
7. *IF degreeOfDistress IS MEDIUM THEN OVERALL RISK IS HIGH*
8. *IF collisionFactor IS HIGH THEN OVERALL RISK IS HIGH*

Clearly the SGERD rule base is far simpler: it has fewer rules and fewer conditions. Indeed, in our experiments SGERD's rule bases contained an average of 6.75 conditions while IVTUR's rule bases contained an average of 19.47 conditions. However, this comes at the cost of accuracy: SGERD managed an average accuracy of 77.5% yet IVTURS achieved 96.2%.

3.4 Chapter Summary

In this chapter, GFSs have been applied to the problem of assessing the overall risk level of AIS-reporting maritime vessels. The GFSs automatically learn the FRB and membership functions for a FIS which assigns each AIS message emitted by a vessel one of three overall risk levels according to four individual risk features. The data sources include AIS records, weather reports, and maritime incident reports from three regions of the world: the North Atlantic, the Gulf of Guinea, and the Strait of Malacca.

The datasets were fed to fourteen GFS algorithms via the KEEL framework and the resulting FRBs were evaluated according to their accuracy (F-measure) and interpretability (total rule length). The experimental results indicate that IVTURS, LogitBoost, and NSLV generate the most accurate rule bases while SGERD, GCCL, NSLV, and GBML each generate highly interpretable rule bases. Finally, IVTURS, NSLV, and GBML algorithms offer a reasonable compromise between accuracy and interpretability.

It was also observed that the rule bases produced for each case study differed. The frequency with which each risk factor appears in the rules characterizes the unique risk landscape of each AOI.

Chapter 4

Automatic Identification of Maritime Incidents from Unstructured Articles

This chapter introduces a suite of natural language processing (NLP) techniques that are used to learn about maritime incidents by processing unstructured articles from magazines and newspapers as well as unstructured incident reports. These techniques allow the Risk Management Framework's (RMF) *regional hostility* metric to automatically adapt to changing geopolitical conditions over time.

4.1 Introduction

One key aspect of maritime safety is the awareness of maritime incidents such as robberies, kidnappings, and hijackings at sea. These hostile actions are perpetrated by bad actors generally referred to as pirates. The areas in which these incidents take place are important for maritime routing. Shipping companies may

opt to route their vessels away from areas with high levels of such activities. Conversely, law enforcement agencies such as coast guards may pay special attention to such areas.

Identifying maritime incidents is then a key task of maritime risk assessment (MRA) and more generally, maritime domain awareness (MDA) [5]. Several organizations report maritime incidents in either structured or semi-structured formats. For example, the International Maritime Bureau (IMB) produces an annual *Piracy and Armed Robbery Against Ships*[1] report that gives structured descriptions of maritime incidents reported to the IMB throughout the year. The National Geospatial-Intelligence Agency (NGA) compiles a weekly *Worldwide Threats to Shipping*[2] (WWTTS) report that gives semi-structured descriptions of incidents reported to the NGA throughout the week. Both of these textual sources have been recently used to synthesize a risk-aware picture of the maritime environment together with other structured, sensor-emitted data sources such as AIS [17] [9].

This chapter attempts to exploit an additional type of data source: unstructured articles from maritime magazines such as *The Maritime Executive*[3], *gCaptain*[4], *World Maritime News*[5], *Marine Log*[6], *Marine Link*[7], non-maritime newspapers such as *The New York Times*[8], as well as unstructured reports from the *Regional Cooperation Agreement on Combating Piracy and Armed Robbery against Ships in Asia (ReCAAP)*[9]. Unlike IMB, NGA, and ReCAAP reports which are dedicated to describing maritime incidents, most magazine articles discuss the economics, politics, and news of the maritime world; only a small subset of articles describe maritime incidents. Additionally, the magazine articles are unstructured, which

[1] `https://www.icc-ccs.org/index.php/piracy-reporting-centre`
[2] `https://msi.nga.mil/NGAPortal/MSI.portal?_nfpb=true&_pageLabel=msi_portal_page_64`
[3] `https://maritime-executive.com/`
[4] `https://gcaptain.com`
[5] `https://worldmaritimenews.com`
[6] `www.marinelog.com`
[7] `www.marinelink.com`
[8] `www.nytimes.com`
[9] `http://www.recaap.org/`

makes it more difficult to automatically extract the details of the incident. Finally, the articles may lack specific details about the incident (e.g. the exact location of the incident) which are usually found in the IMB and NGA reports.

This work makes two key contributions to deal with the issues identified above. The first is a document classification scheme that determines if an article describes at least one maritime incident. The articles are converted to bags-of-words [34] (two versions of bag-of-words are tested: binary and frequency) and classified with 40 machine learning algorithms from Weka [61] version 3.8. The second contribution is a suite of techniques for extracting information from articles that describe at least one maritime incident. In particular, these information extraction methods determine the location of the incident (e.g. Gulf of Guinea), the type of incident (e.g. kidnapping, robbery), and the type of the victim's vessel (e.g. cargo ship, fishing vessel). The information extraction techniques use regular expressions, named entity recognition, and contextual cues to identify relevant information and to deal with conflicting information. Together, these techniques form a pipeline where the positively labelled articles from the document classification algorithm (i.e. articles describing maritime incidents) are fed into the information extraction algorithms.

The experimental results reveal good performance of all techniques: the best document classification algorithm (Logistic Regression) achieves an accuracy of 98.5%, the location extraction algorithm achieves 87.9% accuracy, the vessel type identification approach achieves an F-measure of 92.2%, and the incident type identification approach achieves an F-measure of 88.7%.

The rest of this chapter is structured as follows. Section 4.2 describes the methods employed in this chapter, including a description of the data sources. Section 4.3 presents the experimental results. Finally, Section 4.4 discusses the results and concludes the chapter.

4.2 Proposed Approach

4.2.1 Data Sources

Due to the specific nature of this work, there are no known publicly available repositories of appropriately labeled data. Therefore, part of this work has been to manually generate a suitable dataset by downloading, reading, and labeling a total of 602 articles from *The Maritime Executive*, *gCaptain*, *World Maritime News*, *Marine Log*, *Marine Link*, *The New York Times*, and *ReCAAP*.

To establish the ground truth for the document classification scheme, each article was assigned the class label of *incident* if it describes one or more maritime incidents and *no-incident* otherwise; 303 were labelled as *incident* and 299 were labelled as *no-incident*. Figure 4.1 shows the distribution of *incident* articles vs *no-incident* articles from each data source. In order to avoid the complications of an imbalanced dataset, the datasets was intentionally populated with roughly half of each class. In reality, the dataset is actually imbalanced (Section 4.2.2).

To establish the ground truth for the information extraction techniques, each article that describes a maritime incident was annotated with a list of acceptable answers to the questions: "where did the incident occur?", "what type of vessel was targeted" and "what type of incident occurred?". For example, if an article contains the sentence "*A cargo ship was hijacked near Port Harcourt off the coast of Nigeria*" then the victim ship type would be "cargo ship", the incident type would be "hijacking" and the location would be either "Port Harcourt" or "coast of Nigeria".

In addition to the maritime articles, the techniques also makes use of several

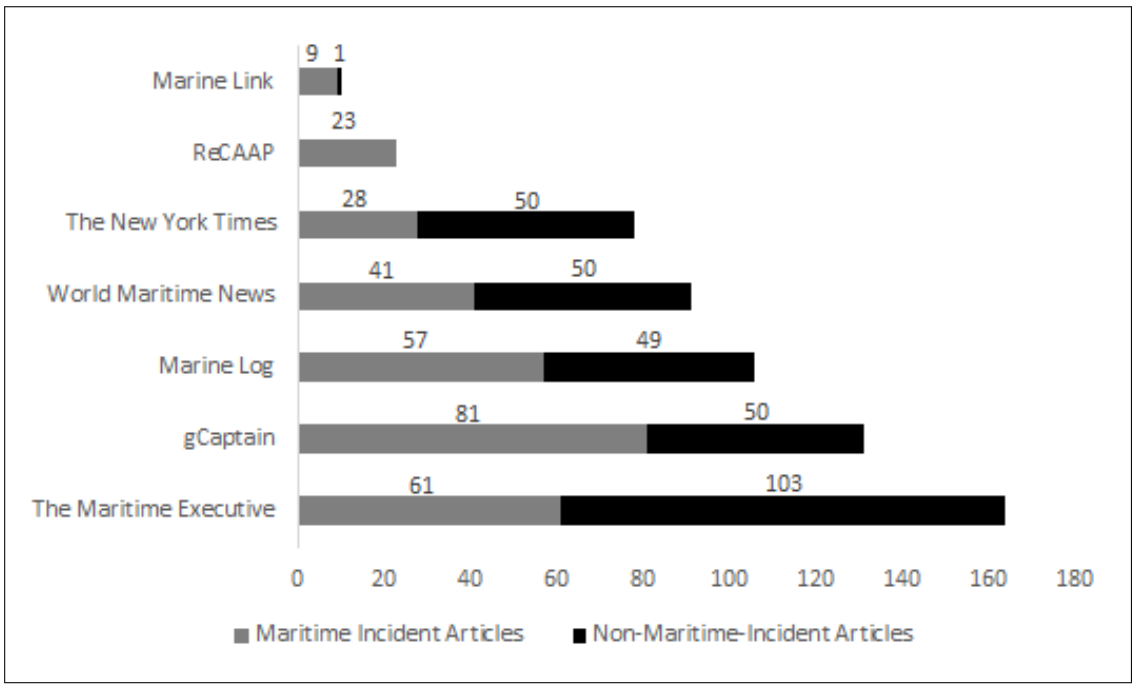

FIGURE 4.1: Number of articles from each data source

lists of geographical locations including countries[10], cities[11], seas[12], straits[13], and archipelagos[14].

Finally, the approach relies on a list of incident-related keywords that was manually created for this work. These words are: *hijack*{*ed,ing*}, *board*{*ed,ing*}, *attack*{*ed*}, *took, taken, stole, captur*{*ed*}, *fired on, seize*{*d*}, *lost, approach*{*ed*}, *abduct*{*ed*}, *pirate*{*s*}, *sank, shell*{*ed*}, *gunfire, kill*{*ed*}, and *kidnap*{*ped*}. This list of keywords is augmented with some of the words from the lexicon of [17]: *rob*{*bed,bing,bers*}, *chas*{*e,ed,ing*}, *clash*{*ed*}. These words serve as contextual cues which indicate that a sentence is likely describing the specifics of a maritime incident.

Our corpus of maritime incident articles has been made publicly available[15].

[10] `https://simple.wikipedia.org/wiki/List_of_countries`
[11] Compiled manually based on cities mentioned in the corpus
[12] `http://www.blue-growth.org/Oceans_Rivers_Seas/Index_Oceans_Seas_Bays_Gulfs_Of_The_World_%20A_To_Z_Lists.htm`
[13] `https://en.wikipedia.org/wiki/List_of_straits`
[14] `https://en.wikipedia.org/wiki/List_of_archipelagos`
[15] `https://github.com/alex-teske/maritime-incident-article-repository`

4.2.2 Article Classification

Most of the articles published in maritime magazines discuss the economics, politics, and news of the maritime world. Only a small subset of these articles actually describe maritime incidents. Therefore, document classification is used to automatically identify articles which describe one or more maritime incidents and discard articles that discuss unrelated topics. The "Document Classification" block of Figure 4.2 shows the block diagram of the text classification approach. The steps are now described:

1. **Preprocessing:** Punctuation, numbers, and stop words are removed from the articles. The remaining words are lemmatized using Natural Language ToolKit (NLTK)' [62] *WordNetLemmatizer*.

