# CONTEXTUAL BEHAVIOURAL MODELLING AND CLASSIFICATION OF VESSELS IN A MARITIME PIRACY SITUATION

by

**Joel Janek Dabrowski**

Submitted in partial fulfilment of the requirements for the degree

Philosophiae Doctor (Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology
UNIVERSITY OF PRETORIA

June 2014

# SUMMARY

## CONTEXTUAL BEHAVIOURAL MODELLING AND CLASSIFICATION OF VESSELS IN A MARITIME PIRACY SITUATION

by

**Joel Janek Dabrowski**

| | |
|---|---|
| Promoter(s): | Dr. Johan Pieter de Villiers |
| Department: | Electrical, Electronic and Computer Engineering |
| University: | University of Pretoria |
| Degree: | Philosophiae Doctor (Engineering) |
| Keywords: | Maritime Domain Awareness, Maritime piracy, Multi-Agent Simulation, Information Fusion, Dynamic Bayesian Network, Switching Linear Dynamic System, Classification, Inference, Behavioural Modelling, Sequential Analysis. |

In this study, a method is developed for modelling and classifying behaviour of maritime vessels in a piracy situation. Prior knowledge is used to construct a probabilistic graphical model of maritime vessel behaviour. This model is a novel variant of a dynamic Bayesian network (DBN), that extends the switching linear dynamic system (SLDS) to accommodate contextual information. A generative model and a classifier model are developed. The purpose of the generative model is to generate simulated data by modelling the behaviour of fishing vessels, transport vessels and pirate vessels in a maritime piracy situation. The vessels move, interact and perform various activities on a predefined map. A novel methodology for evaluating and optimising the generative model is proposed. This methodology can easily be adapted to other applications. The model is evaluated by comparing simulation results with 2011 pirate attack reports. The classifier model classifies maritime vessels into predefined categories according to their behaviour. The classification is performed by inferring the class of a vessel as a fishing, transport or pirate vessel class. The classification method is evaluated by classifying the data generated by the generative model and comparing it to the true classes of the simulated vessels.

# OPSOMMING

---

## KONTEKSTUELE GEDRAGS MODELLE AND KLASSIFIKASIE VAN VOORWERPE IN 'N MARITIEME SEEROWER SITUASIE

deur

**Joel Janek Dabrowski**

| | |
|---|---|
| Promotor(s): | Dr. Johan Pieter de Villiers |
| Departement: | Elektriese, Elektroniese en Rekenaar-Ingenieurswese |
| Universiteit: | Universiteit van Pretoria |
| Graad: | Philosophiae Doctor (Ingenieurswese) |
| Sleutelwoorde: | Maritieme Domain bewustheid, zeepiraterij, Multi-agent Mobielening, Inligting Fusion, Dynamic Bayesiaanse Network, skakel Lineêre dinamiese sisteem, klassifikasie, Inferensie, houding Modelling, Sekwensiële Ontleding. |

In hierdie studie word 'n metode word ontwikkel om die gedrag van maritieme vaartuie in 'n seerower situasie te modelleer en te klassifiseer. Voorafkennis word gebruik om 'n grafiese waarskynlikheidsmodel van maritieme vaartuig gedrag te ontwikkel. Hierdie model is 'n nuwe variant van 'n dinamiese Bayesiese netwerk (DBN), waar die skakelende lineêre dinamiese stelsel (SLDS) uitgebrei word om konteksinligting te akkomodeer. 'n Generatiewe model en 'n klassifiseerder model is ontwikkel as deel van hierdie studie. Die doel van die generatiewe model is om gesimuleerde data te genereer deur die gedrag van vissersbote, vervoerskepe en seerowervaartuie in 'n seerowersituasie te modelleer. Die vaartuie beweeg, het interaksie met mekaar en voer verskeie gesimuleerde aktiwiteite uit op 'n voorafgedefinieerde kaart. 'n Nuwe metode vir die evaluering en die optimering van die generatiewe model word voorgestel. Hierdie metode kan maklik aangepas word vir ander toepassings. Die model is geëvalueer deur simulasie resultate met 2011 seerower aanval data te vergelyk. Die klassifikasie model klassifiseer maritieme vaartuie volgens gedefinieerde kategorieë na gelang van hul gedrag. Die klassifikasie word uitgevoer deur die klas van 'n vaartuig te bepaal as 'n vissersboot, 'n vervoerskip of 'n seerowervaartuig. Die klassifikasie metode word geëvalueer deur die data te klassifiseer wat gegenereer word deur die generatiewe model, en dit met ware klasse van die gesimuleerde vaartuie te vergelyk.

# ACKNOWLEDGEMENTS

The author would like to thank:

The Council for Scientific and Industrial Research (CSIR) for their support in funding this research in years 2012 and 2013, with funding obtained from the Advanced Sensors and Electronics Defence (ASED) of the King Abdulaziz City for Science and Technology (KACST).

The National Research Fund (NRF) of South Africa for their support in funding this research in the year 2014.

My study leader, Dr. Pieter de Villiers. I am sincerely grateful for the guidance and support he has provided throughout this study. I am thankful for the countless proof readings he has performed. I am grateful for the many hours he has given in discussions, be it in person or via email. I have learned an enormous amount from him.

My wife, Joeline and my daughter Emma. I am especially grateful for their love, encouragement and support. I would also like to thank Joeline for her input and the reviews she has performed.

My father, Mike Dabrowski for his in-depth and detailed review of this dissertation.

My parents (Mike and Lindy), my sister (Aimee) and my parents in law (Ian and Glennys) for their support and encouragement.

Most importantly, I would like to thank God; “For the LORD gives wisdom; from His mouth comes knowledge and understanding” (Proverbs 2:7, English Standard Version).