2. **Article to Vector:** Each article is converted to a bag-of-words vector. Two versions of bag-of-words are tested: binary and frequency (Section 2.4.1). The class label of each article is appended to the vector.

3. **Feature Selection:** Two feature selection approaches are tested. The first is Weka's [61] *CsfSubsetEval*, which searches for subsets of attributes that are highly correlated to the class while having low intercorrelation. The second approach is to tally all the words from each article to determine which words occur most frequently, and to retain the n top words from this list. Through trial-and-error, a value of $n = 300$ was determined to achieved a good balance between the size of the vector and accuracy. The number of features selected by each approach is given in Table 4.2.

4. **Classification:** 40 classifiers are trained on the dataset using the Weka tool. Table 4.1 lists all the classifiers that were tested. In all cases, the classifier's default parameters were used, and the performance was tested with 10-fold cross-validation. For the fourteen meta-classifiers in Table 4.1, BayesNet is

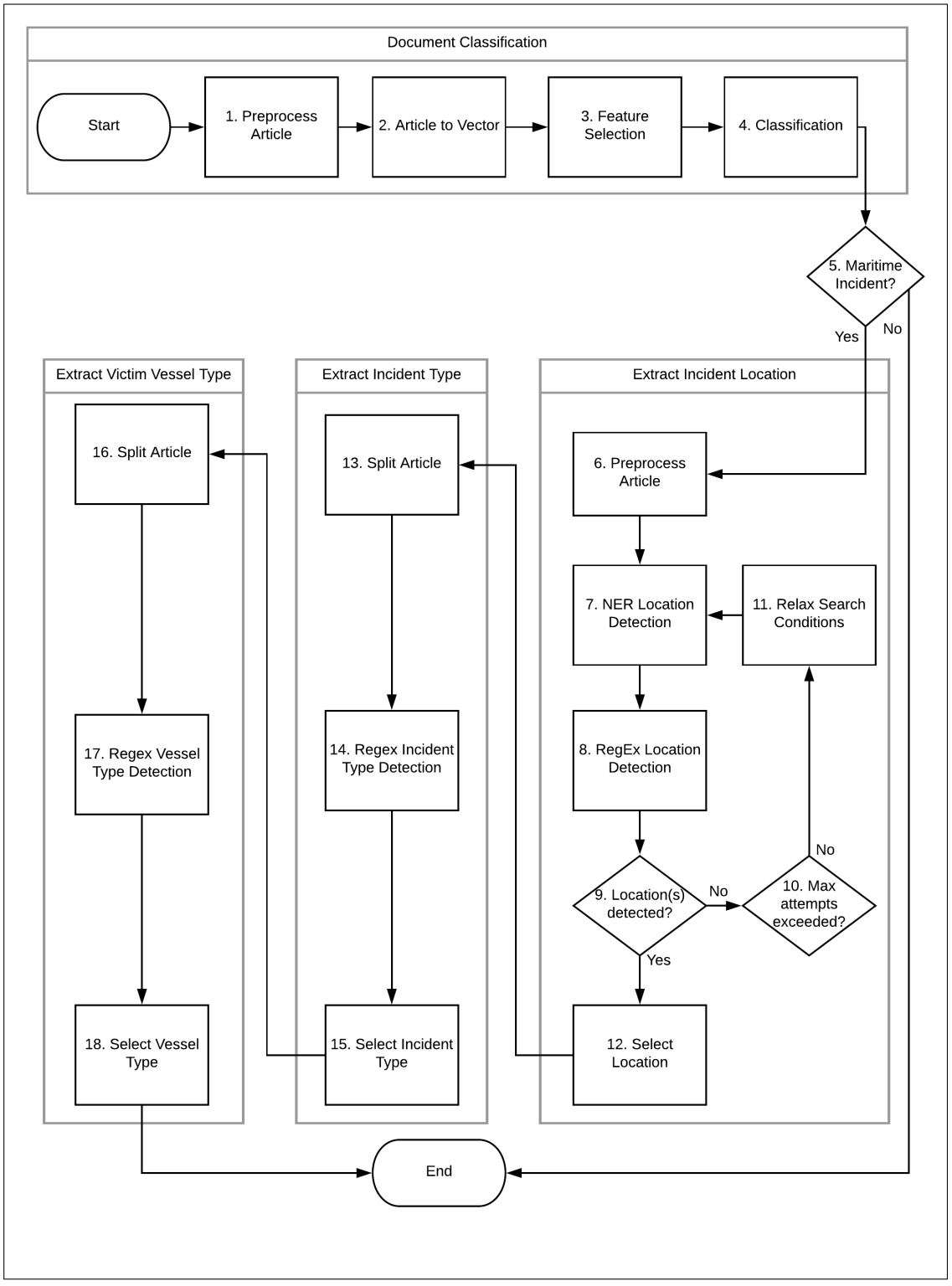

FIGURE 4.2: Natural language processing architecture

TABLE 4.1: Weka 3.8 Classifiers

Classifier Family	**Classifiers**
Bayes	BayesNet, NaiveBayes, NaiveBayesMultinomialText, NaiveBayesUpdatable
Function	Logistic, MultilayerPerceptron, SGD, SGDText, SimpleLogisic, SMO, VotedPerceptron
Lazy	IBk, KStar, LWL
Meta	AdaBoostM1, AttributeSelectedClassifier, Bagging, ClassificationViaRegression, CVParameterSelection, FilteredClassifier, IterativeClassifierOptimizer, LogitBoost, MultiClassClassifier, MultiClassClassifierUpdateable, RandomCommittee, RandomizableFilteredClassifier, RandomSubSpace, WeightedInstancesHandlerWrapper
Rules	DecisionTable, JRip, OneR, PART, ZeroR
Trees	DecisionStump, HoeffdingTree, J48, LMT, RandomForest, RandomTree, REPTree

employed as the base classifier if applicable. Otherwise, Weka's default value is used. The performance of these classifiers is compared in Section 4.3.1.

4.2.3 Maritime Incident Information Extraction

Given a set of articles, each describing at least one maritime incident, it would be helpful to employ a suite of information extraction techniques to automatically extract key information about the incidents from the articles. The following subsections describe the proposed approaches to determining the location, incident type, and victim vessel type.

4.2.3.1 Extracting the Incident Location

The location of a maritime incident is perhaps the most important detail to be extracted, yet it is also the most complex detail to accurately determine. In addition to the expected challenges of finding information in unstructured articles, extracting the incident location in an article poses some additional challenges. Firstly, articles tend to mention multiple locations which are merely peripherally related to the incident location (e.g. "the vessel was en route to *Yemen* but it was intercepted *10 nm south of Socotra Island*"). Secondly, locations can be

represented in several different formats (e.g. coordinates, relative locations, names of specific places).

The proposed solution is shown in the "Extract Incident Location" block of Figure 4.2. The steps of the algorithm are now described:

1. **Preprocessing** To preprocess the data, any periods are removed from abbreviations in the article (e.g. "U.S.A" becomes "USA"). Occurrences of "No." are replaced with "Number". These steps improve NLTK's [62] ability to split the article into sentences. Next, NLTK's *sent_tokenize, word_tokenize*, and *pos_tag* functions are used. These functions respectively (1) split the text into sentences, (2) tokenize the individual sentences, and (3) assign part-of-speech (POS) tags (e.g. noun, verb) to each token.

2. **Named Entity Recognition Location Detection** NLTK's *ne_chunk* is used to detect named entities. Since the named entities may be the names of people, organizations, or geopolitical entities, named entities not meeting at least one of the following criteria are discarded:

   (a) the named entity contains the word "island", "strait", "port", "terminal", or "anchorage".

   (b) the entity is the name of a country, sea, archipelago, strait, or city.

   To check condition (b), the named entities are compared to a list of countries, seas, etc. as mentioned in Section 4.2.1.

3. **Regular Expression Location Detection** Occasionally, the NER location detection step can fail to detect seemingly obvious locations. For example, "Gulf of Guinea" will likely be recognized as a named entity but "gulf of guinea" might not. Therefore the NER approach is supplemented with regular expressions that detect coordinates, islands, ports, coasts, rivers, etc. For example, the regular expression for detecting ports is `(?:Port|port)(?:`

`of)?` `[A-Z][a-z]*`. The complete list of regular expressions is given in Appendix A.

4. **Locations(s) Detected? / Max Attempts Exceeded?** The previous two steps may detect one or more candidate locations. If so, the algorithm proceeds to the "Select Location" step. Otherwise, the algorithm proceeds to the "Relax Search Conditions Step" if there is a condition to relax (see below), otherwise the algorithm terminates.

5. **Relax Search Conditions** Throughout the location identification process, the algorithm uses two conditions to focus the search on the parts of the article which are more likely to contain the location of the incident. These conditions are:

   Condition A - First three sentences Based on manual review, the location of the incident is often mentioned in the first three sentences of the article. When this condition is active, the algorithm only searches for locations in the first three sentences.

   Condition B - Filter incident sentences The location of the incident is likely to be mentioned in sentences that contain incident-related keywords, e.g. "the vessel was **hijacked** in the Gulf of Guinea". See Section 4.2.1 for a list of these keywords. When this condition is active, the algorithm only searches for locations in sentences that contain incident-related keywords.

   In the first round of location detection, conditions A and B are both active. If no location is detected, condition A is dropped and another round of location detection is conducted. If the second round does not identify any locations, condition B is also dropped and a final round of location detection occurs. If no locations are detected this time, the algorithm fails and returns "Unknown".

6. **Select Location** If more than one candidate location was identified in the previous steps, the Select Location block determines the final answer. The candidate locations are sorted according to their specificity:

   Tier 1 - Coordinate e.g. Lat 14.04, Long 51.63

   Tier 2 - Relative Location e.g. 115nm South of Salalah

   Tier 3 - Specific Location e.g. Lembeh Island

   Tier 4 - Port e.g. Port Klang

   Tier 5 - Coast e.g. Coast of Nigeria

   Tier 6 - City/Provinces/Town e.g. Palau Nipa

   Tier 7 - Nautical Location e.g. Gulf of Guinea

   Tier 8 - Country e.g. Malaysia

   All candidate locations that are not in the most specific tier for the article are discarded. For example, if an article mentions one port and two countries, the countries are discarded. If more than one candidate location still remains, the one that occurred earliest in the text is returned.

4.2.3.2 Extracting the Incident Type

Extracting the incident type of a maritime incident article is a less complex task than determining the incident's location. This is because there is a small set of potential incident types that can be captured by fairly simple regular expression. The goal is to assign each article an incident type from the following list, sorted by severity:

Sank The vessel was grounded, capsized, sank, etc.

Kidnapped The vessel's crew were kidnapped, ransomed, or abducted.