# LIST OF ABBREVIATIONS

| | |
|---|---|
| DB | Bayesian Network |
| DBN | Dynamic Bayesian Network |
| LDS | Linear Dynamic System |
| SLDS | Switching Linear Dynamic System |
| MABS | Multi-agent Based Simulation |
| ID | Influence Diagram |
| GMM | Gaussian Mixture Model |
| EM | Expectation Maximization |
| E-step | Expectation-Step |
| M-step | Maximization-Step |
| GSF | Gaussian Sum Filtering |
| GSS | Gaussian Sum Smoothing |
| AIS | Automatic Identification System |
| HMM | Hidden Markov Model |
| STDF | State Transition Data Fusion |
| DP | Dirichlet Process |
| HDP | Hierarchical Dirichlet Process |
| VAR | Vector Autoregressive (process) |
| ICOADS | International Comprehensive Ocean-Atmosphere Data Set |
| ICC | International Chamber of Commerce |
| IMB | International Maritime Bureau |
| AAPA | American Association of Port Authorities |
| EU | European Union |
| NATO | North Atlantic Treaty Organisation |
| ASED | Advanced Sensors and Electronics Defence |
| KACST | King Abdulaziz City for Science and Technology |
| CSIR | Council for Scientific and Industrial Research |
| US | United States |

# NOTATION

Probability Theory:

| | |
|---|---|
| $p(x)$ | Probability of $x$. |
| $p(x,y)$ | Probability of $x$ and $y$. |
| $p(x \cap y)$ | Probability of $x$ and $y$. |
| $p(x\|y)$ | Probability of $x$ conditioned on $y$. |
| $X \perp\!\!\!\perp Y\|Z$ | Random variable $X$ is independent of random variable $Y$ conditioned on random variable $Z$. |
| $\mathcal{N}(x\|\mu,\Sigma)$ | Gaussian probability density function, parameterised by mean $\mu$ and covariance $\Sigma$. |
| $q(\cdot)$ | Approximate probability distribution. |

Linear Dynamic System (LDS) and Switching Linear Dynamic System (SLDS):

| | |
|---|---|
| $t$ | Time variable. |
| $h_t$ | State vector in a linear dynamic system. |
| $v_t$ | Observation vector in a linear dynamic system. |
| $u_t$ | Control vector in a linear dynamic system. |
| $s_t$ | Switching state variable. |
| $S$ | Number of switching states. |
| $\eta_h$ | Noise process in a linear dynamic system. |
| $\eta_v$ | Measurement noise in a linear dynamic system. |
| $\mathbf{A}$ | State transition or system matrix in a linear dynamic system. |
| $\mathbf{B}$ | Emission or observation matrix in a linear dynamic system. |
| $\mathbf{C}$ | Input matrix in a linear dynamic system. |

Generative Model:

| | |
|---|---|
| $C$ | Class variable/node. |
| $JP$ | Journey Parameters variable/node. |
| $EP$ | External Parameters variable/node. |
| $SP$ | State Parameters variable/node. |
| $OW$ | Ocean/Weather conditions variable/node. |
| $m$ | The $m^{\text{th}}$ sensor (sensor plate). |
| $M$ | Number of sensors (sensor plates). |
| $N$ | Number of maritime vessels. |
| $E$ | Number of external parameters. |
| $T$ | Number of simulation time steps. |

Gaussian Mixture Models:

| | |
|---|---|
| $\bar{b}$ | Vector consisting of the set of pirate attack locations from the 2011 IMB pirate attack dataset. |
| $\bar{b}_j$ | The $j^{\text{th}}$ pirate attack location in the set $\bar{b}$. |
| $\bar{d}$ | Vector consisting of the set of simulated pirate attack locations. |
| $\bar{d}_j$ | The $j^{\text{th}}$ pirate attack location in the set $\bar{d}$. |
| $g$ | Number of Gaussian mixture components. |
| $\psi_r$ | GMM parameters for the real-world dataset. |
| $\psi_s$ | GMM parameters for the simulated dataset. |
| $\phi_i()$ | The $i^{\text{th}}$ GMM component. |
| $\pi_i$ | The mixture weight of the $i^{\text{th}}$ Gaussian mixture component. |

Generative Model Evaluation:

| | |
|---|---|
| $\mathscr{L}(x)$ | The likelihood of $x$. |
| $\rho(p_1, p_2)$ | The Bhattacharyya coefficient between two discrete distributions $p_1$ and $p_2$. |
| $D_B(p_1, p_2)$ | The Bhattacharyya distance between two discrete distributions $p_1$ and $p_2$. |
| $H(p, q)$ | The cross entropy between distributions $p$ and $q$. |
| $K_i$ | Bayes factor for the $i^{\text{th}}$ sample of a dataset. |
| $D_T$ | Training set for the cross validation method. |
| $D_V$ | Validation set for the cross validation method. |
| $D_S$ | Simulation set for the cross validation method. |
| $\psi_T$ | Training set GMM parameters. |
| $\psi_S$ | Simulation set GMM parameters. |

Classification (Inference) Model (see also the notation for the LDS and SLDS):

| | |
|---|---|
| $c$ | Class variable/node. |
| $\bar{a}_t$ | Set of contextual element variables/nodes at time $t$. |
| $\bar{a}_{1:T}$ | Set of all contextual element variables/nodes from time $t = 1$ to time $t = T$. |
| $N_a$ | Number of contextual elements. |
| $v_{1:T}$ | Set of all observation vectors (measurements) from time $t = 1$ to time $t = T$. |

Classification (Inference) Model Evaluation:

| | |
|---|---|
| $R_j$ | Recall measure for class $j$. |
| $P_j$ | Precision measure for class $j$. |
| $F_j$ | F-score for class $j$. |

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

MARITIME PIRACY is a problem of international concern. Piracy poses an economic, humanitarian and environmental threat [1]. Thirty six percent of pirate attacks occurred in the high traffic shipping regions surrounding Somalia in the year 2011 [2]. The economic cost of piracy in Somalian waters during 2011 is estimated between 6.6 and 6.9 billion American dollars [3]. This translated to approximately fifty billion South African rand at the time. The costs associated with maritime piracy are a result of various factors These factors include increased vessel speed, re-routing, security equipment, insurance and military costs. Costs affect entities such as shipping companies, end-customers and governments.

At least 3863 seafarers were attacked by maritime pirates in the Somali region in 2011 [4]. Of these seafarers, a total of 561 hostages were captured by Somali pirates in 2011. Including hostages captured in 2010, a total of 1206 hostages were held by Somali pirates in 2011. The duration of captivity of hostages is estimated at an average of eight months [4].

In late 2008 three counter-piracy missions were deployed. These included the EU's Operation 'Atlanta', NATO's Operation 'Ocean Shield' and the US-led Combined Task Force-151 [5]. War ships have been deployed in these operations to patrol high risk regions and assist maritime piracy victims. Due to the vast patrol regions, efforts are only partially successful. The use of technology is proposed to assist in monitoring the regions [6].

In this study, automated methods for countering maritime piracy are presented. The final outcome of this study is a novel method for modelling and identifying maritime pirates based on their behaviour. Maritime vessels may be surveyed using electronic surveillance sensors. The sensors may be land based, air based or positioned on a marine vessel. The sensors provide information regarding the

kinematic properties and the position of marine vessels. The methods proposed in this study model vessel behaviour according to these properties. The proposed method uses the behavioural model to classify maritime vessels. The classes that are considered include transport vessels, fishing vessels and pirate vessels. By classifying a vessel as a pirate vessel, a potential threat warning may be issued. The proposed method may be useful to commercial or private vessels for the purpose of avoiding pirate attacks. Military patrolling vessels may utilise the proposed method for seeking out potential pirate vessels and threats. The methods developed in this study are not constrained to the maritime piracy application. They may be applied in various other applications and domains.

## 1.1 MOTIVATION AND PROBLEM STATEMENT

### 1.1.1 The Maritime Piracy Surveillance Problem

The increase in global maritime piracy has led to various counter-piracy efforts. Maritime surveillance is essential in these efforts. Effective patrolling of vast regions with limited military resources may be difficult. Electronic surveillance provides a compelling tool for counter-piracy efforts. Electronic surveillance of large regions however produces immense datasets. These datasets are generally too large for human operators to analyse efficiently.

### 1.1.2 Maritime Piracy Data Source Problem

An early detection or warning system strives to detect a threat before an attack occurs. Early detection and warning systems are generally developed according to prior knowledge of circumstances that lead to an event of interest. Data describing prior knowledge of pirate behaviour before an attack is scarce. More generally, any real-world data relating illegal activities such as maritime piracy is scarce [7, 8]. The International Maritime Bureau (IMB) publish maritime piracy reports [2]. Ship masters and operators are encouraged to report piracy incidents through the web-based IMB piracy reporting centre. The reports describe pirate attacks and are incidental by nature. They do not provide information the pirate behaviour before the attack.

### 1.1.3 Behavioural Modelling Problem

Behaviour may be described according to the activities an object of interest engages in. In this study, behaviour is defined by the kinematic activities of maritime vessels. The behaviour of a maritime

vessel is influenced by various contextual elements. These elements may include time, location, other vessels, weather and ocean conditions. A behaviour-set is often comprised of multiple nonlinear and discontinuous entities. The process of analysing and modelling behaviour can therefore be extremely complex and tedious.

### 1.1.4 Classification Problems

Owing to the large extent of the regions encountered in the maritime domain, many vessels may be considered for classification at a single point in time. Maritime pirate vessels are expected to be a minority among the set of vessels considered in a region. Consider a large dataset consisting of behavioural attributes of various vessels operating in such an area. Abnormal behaviour associated with piracy is will be sparse within the dataset. Modelling and detection of sparse anomalies within a large datasets is therefore non-trivial [9]. For example, a general machine learning classifier would require many examples of pirate behaviour to sufficiently learn from the data.

In this study, the behavioural data considered for classification is sequential in nature. The classifier is required to observe the surveyed object over time. In general, each observation of behaviour is subject to a previous observation. The classifier is required to consider the dependence between behaviour observations over time. General classification methods such as neural networks and support vector machines do not consider time variation of data in this manner.

### 1.1.5 Research Questions

Research questions are intended to encapsulate the research problems. The identified research problems will be addressed if the research questions are sufficiently answered. The research questions for this study are presented as follows:

**Research Questions:**

1. What describes maritime pirate vessel behaviour?
2. How can pirate vessel behaviour be modelled?
3. How can a pirate behavioural model be used to identify pirate vessels?

## 1.2 RESEARCH OBJECTIVES AND HYPOTHESES

The research objectives may be formulated according to the research problems and questions described in Section 1.1. For the maritime piracy and surveillance problem, a method is to be proposed for automatically identifying maritime pirate vessels. The maritime piracy data source problem is addressed by developing a method for generating synthetic data. To formulate the classifier and generative models, the behavioural modelling problem must be addressed. The construction of a behavioural model of maritime vessels is required. The behavioural model may be used for the purpose of synthetic data generation and for the classification of maritime vessel behaviour. The research objectives may thus be summarised as follows:

**Research Objectives:**

1. A set of behavioural attributes are to be identified that describe maritime vessel behaviour in a pirate situation.
2. A generative model is to be proposed for the generation of synthetic data of maritime vessel behaviour in a pirate situation.
3. A classification method is to be proposed for classifying maritime vessel behaviour in a pirate situation.

A classification method for identifying maritime pirates before they attack is required. If a vessel exhibits pirate behaviour, a pirate threat may be identified. The behavioural classification method is applied to classify fishing, transport and pirate behaviour. Maritime vessel behaviour may be described according to kinematic activity. The kinematic activity is to be modelled using a statistical model. This study is constrained to consider the dynamic Bayesian network (DBN) for modelling the kinematic behaviour. A generative DBN model of the behaviour may be used for generating synthetic maritime situation data. A DBN model of the behaviour may be used for classification of maritime vessels. Classification of vessels is performed by inference of a class variable in the DBN. The research hypotheses may be summarised as follows:

**Research Hypotheses:**

1. Kinematic behaviour of maritime vessels may be represented using a dynamic Bayesian network.
2. A dynamic Bayesian network behavioural model may be used to infer a vessel class.

## 1.3 RESEARCH METHODOLOGY

The research is composed of two phases. The first phase involves the development of a generative model. The objective of this model is to simulate maritime vessel behaviour in a piracy situation. A set of vessels are placed on a map. The generative model simulates the vessels kinematic activities over a period of time. The second phase of the study involves the development of a classification method for classifying maritime vessels in a piracy situation. The latter method is required to classify the maritime vessels simulated by the method developed in the first phase.

The research methodology may be described by the following procedures for each phase of the study:
**Phase 1:**

1. Identify behavioural attributes of various maritime vessel classes.
2. Define a generative DBN that models maritime vessel behaviour.
3. Generate maritime pirate data using the generative model.
4. Evaluate the generative model and data according to real-world maritime pirate data.

**Phase 2:**

1. Define a DBN for classifying maritime vessel behaviour.
2. Apply an inference algorithm to the DBN for classification.
3. Test and evaluate the classification method using the synthetic maritime pirate data generated in phase 1.