Hijacked Intruders took control of a vessel, without kidnapping being explicitly mentioned (e.g. crew were abandoned on lifeboats).

Boarded Intruders boarded a vessel without gaining control of the helm.

Robbed Cargo, fuel, equipment or possessions were stolen from a vessel.

Fired Upon The vessel was attacked with small arms, rocket propelled grenades, etc.

Attempted Hijacking/Attempted Boarding Attempted hostile action was prevented by evading, returning fire, third party intervention, etc.

Suspicious Approach The vessel was followed or approached but no overt hostile action occurred.

Unknown If the article does not give enough details to assign one of the above categories. This is quite common, as many articles will describe the incident as a "pirate attack" without giving more specific information.

The incident type extraction scheme is shown in the "Exract Incident Type" block of Figure 4.2. Each step is now described:

13. **Split Article** The article is split into three segments: the first segment contains only the article's title, the second segment contains the article's title as well as the first two sentences, and the last segment contains the title and the full text. Each segment is independently processed in the subsequent steps and the results are combined to a final answer in the "Select Incident Type" step.

14. **Regex Incident Type Detection** Regular expressions are used to extract words that indicate the type of incident. For example, the regular expression that extracts kidnapping-related keywords is ``kidnap(ped|ping|\W)|abduct|

`hostage''`. Additionally, an "evade" regular expressions detects situations where the attempted incident is unsuccessful (e.g. "navy **foils attempted** hijacking"). A full listing of the regular expressions is given in Appendix B.

15. Select Incident Type First, a candidate answer is selected from each of the article segments from Step 13. In general, the most severe incident type that is mentioned in each segment in selected as the candidate. However, if the "evade" regular expression detects that the incident was avoided, then the incident type is updated accordingly (e.g. "hijacking" is replaced with "attempted hijacking"). This generates up to three candidate incident types, one for each article segment.

Next, a final answer is selected among the candidates. In general, if an incident type was detected in the second segment it is selected as the final answer, otherwise the incident type extracted from the third segment is selected as the final answer. However, there are some exceptions wherein the first segment (i.e. the title) can improve the accuracy. If the final answer is "unknown" (i.e. the incident was referred to simply as a "pirate attack"), the title may contain enough information to conclude that the incident's category was, in fact, "fired upon". Therefore, if the incident type was "unknown" but the title contains a "fired upon" keyword then then answer is changed to "fired upon". Additionally, if the title contains a "hijacking" keyword and no "evade" term is found in the text, then the final answer is changed to "hijacking".

4.2.3.3 Extracting the Victim's Vessel Type

The victim vessel type is extracted in a similar fashion as the incident type. The goal is to assign each article one of the following vessel types: cargo ship, tanker, fishing vessel, supply vessel, tug boat, yacht, cruise ship, military/security ship, dhow (i.e. small sailing vessel), commercial vessel.

The victim vessel type extraction scheme is shown in the "Extract Victim Vessel Type" block of Figure 4.2. Each step is now described:

16. **Split Article** The article is split into two segments: the first segment contains only the article's title, and the second segment contains the article's content. Each segment is independently processed in the subsequent steps and the results are combined to a final answer in the "Select Victim Vessel Type" step.

17. **Regex Vessel Type Detection** Regular expressions are used to extract words that indicate a vessel type. For example, the regular expression that detects shipping vessels is ``` ``fishing (?:boat|vessel)'' ```. A full listing of the regular expressions is given in Appendix C.

18. **Select Victim Vessel Type** First, a candidate answer is selected from each of the article segments from Step 16. In general, the most frequently mentioned vessel type in each segment is selected as the candidate. However, since military ships and dhows are rarely targeted by pirates, these answers are not selected as candidates if other ship types were mentioned. Commercial vessels are similarly discarded since the term is unspecific.

    Next, a final answer is selected among the candidates. In general, if a vessel type was detected in the title, it is selected as the final answer. Otherwise, the candidate from the full text is selected. However, if the ship type from the full text is a subtype of the ship type of the title, then the more specific ship type is selected (e.g. fishing boat vs. commercial vessel).

TABLE 4.2: Classification accuracy by feature selection method, bag-of-words type, and article content

Dataset Version			***Accuracy across 40 classifiers***	
Part of Article	**Bag of Words Type**	**Feature Selection Method (# Features Selected)**	**Average 95% conf.**	**Max**
Title + Content	Binary	300 Most Frequent (300)	91.25% ± 3.84	97.5%
Title + Content	Binary	CfsSubsetEval (90)	91.91% ± 3.69	97.34%
Title + Content	Frequency	300 Most Frequent (300)	88.96% ± 4.03	96.84%
Title + Content	Frequency	CfsSubsetEval (45)	90.96% ± 3.63	97.01%
Title Only	Binary	300 Most Frequent (259)	90.84% ± 3.86	97.67%
Title Only	Binary	CfsSubsetEval (36)	92.12% ± 3.9	98.5%
Title Only	Frequency	300 Most Frequent (259)	90.65% ± 3.81	97.17%
Title Only	Frequency	CsfSubsetEval (34)	91.84% ± 3.91	98.17%

TABLE 4.3: Performance of maritime article classifiers

Classifier Family	**Best Classifier**	**Accuracy**
Function	Logistic Regression	98.5%
Meta	AdaBoostM1 (BayesNet)	98.33%
Bayesian	BayesNet & NaiveBayes	97.67%
Tree	LMT & RandomForest & RandomTree	97.67%
Lazy Learner	Ibk	97.34%
Rule-Based	Jrip	97.34%

TABLE 4.4: Performance metrics for Logistic Regression classifier

Class	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**
Not Maritime Incident	99.0%	2.0%	98.0%	99.0%	98.5%
Maritime Incident	98.0%	1.0%	99.0%	98.0%	98.5%
Weighted Average	98.5%	1.5%	98.5%	98.5%	98.5%

TABLE 4.5: Confusion matrix of the Logistic Regression classifier

	Predicted	
	No-Incident	**Incident**
No-Incident	300	3
Incident	6	293

4.3 Experimental Results

4.3.1 Document Classification Results

The procedure described in Section 4.2.2 is tested with every applicable classifier in Weka, 40 in total. These included tree-based (e.g. J48, RandomForest), rule-based (e.g. Decision Table, JRip), meta-classifiers (e.g. AdaBoost, Stacking), Lazy Learners (e.g. KStar, LWL), function-based (e.g. MultiLayerPerceptron, Logistic Regression) and Bayesian classifiers (e.g. BayesNet, Naive Bayes). Each classifier was tested using its default parameters; for meta-classifiers, BayesNet was selected as the underlying classifier when applicable. Testing was performed with the well-known k-fold cross-validation method with k=10.

In order to maximize the classification accuracy, eight variations of the dataset have been tested. The variations include two versions of the bag-of-words model (binary vs frequency), two feature selection methods (300 most frequent words vs *CfsSubsetEval*), and two versions of the article: in the first version, the title of each article is retained but its content is discarded; in the second version, the article's title and content are both considered. The purpose of this is to determine if the title alone contains enough information to classify the document. The results of this experiment are summarized in Table 4.2. The results favor the title-only version of the article using the binary bag-of-words approach and the *CfsSubsetEval* feature selection method. Therefore, these parameters are employed for the remainder of the experiments.

With this experimental configuration, the accuracy of the classifiers range from 50.33% to 98.5%. Table 4.3 shows the accuracy of the best classifier from each family. The best classifier was the Logistic Regression [63] classifier. It correctly labelled 593/602 articles. Table 4.4 shows the performance of the Logistic Regression classifier, and Table 4.5 reports the related confusion matrix.

4.3.2 Information Extraction Results

The methods described in Section 4.2.3 were applied to 299 articles which describe one or more maritime incidents. As mentioned in Section 4.2.1, each article was labeled with one or more acceptable answers to the questions "where did the incident occur?", "what type of vessel was targeted" and "what type of incident occurred?". The algorithm succeeds at extracting a piece of information when it returns any of the answers from a particular article's list, and fails otherwise.

The experimental results reveal good performance for all information extraction techniques. The location extraction algorithm achieves 87.9% accuracy, the victim vessel type extraction algorithm achieves an F-measure of 92.2%, and the incident type extraction algorithm achieves an F-measure of 88.7%. The success of these algorithms can be attributed to two things. Firstly, the approach for identifying potentially relevant information was very effective: the combination of NER and regular expressions seemed to rarely miss key information from the article. Secondly, the somewhat predictable format of an article (i.e. most important information early in the article, use of certain words, etc.) allowed the algorithm to often choose the correct information.

4.4 Chapter Summary

In this chapter, several NLP techniques for extracting maritime incidents from unstructured articles and incident reports have been presented.

The first technique classifies articles to determine if they describe one or more maritime incidents. The articles were converted to binary bags of words and Weka's *CfsSubsetEval* was used to select the most pertinent features. Machine learning algorithms were applied to the resulting vectors. The experimental results reveal that this approach is extremely successful, as the best classifier achieved

98.5% accuracy. The success of this approach may be related to the fact that articles which describe maritime incidents tend to use very specific words in the title, e.g. pirate, hijack, board, kidnap, etc, whereas these words are not used in no-incident articles. For example: in our corpus, 100% of the articles whose title used the word "pirate" described a maritime incident while 75.44% of the articles whose title did not include the word "pirate" did not describe a maritime incident. Thus, there is a clear distinction between articles which describe maritime incidents and those that do not, providing a basis for the machine learning algorithms to achieve very good results.

Additionally, a suite of information extraction algorithms were proposed to determine the incident location, incident type, and victim vessel type. The approach uses a combination of named entity recognition and regular expressions to identify candidate answers. The candidates were filtered based on indicators such as specificity (e.g. Port Klang is more specific than Malaysia), position within the article (earlier is better), and frequency (i.e. a frequently repeated piece of information is likely important). The approach proved to be fairly reliable on the test data, achieving 87.9% accuracy for location extraction, an F-measure of 92.2% for victim vessel type extraction, and an F-measure of 88.7% for incident type extraction. The success of these algorithms can be attributed to the use of several NLP techniques to extract potentially relevant data and the effective use of context to select the most relevant information.

Together, these techniques form a pipeline, where the first technique identifies articles which describe maritime incidents and the second technique determines the details of the incident. These can then be fed into the RMF's *regional hostility* risk factor.

Chapter 5

Parallel Maritime Risk Assessment

This chapter introduces two improvements that have been made to the Risk Management Framework (RMF): namely, a parallel implementation of the RMF and a newly developed *weather risk* factor.

5.1 Parallel Risk Management Framework

The Parallel Risk Management Framework (PRMF) is a parallel version of the RMF's Risk Feature Extraction and Risk Assessment modules. The goal of parallelizing these modules is to enable real-time risk assessment in systems with large number of system units, complex risk features, etc. The PRMF scales horizontally: that is, by adding additional processors to the system. In this way, the system can scale as the workload increases.

The method by which the Risk Feature Extraction and Risk Assessment modules are parallelized is now described. Let us assume that the PRMF has p processors and the system is assessing n risk levels. One arbitrary processor is deemed the

master, and the remaining processors are denoted workers. The master assigns roughly $\frac{n}{p-1}$ risk assessments to each worker and transmits the relevant input data to each worker. Each worker uses its own instance of the RMF's Risk Feature Extraction and Risk Assessment modules to evaluate their subset of the data. Finally, the workers report their results back to the master. Theoretically, this scheme allows the PRMF's risk assessment performance to scale linearly with p, which is the ideal result.