The outputs of the two phases of the study address the three research objectives described in Section 1.2. The first item of phase 1 addresses objective 1. The output of phase 1 addresses objective 2. The output of phase 2 addresses objective 3.

## 1.4 CONTRIBUTIONS

The intention of this research is to make contributions to research literature and to counter-piracy efforts. These contributions include the development of novel methods, models and evaluation techniques for representation and classification of behaviour. Contributions to the literature are disseminated in the following manuscripts:

**[J1]** J. J. Dabrowski, J. P. de Villiers, "Maritime piracy situation modelling with dynamic bayesian networks", Information Fusion 23 (2015) 116 - 130.

**[J2]** J. J. Dabrowski and J. P. de Villiers, "A Method for Classification and Context Based Behavioural Modelling of Dynamical Systems Applied to Maritime Piracy," submitted to Expert Systems with Applications.

In this research the DBN is applied to the problem of modelling stochastic behaviour. The DBNs proposed in this study are hybrid or mixed variable networks that contain both discrete and continuous variables. The following novel contributions are made:

- A novel generative DBN that models maritime vessel behaviour in a maritime piracy situation is proposed.
- A novel method for evaluating the fidelity of such generative models is proposed.
- A novel classifier DBN model that classifies maritime vessels according to their behaviour is proposed.
- Classification is performed by inference using the Gaussian sum filtering (GSF) algorithm. The GSF algorithm is specifically derived for the classifier DBN model.

The contribution to the counter-piracy effort is achieved by applying the research to the maritime piracy problem. The research provides a computerised simulation environment for generating maritime piracy data. It is intended that the software for this environment be made publicly available to other researchers and interested persons. The classification method that is developed in this study provides a means for maritime pirate vessel identification. This output provides a possible tool for the counter-piracy effort.

## 1.5 THESIS STRUCTURE

This thesis is structured according to the two methodology phases as described in Section 1.3. The first phase of the study considers a generative model for simulating vessels in a maritime piracy situation. The second phase considers a classification method for classifying vessels in a maritime piracy situation. Chapters are generally divided according to the two phases.

The foundational theory for this study is described in Chapter 2. This chapter includes a review on probability theory, Bayesian networks and linear dynamic systems. A literature study is presented

in Chapter 3. In this chapter, background and related work regarding applications to counter-piracy and DBNs is presented. Pirate behaviour, pirate tactics and maritime data sources are described in Chapter 4. The generative model for simulating vessel behaviour is presented in Chapter 5. The inference and classification method is presented in Chapter 6. Simulation results and an evaluation of the generative model are presented in Chapter 7. The inference and classification results are presented in Chapter 8. A generalised methodology of the presented methods is presented in Chapter 9. The study is concluded in Chapter 10.

# CHAPTER 2

# THEORETICAL FOUNDATIONS

THEORETICAL AND METHODOLOGICAL approaches to the study are presented in this chapter. A brief review on probability theory relating to this work is provided. Dynamic Bayesian networks (DBN) and their properties are discussed. An exposition on the switching linear dynamic system (SLDS) is presented. These methods form the foundation for the presentation of the generative model and classification model presented in this study.

## 2.1 PROBABILITY THEORY REVIEW

### 2.1.1 Probability and Events

Probability theory forms part of the foundation of the machine learning and pattern recognition fields. It provides a means to model data in terms of relative frequency and uncertainty. To draw such conclusions about a particular set of data, an experiment may be conducted. The set of all possible outcomes of the experiment is called the *sample space*. An *event* is any subset of possible outcomes of an experiment. The probability of an event may be described by its *relative frequency* of occurrence. The probability of an event may also be described as a *degree of belief* that the particular event occurred. The probability of an event $A$ is represented by a numeric value in the range of $p(A) \in [0,1]$. A high value of probability indicates a high certainty of the events occurrence.

The *relative frequency* view of probability is referred to as the 'frequentist' view [10]. Frequentists view events as noisy observations of some underlying probability model with a fixed unknown parameter. Probability is viewed as a long-run relative frequency over all possible outcomes given the unknown parameter. Inference in frequentist statistics involves the estimation of the fixed unknown

parameter.

The *degree of belief* view of probability is referred to as the 'Bayesian' view [10]. Bayesians view the parameter of an underlying probability model as a variable. Observations are considered to be fixed. Inference in the Bayesian paradigm involves updating the *degree of belief* of the parameters given the observed data. Inference includes the use of a prior density to inform the plausibility of each parameter value.

### 2.1.2 Random Variables and Functions

A *random variable* may be used to represent events in a numeric form. A random variable is defined as a function that maps events from the sample space $S$ to real numbers [11]. The random variable can map events to continuous or discrete numeric values. Probabilities may be associated with the numeric values of random variables.

Let $X$ denote the random variable of a particular event $A$. Let $x$ represent a particular value of the random variable $X$. The *cumulative probability distribution function* of event $A = \{X \leq x\}$ is defined as [12]

$$F_X(x) = p\{X \leq x\}. \tag{2.1}$$

The cumulative probability distribution function, or distribution function, is a monotonically increasing function that increases from zero to one. The *probability density function*, denoted by $f_X(x)$, is defined as the derivative of the distribution function [12]

$$f_X(x) = \frac{dF_X(x)}{dx}. \tag{2.2}$$

For the discrete case, the probability density function is given by [12]

$$f_X(x) = \sum_i p_i \delta(x - x_i). \tag{2.3}$$

where $\delta(x)$ is the impulse or Dirac delta function and $p_i = p\{x = x_i\}$. The term $p_i\delta(x - x_i)$ describes an impulse function at $x = x_i$ with height equal to $p_i$.

The values of the density function may fall within the range of zero to positive infinity. The integral of the density function from negative to positive infinity is unity. The density function describes how the probability is distributed over the space of a random variable $X$. It provides an indication of the likelihood of the occurrence of a particular value $x$ of random variable $X$. Density distributions may be

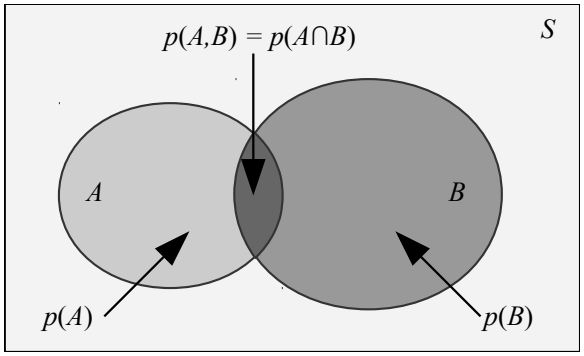



**Figure 2.1:** The joint probability of events $A$ and $B$ is represented as the intersection of the events in the Venn diagram.

represented by various functions. Commonly used functions include Gaussian, Binomial, Uniform, Exponential and Rayleigh functions. The Gaussian density function for the variable $x$ is given by [13], [12]

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{2.4}$$

### 2.1.3 Joint Probability

The joint probability of two or more events is the probability that all the referenced events occur together. The joint probability is defined as the intersection of a set of events $A_1, A_2, ..., A_n$ [12, 11]

$$p(A_1, A_2, ..., A_n) = p(A_1 \cap A_2 \cap ... \cap A_n). \tag{2.5}$$

The joint probability for events $A$ and $B$ in the sample space $S$ may be illustrated by the Venn diagram in Figure 2.1. The joint probability is represented by the region where events $A$ and $B$ overlap.

The joint probability distribution function for random variables $X$ and $Y$, denoted $F_{X,Y}(x,y)$, is defined as [12]

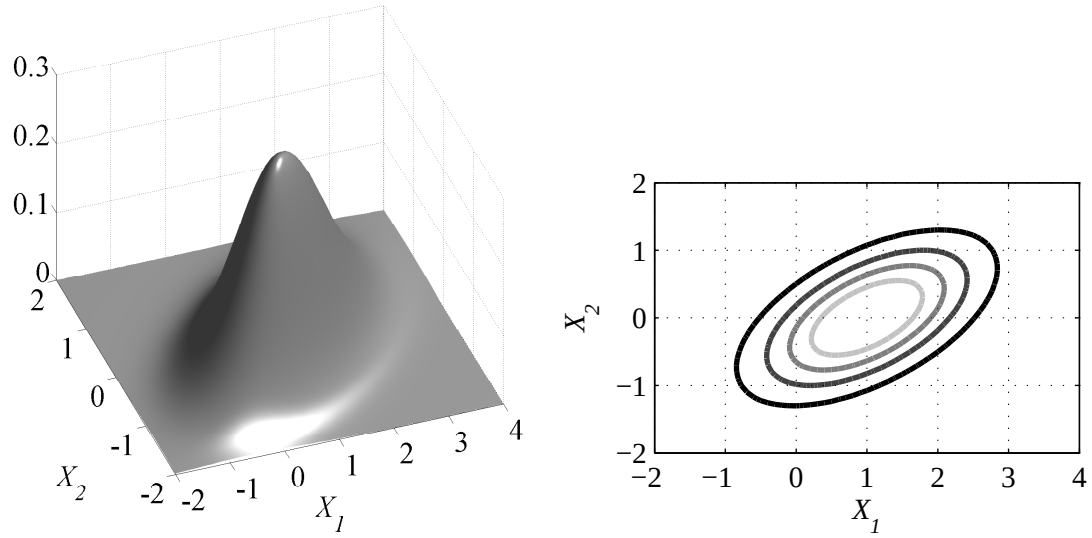
$$F_{X,Y}(x,y) = p\{X \leq x, Y \leq y\}. \tag{2.6}$$

The joint probability density function, denoted by $f_{X,Y}(x,y)$, is defined as [12]

$$f_{X,Y}(x,y) = \frac{\partial^2 F_{X,Y}(x,y)}{\partial x \partial y}. \tag{2.7}$$

The $d$-dimensional multivariate Gaussian probability density function is given as follows [13]

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right). \tag{2.8}$$




**(a)** Gaussian surface plot.

**(b)** Gaussian contour plot.

**Figure 2.2:** A surface and contour plot of a Gaussian joint probability density function of random variables $X_1$ and $X_2$. The lighter shaded contour rings in the contour plot indicate higher probability. The mean and covariance are given by (2.11) and (2.12)

A two dimensional Gaussian probability distribution over variables $X_1$ and $X_2$ is illustrated as the surface and contour plot in Figure 2.2.

The Gaussian distribution is parameterised by the mean $\mu$ and the covariance $\Sigma$ of the distribution. The mean is described by a $d$-dimensional vector as follows

$$\mu = [\mu_1, \mu_2, ..., \mu_d]^T . \tag{2.9}$$

The covariance is represented by a symmetric, positive definite matrix as follows

$$\Sigma = \begin{bmatrix} \sigma_1\sigma_1 & \sigma_1\sigma_2 & \cdots & \sigma_1\sigma_d \\ \sigma_2\sigma_1 & \sigma_2\sigma_2 & \cdots & \sigma_2\sigma_d \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_d\sigma_1 & \sigma_d\sigma_2 & \dots & \sigma_d\sigma_d \end{bmatrix} . \tag{2.10}$$

The inverse of the covariance matrix is the precision matrix $\Lambda \equiv \Sigma^{-1}$. The covariance matrix represents a measure of the degree of uncertainty of a random variable. The precision matrix represents a measure of the degree of certainty of a random variable.

The parameters of the Gaussian distribution represented in Figure 2.2 are provided as follows

$$\mu = [1,0]^T, \tag{2.11}$$

$$\Sigma = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 0.5 \end{bmatrix}. \tag{2.12}$$

The mean describes the Gaussian distributions centre. The diagonal elements of the covariance matrix, $\sigma_1^2$ and $\sigma_2^2$ describe the variance in the $X_1$ and $X_2$ directions respectively. The off-diagonal elements $\sigma_1\sigma_2 = \sigma_2\sigma_1$ describe the correlation between the random variables $X_1$ and $X_2$.

### 2.1.4 Marginal Probability

The *marginal density function* of a random variable is a density function that is a subset of a joint density function. The marginal density function is defined for a random variable in reference to multiple random variables in a joint distribution. Consider the random variables $X$ and $Y$. The marginal density functions are extracted from a joint density function of $X$ and $Y$ as follows [12]

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)dy, \tag{2.13}$$

$$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)dx. \tag{2.14}$$

For the discrete case, the marginal density function is given as

$$f_X(x_i) = \sum_{y_j} f_{X,Y}(x_i,y_j), \tag{2.15}$$

$$f_Y(y_j) = \sum_{x_i} f_{X,Y}(x_i,y_j). \tag{2.16}$$

The marginal density function is only valid within joint probability density functions. Consider Figure 2.1. The marginal density function is determined within the region where the two events $A$ and $B$ intersect.