The above approach assumes that each unit can be evaluated independently, i.e. the risk level of one unit does not depend on the state of another unit. However, there can be exceptions to this. For example, in the maritime domain the *collision factor* of one unit (i.e. vessel) depends on the position of all other units – in particular, the nearest other vessel. This induces a requirement for a preprocessing step in which information that depends on more than one unit is computed. This may require domain-specific parallel algorithms. Section 5.1.1 explains the parallel preprocessing step for the maritime domain's *collision factor*.

5.1.1 Example: Parallelizing Collision Factor Calculations

Calculating the *collision factor* for maritime[1] vessels requires knowledge of each vessel's distance to the nearest other ship. The naive approach is to build an $n * n$ distrance matrix that contains the distance between each pair of vessels. Parallel computing can simplify this by tasking each of the $p-1$ worker nodes to calculate a partial $\frac{n}{p-1} * n$ distance matrix containing the distance between each assigned vessel and every other vessel. The complexity of this process is $\frac{n}{p-1}n \rightarrow O(n^2)$, which is undesirable. Therefore, an alternative approach using KD-Trees [64] is suggested.

[1] In this domain, n technically refers to the number of AIS messages rather than the number of vessels, however for this example it can be assumed that n vessels each emit a single AIS message.

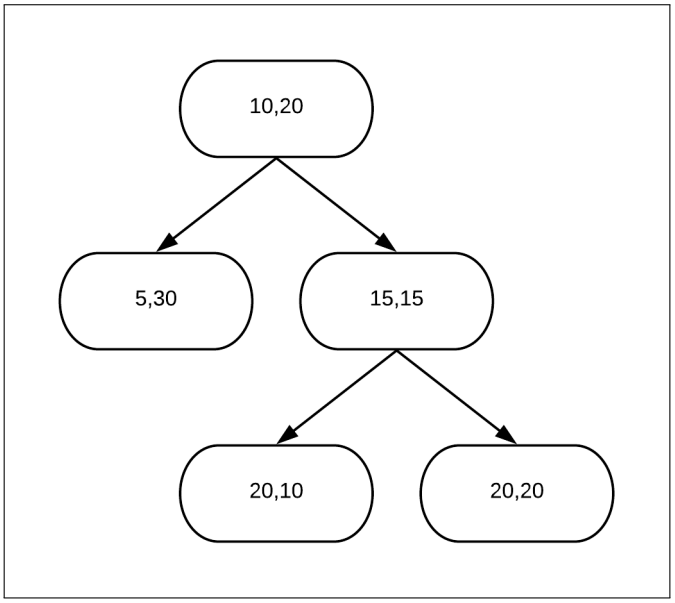

FIGURE 5.1: Example of a KD-Tree

KD-Trees are binary search trees for storing K-dimensional points. They can be constructed in $nlog(n)$ time and nearest neighbor searches can be performed in $log(n)$ time. Figure 5.1 illustrates an example of a 2D KD-Tree. The *collision factor* preprocessing step proceeds as follows: each worker (1) constructs a KD-Tree containing the position of all vessels in $nlog(n)$ time and (2) queries its tree once for each assigned vessel in $\frac{n}{p-1}log(n)$ time for an overall complexity of $nlog(n) + \frac{n}{p-1}log(n) \rightarrow O(nlog(n))$. This is a significant improvement over the distance matrix approach.

The proposed design of the PRMF for Maritime Risk Assessment is shown in Figure 5.2.

5.1.2 Data Sources

The data for our experiments originates from the following sources:

AIS Data This experiment considers one million fictional vessels with random positions. The dataset is created by simultaneously generating a single AIS

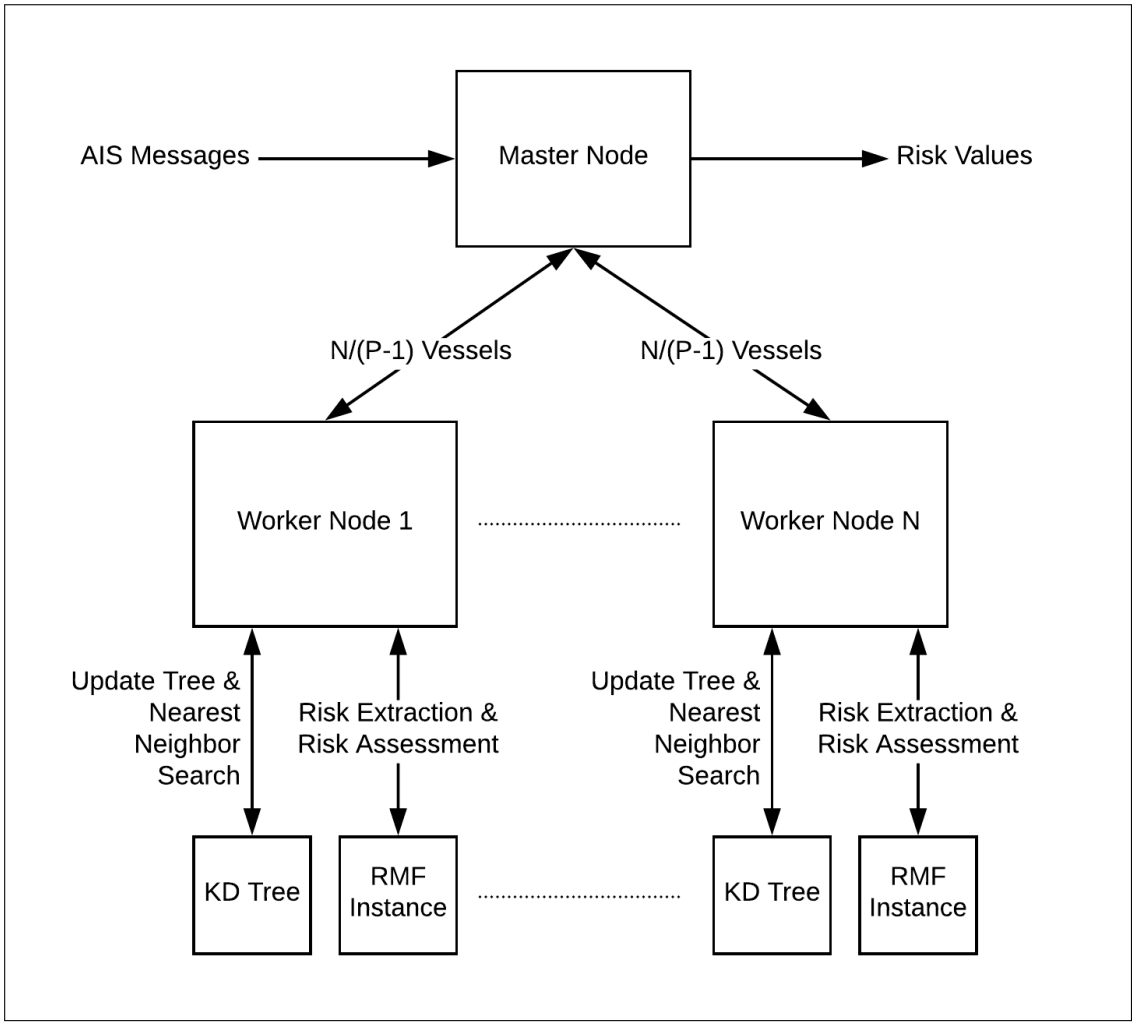

FIGURE 5.2: Proposed design of the Parallel Risk Management Framework for Maritime Risk Assessment

message from each ship. These messages specify the ship's location and ship type.

Weather Data As specified in Section 5.2, this experiment uses weather data from WaveWatch III.

Maritime Incident Reports The International Maritime Bureau (IMB) is a department of the International Chamber of Commerce (ICC) that specializes in fighting maritime crimes. The IMB's *Piracy and Armed Robbery Against Ships Report*[2] lists maritime incidents that occur each year. Figure 5.3 is an

[2] `https://www.icc-ccs.org/`

08.03.2017 0800 UTC Steaming Fired upon	Sofia Bulk Carrier Liberia 32983 9472086	03:20.0N – 004:28.9E, Around 106nm SW of Bayelsa Coast, Nigeria	Seven persons armed with guns in a skiff approached and fired upon the ship underway. Ship increased speed and commenced evasive manoeuvre. All non-essential crew retreated to the Citadel. After 40 minutes, the skiffs aborted the attack and moved away. All crews reported safe.

FIGURE 5.3: Sample incident from IMB's Piracy and Armed Robbery Against Ships report

example of an incident report from the IMB. These reports drive the *regional hostility* risk feature of this experiment.

5.1.3 Benchmark

Two versions of the PRMF have been implemented. The first is implemented with Apache Spark in the Java programming language (Spark-PRMF) and the other is implemented with MPI in the C++ programming language (MPI-PRMF). The benchmarks were run in a parallel environment on SHARCNET's Graham cluster[3]; each node was equipped with Intel E5-2683 processors clocked at 2.1GHz as well as 10GB of RAM.

Figure 5.4 shows the parallel speedup curve of the MPI-PRMF, and Figure 5.5 shows the parallel speedup curve of the Spark-PRFM. The dashed line in each graph indicates the ideal linear speedup curve, i.e. the single core processing time divided by the number of processors. Both implementations achieved a linear speedup curve, although the MPI implementation follows the ideal curve more closely. Figure 5.6 compares the performance of MPI-PRMF and Spark-PRFM. At all levels of parallelism, the MPI-PRMF outperforms the Spark-PRFM. Furthermore, the performance gap increases along with the number of processors, e.g.

[3]The Shared Hierarchical Academic Research Computing Network (SHARCNET) is a Canadian high performance computing consortium. Graham is SHARCNET's most powerful cluster; it has 33448 CPU cores spread over 1043 nodes, a total of 149 TB RAM, and 320 GPUs.