Consider the Gaussian joint density function illustrated in Figure 2.2. The marginal distribution of $X_1$ is obtained by integrating over all values of $X_2$. The marginal probability that $X_1$ takes a particular value is obtained by integrating the joint distribution in the $x_2$ direction at the particular value of $X_1$ [13]. The marginal distribution is illustrated in Figure2.3.

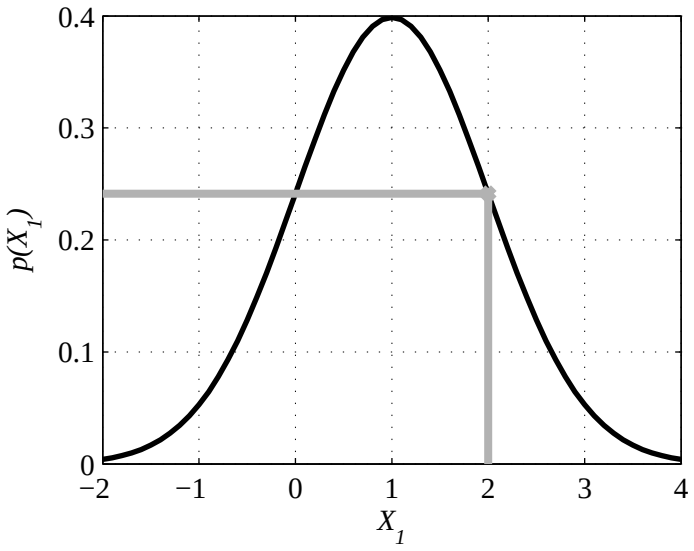



**Figure 2.3:** The marginal distribution of $f_X(x_1) = \int_{-\infty}^{\infty} f_{X_1,X_2}(x_1,x_2)dx_2$ is illustrated by the Gaussian function. The marginal probability of $f_X(x_1 = 2)$ is illustrated by the grey marker. The joint distribution, $f_{X_1,X_2}(x_1,x_2)$ is illustrated in Figure 2.2.

### 2.1.5 Conditional Probability

The *conditional probability* describes the probability of an event $A$, given that an event $B$ has occurred. The conditional probability of event $A$ given event $B$ is defined as follows [11], [12]

$$p(A|B) = \frac{p(A \cap B)}{p(B)} = \frac{p(A,B)}{p(B)}. \tag{2.17}$$

More generally, the conditional probability of events $A_1, A_2, ..., A_n$ given events $B_1, B_1, ..., B_m$ is defined as follows [11], [12]

$$p(A_1, ..., A_n|B_1, ..., B_m) = \frac{p(A_1 \cap A_2 \cap ... \cap A_n)}{p(B_1 \cap B_2 \cap ... \cap B_m)} = \frac{p(A_1, ..., A_n)}{p(B_1, ..., A_n)}. \tag{2.18}$$

The conditional probability essentially restricts the sample space to the space of the conditioned event. Consider the Venn diagram in Figure 2.1. The probability of event $A$ is the ratio of the space of event $A$ with respect to the sample space $S$. The conditional probability of $A$ given $B$ is the ratio of the intersection of events $A$ and $B$ to the probability of event $B$. This describes the ratio of the portion of the space of event $A$ that lies within $B$, with respect to the complete space of event $B$. By conditioning on event $B$, the sample space may be viewed as being restricted to the space of event $B$. The sample space is restricted due to the fact that the conditional indicates that the event occurred in space of $B$.

The conditional density function for random variables $X$ and $Y$ is defined as [11],[12]

$$f_X(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}, \tag{2.19}$$

$$f_Y(y|x) = \frac{f_{X,Y}(x,y)}{f_X(x)}. \tag{2.20}$$

The conditional density function may be computed from the joint and the marginal density functions [13, 11]. The numerator in (2.19) and (2.20) describes the joint density function. The denominator in (2.19) and (2.20) describe marginal density functions. Consider the Gaussian joint density function illustrated in Figure 2.2. The joint density function of $X_1$ given a particular value of $X_2$ is an infinitesimal slice of the joint distribution in the direction of $X_1$ at the particular value of $X_2$. The marginal density function $f_{X_2}$ is computed by integrating the joint density function over $X_1$ as described in Section 2.1.4. The conditional distribution of $f_{X_1}(X_1|X_2 = 1)$ is illustrated in Figure 2.4.

### 2.1.6 Total Probability

The law of total probability expresses marginal probabilities in the form of conditional probabilities. The conditional probability given in (2.19) may be rearranged in terms of the joint distribution as follows

$$f_{X,Y}(x,y) = f_Y(y|x)f_X(x). \tag{2.21}$$

The marginal density function is given in (2.12) and (2.13). Substituting (2.20) into (2.13) results in the following expression for total probability

$$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)dx = \int_{-\infty}^{\infty} f_Y(y|x)f_X(x))dx. \tag{2.22}$$

In discrete form, the law of total probability may be written as follows

$$f_Y(y_j) = \sum_{x_i} f_Y(y_j|x_i)f_X(x_i)). \tag{2.23}$$

In terms of events A and B, the law of total probability may be written as follows

$$p(A) = \sum_n p(A|B_n)p(B_n). \tag{2.24}$$

### 2.1.7 Statistical Independence

Two events are *statistically independent* if they do not have any apparent relationship to each other. That is, the occurrence of one event has no impact on the probability of the other event. Two events

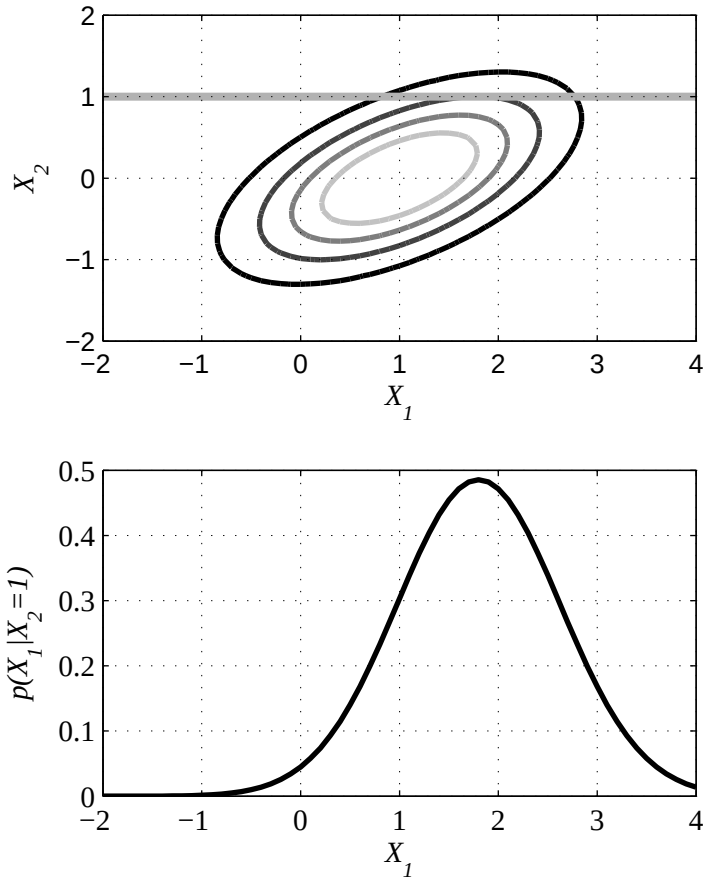



**Figure 2.4:** The conditional distribution of $f_{X_1}(X_1|X_2 = 1)$ is the infinitesimal slice of the joint distribution along the line $x_2 = 1$, which is $f_{X_1,X_2}(X_1, X_2 = 1)$, divided by the marginal density function $f_{X_2}(X_2 = 1)$. The line $x_2 = 1$ is illustrated by the grey horizontal line in the contour plot. The conditional distribution is represented by the Gaussian function in the bottom plot.

are statistically independent if the following two equations apply [11], [12]

$$p(A|B) = p(A) \tag{2.25}$$

$$p(B|A) = p(B). \tag{2.26}$$

This implies that [12]

$$p(A,B) = p(A)p(B) \tag{2.27}$$

Independence can be extended to multiple random variables. Event $A$ is conditionally independent of event $B$ given event $C$ if [14]

$$p(A|B,C) = p(A|C), \tag{2.28}$$

or if

$$p(B,C) = 0. \tag{2.29}$$

Furthermore, event $A$ is conditionally independent of event $B$ given event $C$ if [14]

$$p(A,B|C) = p(A|C)p(B|C). \tag{2.30}$$

Let the symbol '$\perp\!\!\!\perp$' denote conditional independence. Several important conditional independence properties for some random variables, $X$, $Y$, $W$ and $Z$ are given as follows [14]:
symmetry

$$(X \perp\!\!\!\perp Y|Z) \Rightarrow (Y \perp\!\!\!\perp X|Z), \tag{2.31}$$

decomposition

$$(X \perp\!\!\!\perp Y,W|Z) \Rightarrow (X \perp\!\!\!\perp Y|Z), \tag{2.32}$$

weak union

$$(X \perp\!\!\!\perp Y,W|Z) \Rightarrow (X \perp\!\!\!\perp Y|Z,W) \tag{2.33}$$

and contraction

$$(X \perp\!\!\!\perp W|Z,Y)\&(X \perp\!\!\!\perp Y|Z) \Rightarrow (X \perp\!\!\!\perp Y,W|Z). \tag{2.34}$$

### 2.1.8 The Chain Rule of Probabilities

Consider the events $A_1, A_2, ..., A_n$. The joint probability of these events may be written in terms of the conditional distribution given by (2.17) as follows

$$p(A_1, ..., A_n) = p(A_1|A_2, ..., A_n)p(A_2, ..., A_n). \tag{2.35}$$

The second term is a joint probability that may too be expressed using the definition of conditional probability as follows

$$p(A_1,...,A_n) = p(A_1|A_2,...,A_n)p(A_2|A_3...,A_n)p(A_3,...,A_n). \tag{2.36}$$

This process may be continued until all joint distribution terms have been expressed using the definition of conditional probability. The result of this process is given as follows

$$p(A_1,...,A_n) = p(A_1|A_2,...,A_n)p(A_2|A_3...,A_n)\cdots p(A_{n-2}|A_{n-1},A_n)p(A_{n-1}|A_n)p(A_n). \tag{2.37}$$

This equation may be written more compactly as follows

$$p(\cap_{k=1}^{n} A_k) = \prod_{k=1}^{n} p(A_k|\cap_{j=1}^{k-1} A_j). \tag{2.38}$$

### 2.1.9 Bayes' Theorem

The conditional probability given in (2.16) can be rearranged as an expression of the joint distribution as follows

$$p(A,B) = p(A|B)p(B). \tag{2.39}$$

Similarly

$$p(B,A) = p(B|A)p(A). \tag{2.40}$$

The joint distributions $p(A,B)$ and $p(B,A)$ are equivalent. Equations (2.39) and (2.40) are equated as follows

$$p(B|A)p(A) = p(A|B)p(B). \tag{2.41}$$

This result may be rearranged in the form of Bayes' theorem as follows

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}. \tag{2.42}$$

The probability $p(A|B)$ is called the posteriori which is the degree of belief of $A$ having encountered $B$. The probability $p(A)$ is the prior which is the initial degree of belief in $A$. The probability $p(B|A)$ is called the likelihood of $B$ given $A$. The marginal distribution, $p(B)$, functions as a normalizing factor which may be determined from the law of total probability as follows

$$p(B) = \sum_i p(B|A_i)p(A_i). \tag{2.43}$$

In terms of random variables $X$ and $Y$, Bayes' theorem is expressed as follows

$$f_{X|Y}(x|Y=y) = \frac{f_Y(y|X=x)f_X(x)}{f_Y(y)}. \tag{2.44}$$

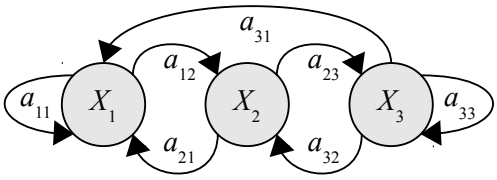



**Figure 2.5:** An example of a Markov model in a state transition diagram form.

The density function, $f_Y(y)$, may be calculated from the law of total probability as follows

$$f_Y(y) = \int f_Y(y|X = x) f_X(x) dx. \tag{2.45}$$

## 2.2 GRAPHICAL MODELS

The graphical model is a representation of a joint probability distribution in a graph or diagram format. A graphical model consists of nodes (or vertices) connected by links (or edges). The nodes describe random variables. The random variables may be continuous or discrete. The links describe probabilistic relationships between the random variables [13]. Properties such as conditional independence are embodied in the structure of the graphical model. Markov chains, hidden Markov models, Bayesian networks, dynamic Bayesian networks and Markov random fields are all examples of graphical models.