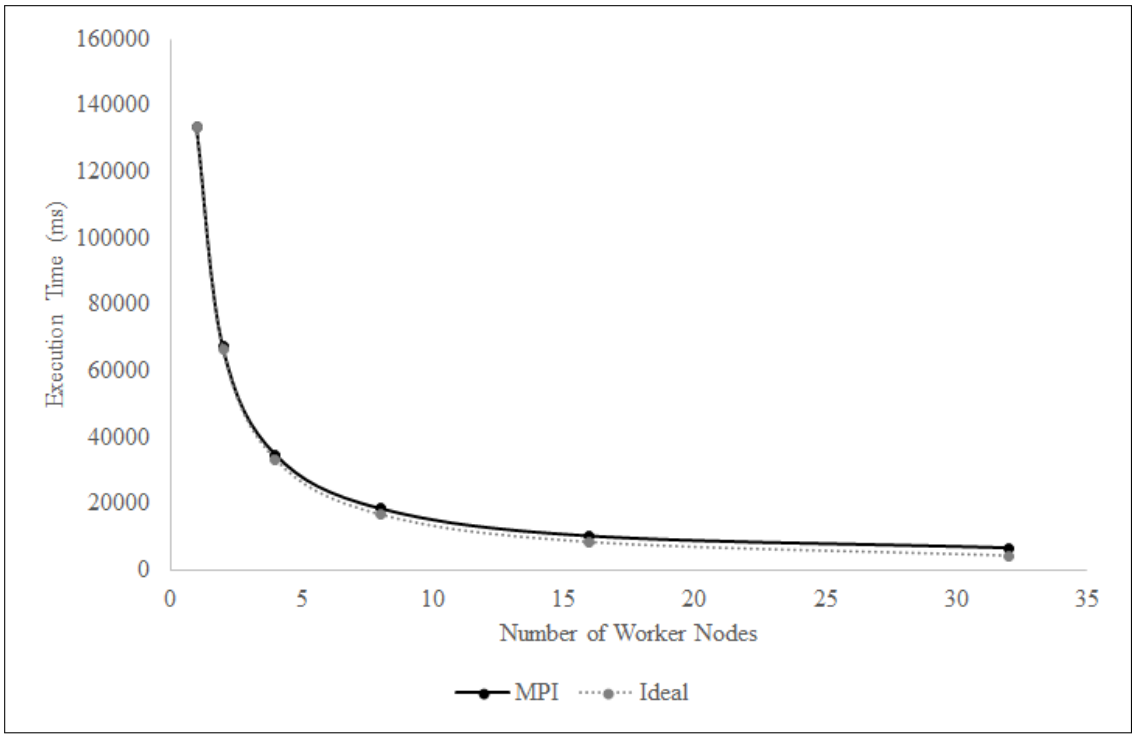

FIGURE 5.4: MPI-PRMF benchmark

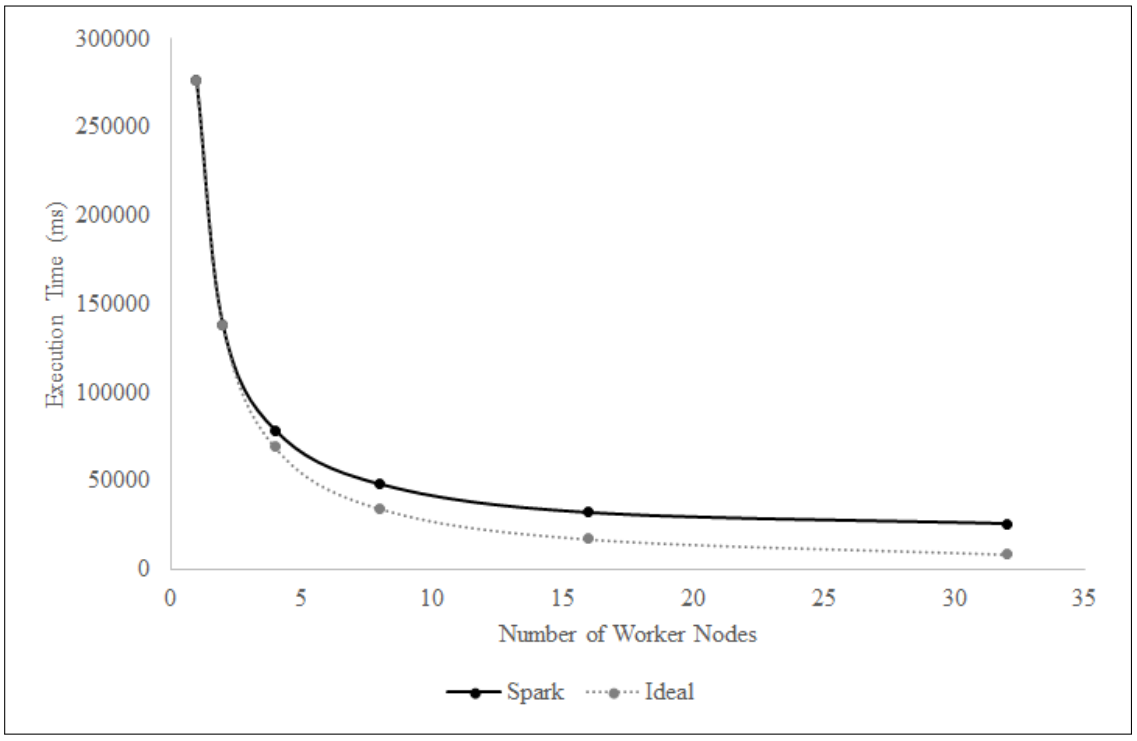

FIGURE 5.5: Spark-PRFM benchmark

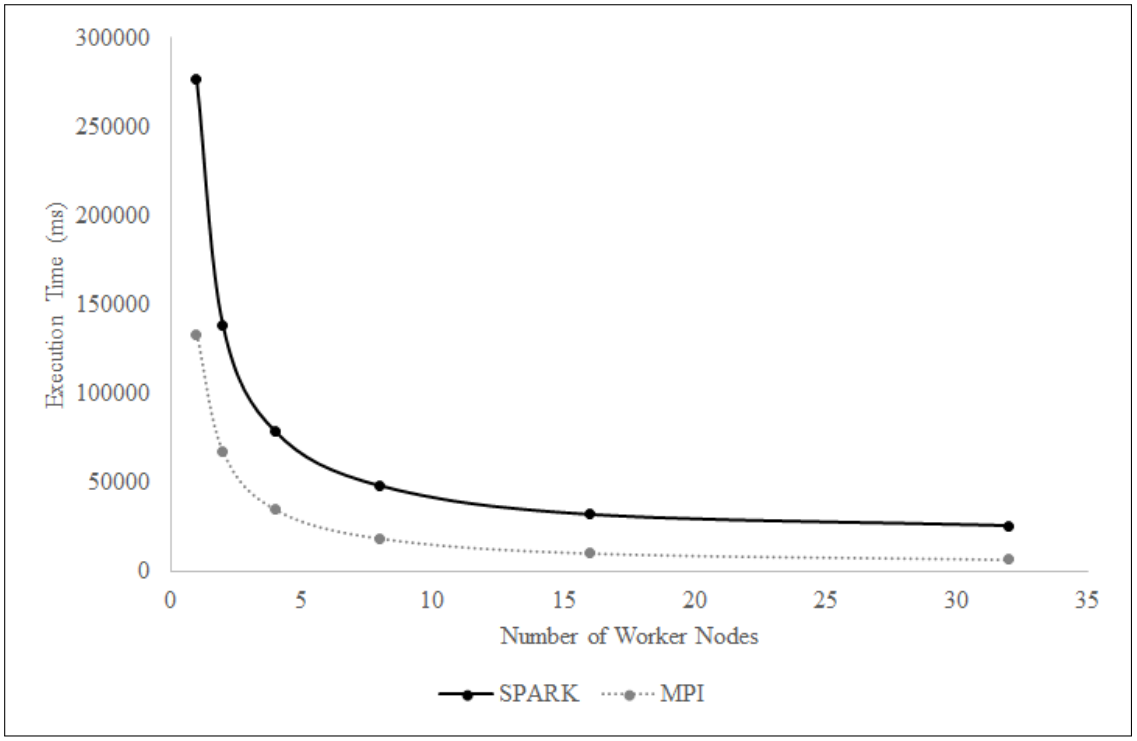

FIGURE 5.6: Spark-PRFM vs. MPI-PRMF benchmark

with one worker node the MPI-PRMF is roughly twice as fast as the Spark-PRMF but with 32 workers the MPI-PRMF is nearly four times faster than the Spark-PRMF. This is because as the number of processors increases, the deviation from the ideal linear curve increases – and this is more so true for the Spark-PRMF.

5.1.4 Spark vs. MPI

Figure 5.6 reveals a clear performance difference between the MPI and Spark implementations of the PRMF. Across all numbers of worker nodes, the MPI-PRMF is consistently faster than the Spark counterpart. The performance gap ranges from 2x-4x as the number of processors increases. Overall, this performance difference is unsurprising since the low-level MPI makes the programmer responsible for the communication layer. Each send and receive must be manually crafted, which can lead to thoroughly optimized parallel computing.

What Spark lacks in raw performance, it trades in ease of use and fault tolerance. As a high-level framework, it is much easier both to develop algorithms from scratch and to make changes to existing algorithms. Another significant advantage is Spark's fault tolerance. When a Spark worker node fails, some progress may be lost but the master node will automatically reassign the work to another node. However, if the driver node fails then the Spark context is lost and all progress is lost.

5.2 Weather Risk Factor

Weather conditions can pose a serious risk to a ship's safety. The relevant aspects of weather that could impact safety include visibility, ice conditions, currents, etc. However, the single most important weather factor that impacts risk is *wave height* originating from wind and swell [65]. Therefore, wave height is used as the sole indicator for the *weather risk* feature.

Previous versions of the RMF have proposed a *sea state* risk feature, however they lacked a data source to drive the risk feature so random *sea state* values were used. In this work, a new *weather risk* feature is proposed. The National Oceanic and Atmospheric Administrations's[4] (NOAA) WaveWatch III[5] has been identified as a suitable data source for weather data. Wavewatch III provides various weather forecasts in the GRIB (Gridded Binary) file format [66]. The GRIB format divides the globe into a grid and reports various weather conditions for each grid cell. These conditions include wave height, which is used throughout this work. Figure 3.4 is an example of a visualized wave height GRIB file.

The wave height at a vessel's position is mapped to a risk value via a trapezoidal membership function with a=1.25 m, b=14 m, c=d=INF. This configuration is

[4]NOAA is an American agency that is concerned with climate monitoring
[5]`ftp://polar.ncep.noaa.gov/pub/history/waves`

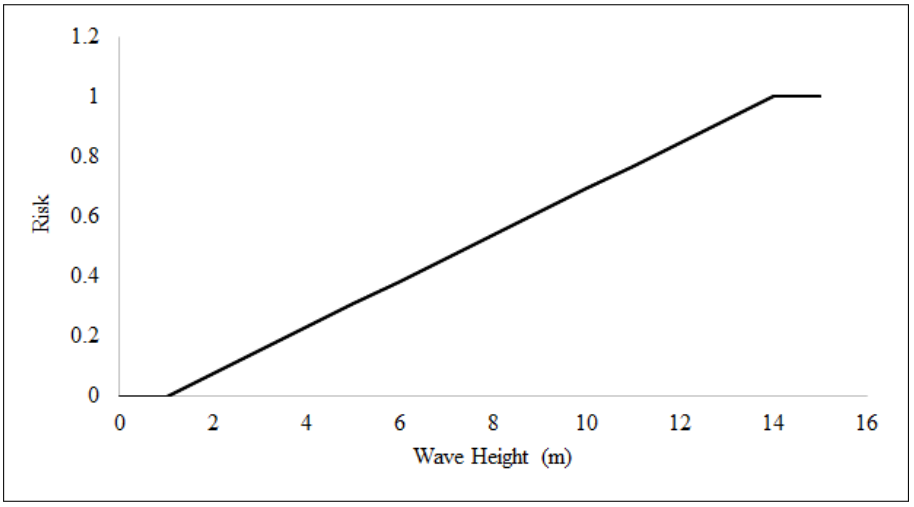

FIGURE 5.7: Trapezoidal membership function to map wave height to risk

inspired by the World Meteorological Organization (WMO) sea state code[6], according to which waves are "moderate" at 1.25 m and "rough" by 2.5 m. Figure 5.7 illustrates the membership function that maps the wave height to a risk value.

5.3 Chapter Summary

This chapter has introduced two improvements that have been made to the RMF.

The first is the design and implementation of the PRMF: a parallel implementation of the RMF's Risk Feature Extraction and Risk Assessment modules. The PRMF was implemented in MPI (C++) and Apache Spark (Java). The benchmarks revealed that both implementations' performance increases linearly with the number of worker nodes assigned, but MPI had superior performance.

The second is the development of a weather-based risk feature. A suitable data source for wave height data was identified. The RMF then determines the wave

[6] `https://www.nodc.noaa.gov/woce/woce_v3/wocedata_1/woce-uot/document/wmocode.htm`

height at a vessel's position and maps this to a risk value with a trapezoidal membership function. This risk feature is used in this chapter's experiments (Section 5.1.2), as well as the experiments in Chapter 3 (Section 3.2.1).

Chapter 6

Maritime Risk Visualization

This chapter presents a Cesium[1]-based maritime risk visualization tool. This provides a visual representation of the work described in the previous chapters.

6.1 Proposed Visualization System

This chapter proposes a Cesium-based maritime risk visualization system that fulfills the role of the "User Interface" module of the Risk Management Framework (RMF) (Figure 2.1). The user-interface predominantly features an interactive map of the world. The positions of vessels are indicated with arrows, coloured according to their current risk level (i.e. green = low risk, yellow = medium risk, red = high risk). Figure 6.2 shows the system as it surveils a single medium-risk vessel, indicated by the yellow arrow. Additional functionality is provided in the following side panels:

Visualization Layers Allows the user to toggle information to be displayed on the map. The layers include: weather information, the locations of maritime

[1] `https://cesiumjs.org/`

incidents, the locations of ports, and known traffic lanes. For example, in Figure 6.1 the system is displaying wave height data (indicated as a heat map) as well as the locations of maritime incidents (indicated with placemark icons).

Vessel Information Provides in-depth information about a particular vessel. This includes static information such as the type of vessel as well as dynamic information such as the speed and current destination. This panel also provides in-depth information about the vessel's current risk status as well as its current schedule. This panel can be seen in Figure 6.2.

Fleet Monitoring Provides an overview of several vessels of interest. This information includes the vessels' names, destinations, statuses, and a summary of the vessels' risk status. This panel is illustrated in Figure 6.3.

Settings Allows an operator to adjust three settings: (1) the risk thresholds, i.e. the boundary between low/medium and medium/high risk, (2) the time for which a vessel must remain in the high risk status before being deemed a vessel-in-distress, and (3) the option to rewind and play back the current scenario. This panel can be seen in Figure 6.4.

Response Selection Although response generation is not a focus of this thesis, the visualization system does feature a response selection mechanism. For example, in Figure 6.5, a vessel's planned route traverses a dangerous weather event. The system responds by suggesting several alternative routes and giving the user the option of changing the vessel's course. The vessel's original route is indicated by the dashed white line, and one of the suggested routes is indicated with a dashed purple line.

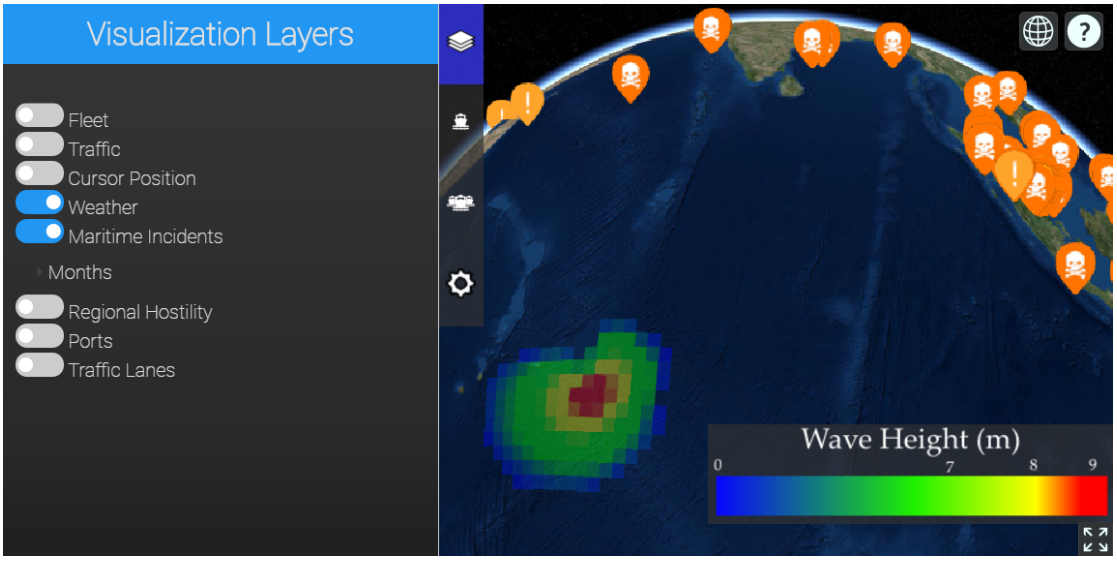

FIGURE 6.1: Visualization layers panel

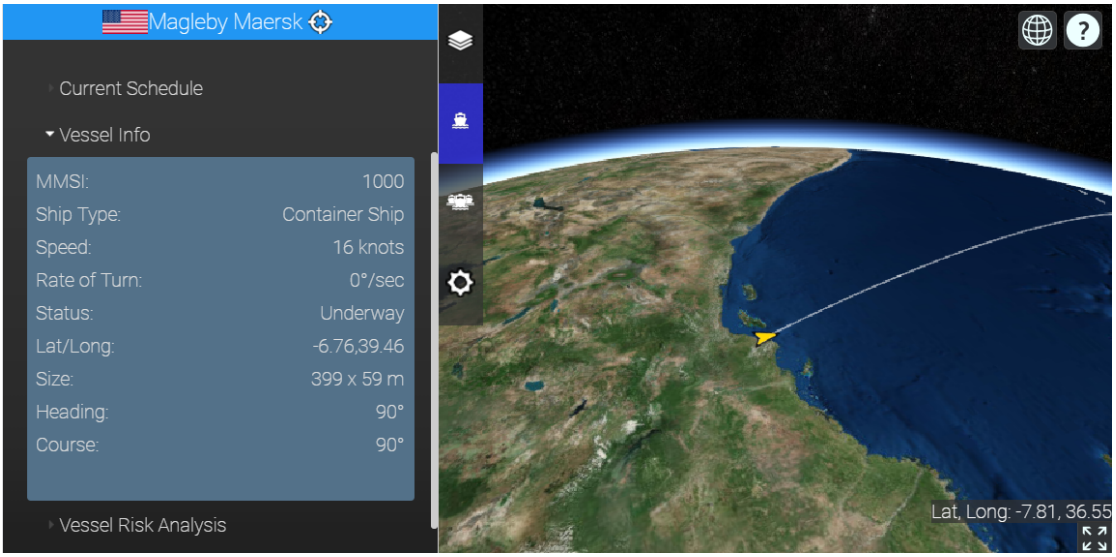

FIGURE 6.2: Vessel information panel

6.2 Chapter Summary

This chapter introduced a maritime risk visualization system built with Cesium. This system displays the positions and risk levels of vessels on a large interactive map, and provides additional details about vessels in several side panels. Additional information (e.g. weather, port locations) toggled on or off according to the user's preferences.

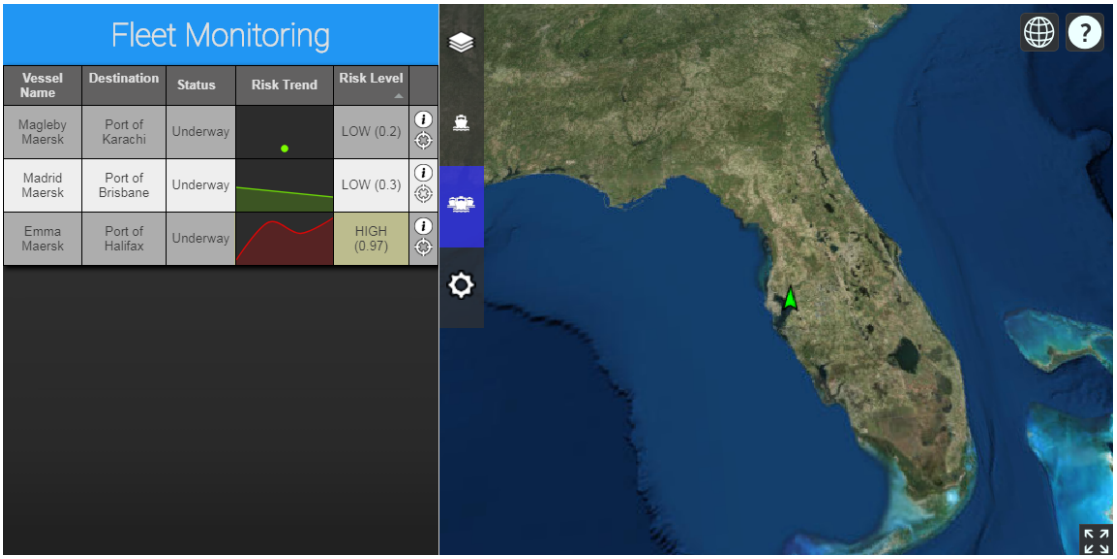

FIGURE 6.3: Fleet monitoring panel

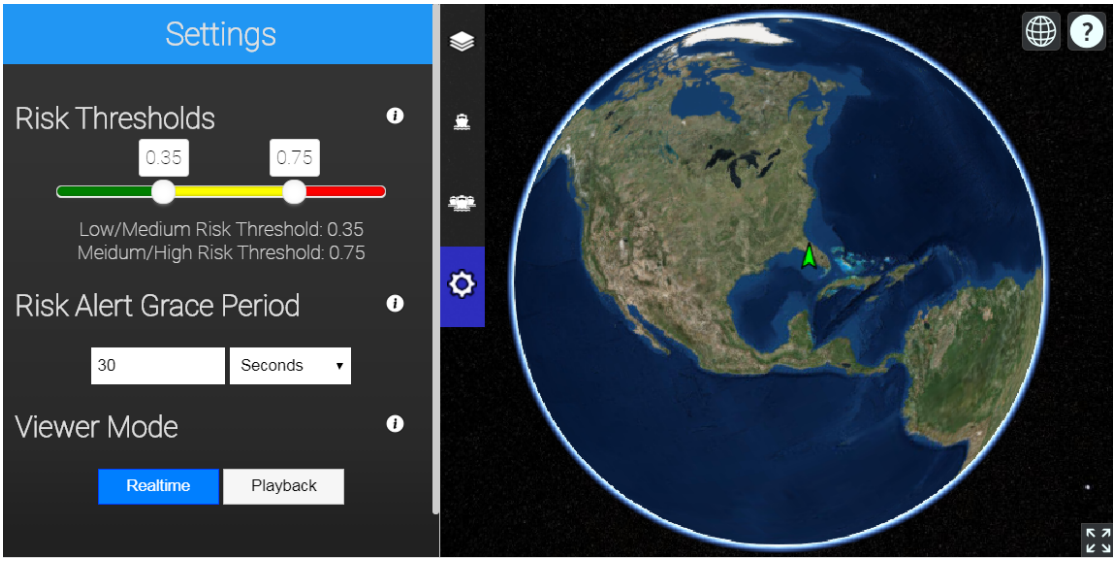

FIGURE 6.4: Visualization settings panel

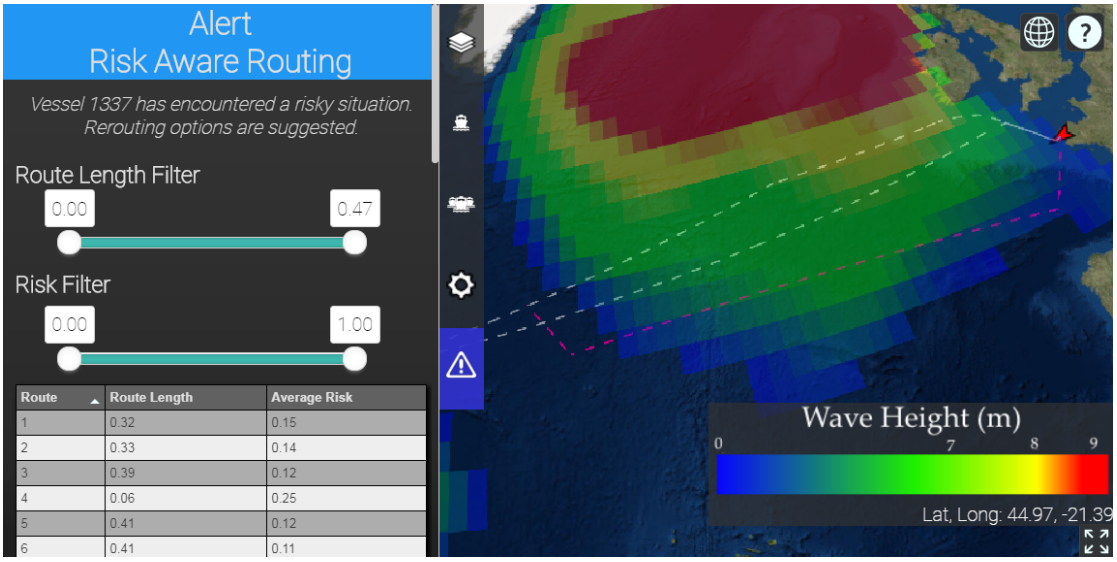

FIGURE 6.5: Response selection panel

Chapter 7

Conclusion

This chapter concludes the thesis by summarizing the contributions unveiled in the previous chapters and by outlining several potential future research directions.

7.1 Summary of Contributions

The main contribution of this thesis is automating two key aspects of an existing Risk Management Framework:

- Genetic fuzzy system (GFS) were integrated with the RMF's Risk Assessment module. This reduces the system's reliance on a domain expert by learning the rule base of the fuzzy inference system (FIS) directly from data. Therefore, GFSs contributes to the automation of the system.
- A suite of Natural language processing (NLP) techniques have been developed to extract details about maritime incidents (i.e. piracy) from unstructured articles published by newspapers or magazines. This allows the RMF to passively learn about maritime incidents as they occur, and adjust the *regional hostility* risk metric accordingly.

In addition to these contributions, the following improvements to the RMF have been made:

- The design and implementation of a parallel version of the RMF's Risk Feature Extraction and Risk Assessment modules. This parallel RMF (PRMF) has been implemented in MPI and Apache Spark, and has been shown to scale linearly with the number of processors allocated to it. This enables the RMF to scale horizontally in order to handle increasingly large volumes of incoming data.
- The development of a new *weather risk* factor for the RMF. This feature uses real weather forecasts to determine the risk imposed by a vessel's local weather conditions.
- A maritime visualization system built with Cesium. This provides a visual representation of all of the above contributions, and serves as a prototype of a decision support system with the RMF at its core.

7.2 Future Research Directions

Several improvements can yet be made to the RMF to better manage maritime risk:

- The RMF's Risk Feature Extraction module still relies on domain experts to specify the mapping between raw data and risk features, however it may be possible to automatically generate these risk features directly from data.
- The *collision factor* risk feature could be updated to take speed and heading into account in addition to proximity. It could also take into account the locations where collisions have historically been more likely.

- The PRMF could be improved by designing a more efficient method for calculating each vessel's distance to the nearest other vessel: although the proposed KD-Trees are very efficient, they are limited by the need for each worker node to constructs a full KD-Tree with the position of all vessels. However, the performance gain from doing so would be questionable since constructing and querying the KD-Trees took only a small fraction of the total processing time in our experiments.

Appendix A

Regular Expressions for Detecting Locations

Island `(([A-Z]([a-z]|-[A-Z])+ )+(i|I)slands?)|(i|I)slands? of ( [A-Z]([a-z]|-[A-Z])+)+`

Coordinates `(?:lat(?:itude)?:? ?)?(?:(?:(?:\d|\.|:| |°|'|"|,)+) (?:north|east|south|west|n|e|s|w))(?: |,|:|longitude|long|-|\) |\()*(?:(?:\d(?:\d|\.|:| |°|'|"|,)+)(?:north|east|south|west|n|e|s|w))`

Relative Location `(?:(?:one |two |three |four |five |six |seven |eight |nine |ten)|(?:\d(?:,|\d|\.)*)) ?(?:nautical miles|nm|miles|kilometers) (?:\(.*\))?(?:to the)?(?:(?:South|East|North|West|southwest|southeast| northwest|northeast|N|E|S|W|NW|NE|SE|SW)(?: |-))*(?:off of|offshore|off |of|from|from the|out from)(?: the)?((?:east|west|north|south|-)*(ern)?)`

Anchorage `[A-Z][a-z]* (A|a)nchorage`

Gulf/Strait `(?:Port|Gulf|Strait|gulf|strait)(?: of)? [A-Z][a-z]*`

Port `(?:Port|port)(?: of)? [A-Z][a-z]*`

Coast `(?:[A-Z][a-z]* )+(C|c)oast(\W|$)(?!Guard)|`
`(?:west(?:ern)? |north(?:ern)? |south(?:ern)? |east(?:ern)? )?coast`
`of [A-Z][a-zA-Z]+`

River `(?:[A-Z](?:[a-z]|-[A-Za-z]| [A-Z])+) River([â-zA-Z]|$)`

Appendix B

Regular Expressions for Detecting Incident Types

Hijacked `hijack\W|hijacked|took control|hijacking|seize|was pirated`

Kidnapped `kidnap(ped|ping|\W)|abduct|hostage`

Shots Fired `killed|fired (?:on|upon|back)|(?<!warning )shot|(?:returned|opened) fire| kill(ed)? |(?:fire|firing|fired) (?!warning|hose|pump)|AK-47|RPG| rocket.propelled.grenade`

Attacked `attack(?:ed)?`

Boarded `(?<!on)\Wboard\W|boarded|stormed|unauthorized boarding`

Attempted Boarding `(try|attempt).{0,10}board[\w]*|(?:(?:failed|attempt|try |tried)\w*)(?: \w*){0,2} (?:on)?board`

Approached `chasing|chased|suspicious approach|came alongside|suspicious`

Evaded `evade|thwart|foil|unsuccessful|prevented|failed|repel|deter`

Robbed `robbed`

Sank `sank`

Appendix C

Regular Expressions for Detecting Vessel Types

Cargo Ship `(?:car|vehicle) carrier|(?:cargo|container) (?:ship|vessel)|bulk carrier|containership|freighter|bulker|ro/?ro|livestock carrier`

Tanker `(?:oil|products?|chemical|fuel|gas|LPG|LNG) ?(?:tanker|carrier|ship)|tanker|(?:chemical/)?oil products tanker|ocean rig drillship`

Fishing Vessel `fishing (?:boat|vessel)`

Military/Security `naval ship|warship|patrol vessel|frigate|security vessel`

Supply Vessel `supply vessel|offshore supply vessel|osv|crewboat|food-aid ship`

Tug Boat `tug|tugboat|tug boat`

Dhow `dhow`

Yacht `yacht`

Cruise Ship `cruise ship`

Accommodation Barge `accommodation barge`

Generic Vessel `(?:merchant|commercial) (?:ship|vessel)`

Bibliography

[1] A. Teske, R. Falcon, R. Abielmona, and E. Petriu. Genetic fuzzy system for automating maritime risk assessment. In *Uncertainty Management with Fuzzy and Rough Sets: Recent Advances and Applications*, Upcoming - Fall 2018. Reproduced with permission of SNCSC.

[2] A. Teske, R. Falcon, R. Abielmona, and E. Petriu. Automatic identification of maritime incidents from unstructured articles. In *2018 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, pages 42–48, June 2018. ©IEEE. Reprinted with permission.

[3] R. Falcon, A. Nayak, and R. Abielmona. An evolving risk management framework for wireless sensor networks. In *2011 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proceedings*, pages 1–6, Sept 2011. doi: 10.1109/CIMSA.2011.6059924.

[4] R. Falcon and R. Abielmona. A response-aware risk management framework for search-and-rescue operations. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, June 2012. doi: 10.1109/CEC.2012.6256538.

[5] Rami Abielmona. Tackling big data in maritime domain awareness. *Vanguard Magazine*, Aug/Sep:42–43, 2013.

[6] International Association of Classification Societies. A guide to risk assessment in ship operations. 2012.

[7] X. Tan, Y. Zhang, X. Cui, and H. Xi. Using hidden markov models to evaluate the real-time risks of network. In *Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008. IEEE International Symposium on*, pages 490–493, Dec 2008. doi: 10.1109/KAMW.2008.4810531.

[8] Yuping Wang, Yiu-ming Cheung, and Hailin Liu, editors. *Computational Intelligence and Security, International Conference, CIS 2006, Guangzhou, China, November 3-6, 2006, Revised Selected Papers*, volume 4456 of *Lecture Notes in Computer Science*, 2007. Springer. ISBN 978-3-540-74376-7.

[9] R. Falcon, R. Abielmona, S. Billings, A. Plachkov, and H. Abbass. Risk management with hard-soft data fusion in maritime domain awareness. In *the 2014 Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pages 1–8, Dec 2014. doi: 10.1109/CISDA.2014.7035641.

[10] Jinfen Zhang, Ângelo P Teixeira, C. Guedes Soares, Xinping Yan, and Kezhong Liu. Maritime transportation risk assessment of tianjin port with bayesian belief networks. *Risk Analysis*, 36(6):1171–1187, 2016. ISSN 1539-6924. doi: 10.1111/risa.12519. URL `http://dx.doi.org/10.1111/risa.12519`.

[11] Leonardo Vanneschi, Mauro Castelli, Ernesto Costa, Alessandro Re, Henrique Vaz, Victor Lobo, and Paulo Urbano. *Improving Maritime Awareness with Semantic Genetic Programming and Linear Scaling: Prediction of Vessels Position Based on AIS Data*, pages 732–744. Springer International Publishing, Cham, 2015. ISBN 978-3-319-16549-3. doi: 10.1007/978-3-319-16549-3_59. URL `http://dx.doi.org/10.1007/978-3-319-16549-3_59`.

[12] Rafael Falcon and Rami Abielmona. A Response-Aware Risk Management Framework for Search-and-Rescue Operations. In *2012 IEEE Congress on Evolutionary Computation (CEC)*, pages 1540–1547, Brisbane, Australia, June 2012.

[13] Rafael Falcon, Benjamin Desjardins, Rami Abielmona, and Emil Petriu. Context-driven dynamic Risk Management for maritime domain awareness. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, dec 2016. ISBN 978-1-5090-4240-1. doi: 10.1109/SSCI.2016.7850070. URL `http://ieeexplore.ieee.org/document/7850070/`.

[14] N. Friedman. *The Naval Institute Guide to World Naval Weapon Systems*. Naval Institute Press, 2006. ISBN 1557502625,9781557502629.

[15] K.E. Moore. Predictive analysis for naval deployment activities. *PANDA BAA*, (05-44), 2005.

[16] I. Lim and F. Jau. Comprehensive maritime domain awareness: an idea whose time has come? *Defence, Terrorism and Security, Globalisation and International Trade*, October 2007.

[17] Amir H. Razavi, Diana Inkpen, Rafael Falcon, and Rami Abielmona. Textual risk mining for maritime situational awareness. *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, CogSIMA 2014*, (2):167–173, 2014. doi: 10.1109/CogSIMA.2014.6816558.

[18] E H Mamdani. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. URL `https://pdfs.semanticscholar.org/5f51/e8e90ee966a787ad5ee56506769f8d11d81d.pdf`.

[19] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1):116–132, jan 1985. ISSN 0018-9472. doi: 10.1109/TSMC.1985.6313399. URL `http://ieeexplore.ieee.org/document/6313399/`.

[20] Chuck Karr. Genetic algorithms for fuzzy controllers. *AI Expert*, 6(2):26–33, February 1991. ISSN 0888-3785. URL `http://dl.acm.org/citation.cfm?id=129459.129463`.

[21] Manuel Valenzuela-Rendón. The Fuzzy Classifier System: a Classifier System for Continuously Varying Variables, 1991.

[22] Francisco Herrera and Luis Magdalena. Genetic Fuzzy Systems: A Tutorial. *Tatra Mountains Mathematical Publications*, 13(June 1997):93–121, 1997.

[23] Philip R Thrift. Fuzzy logic synthesis with genetic algorithms. 1991.

[24] Francisco Herrera. Genetic fuzzy systems: Taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1(1):27–46, 2008. ISSN 18645909. doi: 10.1007/s12065-007-0001-5.

[25] W Dong, Zhengxing Huang, Lei Ji, and Huilong Duan. A genetic fuzzy system for unstable angina risk assessment. *BMC Med Inform Decis Mak*, 14:12, 2014. ISSN 1472-6947. doi: 10.1186/1472-6947-14-12\r1472-6947-14-12[pii]. URL `http://www.ncbi.nlm.nih.gov/pubmed/24548742`.

[26] Mahyar Taghizadeh Nouei, Ali Vahidian Kamyad, MahmoodReza Sarzaeem, and Somayeh Ghazalbash. Developing a genetic fuzzy system for risk assessment of mortality after cardiac surgery. *Journal of Medical Systems*, 38 (10):102, 2014. ISSN 1573-689X. doi: 10.1007/s10916-014-0102-5. URL `http://dx.doi.org/10.1007/s10916-014-0102-5`.

[27] José Luis Aznarte, Jesús Alcalá-Fdez, Antonio Arauzo-Azofra, and José Manuel Benítez. Financial time series forecasting with a bio-inspired fuzzy model. *Expert Systems with Applications*, 39(16):12302 – 12309, 2012. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2012.02.135. URL `http://www.sciencedirect.com/science/article/pii/S0957417412003983`.

[28] Chih-Feng Liu, Chi-Yuan Yeh, and Shie-Jue Lee. Application of type-2 neuro-fuzzy modeling in stock price prediction. *Applied Soft Computing*, 12(4):1348 – 1358, 2012. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2011.11.028. URL http://www.sciencedirect.com/science/article/pii/S1568494611004704.

[29] Francisco Serdio, Edwin Lughofer, Kurt Pichler, Thomas Buchegger, and Hajrudin Efendic. Residual-based fault detection using soft computing techniques for condition monitoring at rolling mills. *Information Sciences*, 259:304 – 320, 2014. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2013.06.045. URL http://www.sciencedirect.com/science/article/pii/S0020025513004829.

[30] Azizul Azhar Ramli, Junzo Watada, and Witold Pedrycz. A combination of genetic algorithm-based fuzzy c-means with a convex hull-based regression for real-time fuzzy switching regression analysis: application to industrial intelligent data analysis. *IEEJ Transactions on Electrical and Electronic Engineering*, 9(1):71–82, 2014. ISSN 1931-4981. doi: 10.1002/tee.21938. URL http://dx.doi.org/10.1002/tee.21938.

[31] Alberto Fernández, Victoria López, María José del Jesus Del Jesus, and Francisco Herrera. Revisiting Evolutionary Fuzzy Systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems*, 80:109–121, 2015. ISSN 09507051. doi: 10.1016/j.knosys.2015.01.013.

[32] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera. Keel: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2009. ISSN 1433-7479. doi: 10.1007/s00500-008-0323-y. URL http://dx.doi.org/10.1007/s00500-008-0323-y.

[33] Isaac Triguero, Sergio González, Jose M Moyano, Salvador García, Jesús Alcalá-Fdez, Julián Luengo, Alberto Fernández, Maria José del Jesús, Luciano Sánchez, and Francisco Herrera. Keel 3.0: an open source software for multi-stage analysis in data mining. *International Journal of Computational Intelligence Systems*, 10(1):1238–1249, 2017.

[34] Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. 2001. ISSN 03600300. doi: 10.1145/505282.505283. URL `http://arxiv.org/abs/cs/0110053`.

[35] Sunita Sarawagi et al. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377, 2008.

[36] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1):233–272, Feb 1999. ISSN 1573-0565. doi: 10.1023/A:1007562322031. URL `https://doi.org/10.1023/A:1007562322031`.

[37] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL `http://dl.acm.org/citation.cfm?id=645530.655813`.

[38] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1625275.1625705`.

[39] Ralph Grishman. *Information extraction: Techniques and challenges*, volume 1299 of *Lecture Notes in Computer Science (including subseries Lecture Notes*

in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 11–27. Springer Verlag, 1997. ISBN 354063438X.

[40] Sunita Sarawagi. *Information Extraction*. Now Foundations and Trends, 2008. ISBN 9781601981882. doi: 10.1561/1900000003. URL `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=8187514`.

[41] Rexy Arulanandam, Bastin Tony Roy Savarimuthu, and Maryam A. Purvis. Extracting Crime Information from Online Newspaper Articles. *Proceedings of the Second Australasian Web Conference (AWC 2014), Auckland, New Zealand*, (Awc):31–38, 2014.

[42] Alberto Téllez-Valero, Manuel Montes, and Luis Villaseñor-Pineda. Using machine learning for extracting information from natural disaster news reports. 13, 09 2009.

[43] Ajay D. Kshemkalyani and Mukesh Singhal. *Distributed Computing: Principles, Algorithms, and Systems*. Cambridge University Press, New York, NY, USA, 1 edition, 2008. ISBN 0521876346.

[44] Eric Aubanel. *Elements of Parallel Computing*. Chapman & Hall/CRC, 1st edition, 2016. ISBN 1498727891, 9781498727891.

[45] Blaise Barney et al. Introduction to parallel computing. *Lawrence Livermore National Laboratory*, 6(13):10, 2010.

[46] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 2–2. USENIX Association, 2012.

[47] Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.

[48] F.J. Berlanga, A.J. Rivera, M.J. del Jesus, and F. Herrera. Gp-coach: Genetic programming based learning of compact and accurate fuzzy rule based classification systems for high dimensional problems. *Information Sciences*, 180(8):1183–1200, 2010.

[49] H. Ishibuchi, T. Nakashima, and T. Murata. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems and Man and and Cybernetics and Part B: Cybernetics*, 29(5):601–618, 1999.

[50] E.G. Mansoori, M.J. Zolghadri, and S.D. Katebi. Sgerd: A steady-state genetic algorithm for extracting fuzzy classification rules from data. *IEEE Transactions on Fuzzy Systems*, 16(4):1061–1071, 2008.

[51] H. Ishibuchi, T. Yamamoto, and T. Nakashima. Hybridization of fuzzy gbml approaches for pattern classification problems. *IEEE Transactions on Systems and Man and Cybernetics - Part B: Cybernetics*, 35(2):359–365, 2005.

[52] L. Sánchez, I. Couso, and J.A. Corrales. Combining gp operators with sa search to evolve fuzzy rule based classifiers. *Information Sciences*, 136(1-4): 175–192, 2001.

[53] J. Sanz, A. Fernandez, H. Bustince, and F. Herrera. Ivturs: a linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy

reasoning method with tuning and rule selection. *IEEE Transactions on Fuzzy Systems*, 21(3):399–411, 2013.

[54] D. Garcia, A. Gonzalez, and R. Perez. Overview of the slave learning algorithm: A review of its evolution and prospects. *International Journal of Computational Intelligence Systems*, 7(6), 2014.

[55] J. Otero and L. Sánchez. Induction of descriptive fuzzy classifiers with the logitboost algorithm. *Soft Computing*, 10(9):825–835, 2006.

[56] L. Sánchez and J. Otero. Boosting fuzzy rules in classification problems under single-winner inference. *International Journal of Intelligent Systems*, 22(9): 1021–1034, 2007.

[57] M.J. del Jesus, F. Hoffmann, L. Junco, and L. Sánchez. Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms. *IEEE Transactions on Fuzzy Systems*, 12(3):296–308, 2004.

[58] M. J. Gacto, R. Alcalá, and F. Herrera. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, 181(20):4340–4360, 2011. ISSN 00200255. doi: 10.1016/j.ins.2011.02.021. URL `http://dx.doi.org/10.1016/j.ins.2011.02.021`.

[59] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

[60] Krishnadev Calamur. High traffic, high risk in the strait of malacca. *The Atlantic*, Aug 2017. URL `https://www.theatlantic.com/international/archive/2017/08/strait-of-malacca-uss-john-mccain/537471`.

[61] Eibe Frank, Mark A Hall, and Ian H Witten. The WEKA Workbench. *Morgan Kaufmann, Fourth Edition*, pages 553–571, 2016. URL `https://www.cs.waikato.ac.nz/ml/weka/Witten{_}et{_}al{_}2016{_}appendix.pdf`.

[62] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[63] S. le Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

[64] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[65] Nathaniel Bowditch. Weather Routing. *The American Practical Navigator: an Epitome of Navigation*, page 896, 2002. URL `http://msi.nga.mil/MSISiteContent/StaticFiles/NAV{_}PUBS/APN/Chapt-38.pdf`.

[66] World Meteorological Organization. Guide to GRIB, 2003. URL `https://www.wmo.int/pages/prog/www/WDM/Guides/Guide-binary-2.html`.