### 2.2.1 The Markov Model

The Markov model is a stochastic model containing a set of random variables that are related by the Markov property. The Markov property limits the dependence between random variables over time. An example of a Markov model that transitions between a set of states $\{X_1, X_2, X3\}$ is illustrated as a state transition diagram in Figure 2.5. The system transitions between the states $X_1$, $X_2$ and $X_3$ according to the conditional probabilities $a_{ij} = p(X_j|X_i)$. The conditional probabilities in Figure 2.5 may be represented as the transition matrix

$$P = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

The $i^{\text{th}}$ row of the transition matrix describes all the outgoing transition probabilities from the $i^{\text{th}}$ node. Each column along the row describes the $j^{\text{th}}$ node that the $i^{\text{th}}$ node transitions too. For example, $a_{12}$

describes the probability of transitioning from node 1 to node 2. The rows of the transition matrix sum to unity.

The state of the system at any point in time is modelled with a random variable $x_t$. The model in Figure 2.5 is with random variable $x_t \in \{X_1, X_2, X_3\}$. The random variables $x_{0:T} = x_0, \ldots, x_T$ are related by the Markov property. The $K^{\text{th}}$ order Markov property assumes that the state at time $t$ is conditioned only on the previous $K$ states. The $K^{\text{th}}$ order Markov assumption in time may be expressed as

$$p(x_{0:T}) = \prod_{t=0}^{T-1} p(x_t | x_{t-K:t-1}). \tag{2.46}$$

The first order Markov model assumes that $x_t$ is only conditioned on its previous state $x_{t-1}$. The Markov chain is a first order Markov process [15]. The Markov chain is defined such that [15, 14]

$$p(x_{0:T}) = \prod_{t=0}^{T-1} p(x_t | x_{t-1}). \tag{2.47}$$

The Markov assumption simplifies the analysis of time series. For example, consider the sequence $X_1, X_2$ given the initial state of $X_3$ for the model illustrated in Figure 2.5. If the model is assumed to be a first order Markov process, the joint probability if the sequence is calculated using the chain rule of probability. The probability of the sequence is computed as follows

$$\begin{aligned} p(x_1 = X_2, x_2 = X_1 | x_3 = X_3) &= p(x_1 = X_2 | x_2 = X_1, x_3 = X_3) p(x_2 = X_1 | x_3 = X_3) \\ &= p(x_1 = X_2 | x_2 = X_1) p(x_2 = X_1 | x_3 = X_3) \\ &= p(X_2 | X_1) p(X_1 | X_3) \\ &= a_{12} a_{31}. \end{aligned}$$

The factor $p(x_1 = X_2 | x_2 = X_1, x_3 = X_3)$ is reduced to $p(x_1 = X_2 | x_2 = X_1)$ according to the Markov property.

A useful property to consider for a Markov model is the stationary distribution. The stationary distribution of a Markov model is the distribution over the states as $t \to \infty$. The stationary distribution provides an indication of the probability of being in a particular state at any time. The definition of a stationary distribution is given as follows [15]:

> **Definition 1** *The vector $\pi$ is called a stationary distribution of the chain with state transition probability matrix P if $\pi$ has entries $(\pi_j : j \in S)$ such that:*

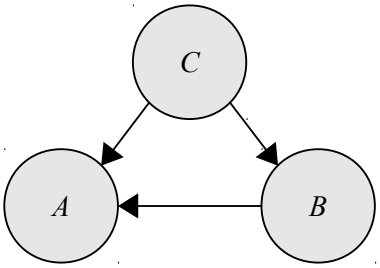



**Figure 2.6:** The BN for the joint distribution given by (2.46).

1. $\pi_j \geq 0$ *for all* $j$ *and* $\sum_j \pi_j = 1$,
2. $\pi = P\pi$, *which is to say that* $\pi_j = \sum_i \pi_i p_{ij}$ *for all* $j$.

The stationary distribution is determined by solving $\pi = P\pi$ for $\pi$ with the property $\sum_j \pi_j = 1$.

### 2.2.2 The Bayesian Network

Many problems involve the analysis of multiple random variables. The joint distribution may be considered for analysing such problems. The explicit representation of a joint distribution is often intractable. For example, the representation of a joint distribution over a set of $n$ binary random variables requires $2^n - 1$ numbers to be specified [14]. Increasing the 'resolution' of the random variables results in a significant increase in size and complexity of the joint distribution. The Bayesian network exploits independence properties to form more compact representations of the joint distribution [14].

The Bayesian network (BN) is a directed graphical model. The links between the nodes have a specified direction and are indicated by arrows. The directions of the links indicate causality. An arrow from a node $A$ to a node $B$ indicates that random variable $A$ has a causal effect on random variable $B$.

As an example, consider the joint distribution $p(A,B,C)$ over variables $A$, $B$ and $C$. By the chain rule of probability, the joint distribution can be written as

$$p(A,B,C) = p(A|B,C)p(B|C)p(C). \tag{2.48}$$

The BN for this representation is illustrated in Figure 2.6.

A practical example of the BN in Figure 2.6 is a reasoning on why a patch of grass in a garden is

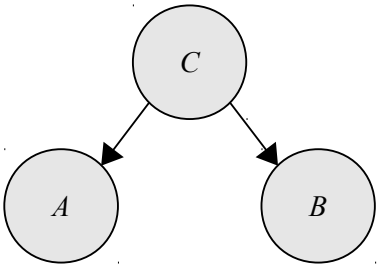



**Figure 2.7:** The BN for the joint distribution given by (2.2.1).

wet. The grass could be wet due to rain or due to a sprinkler. The variable $A$ in Figure 2.6 represents the grass being wet. The variable $B$ represents the sprinkler. The variable $C$ represents rain. The wet grass variable $A$ is conditioned on the rain $C$ and sprinkler $B$ variables. The sprinkler variable $B$ is conditioned on the rain variable $C$. This conditioning is due to the fact that the sprinkler will not be turned on if it is raining.

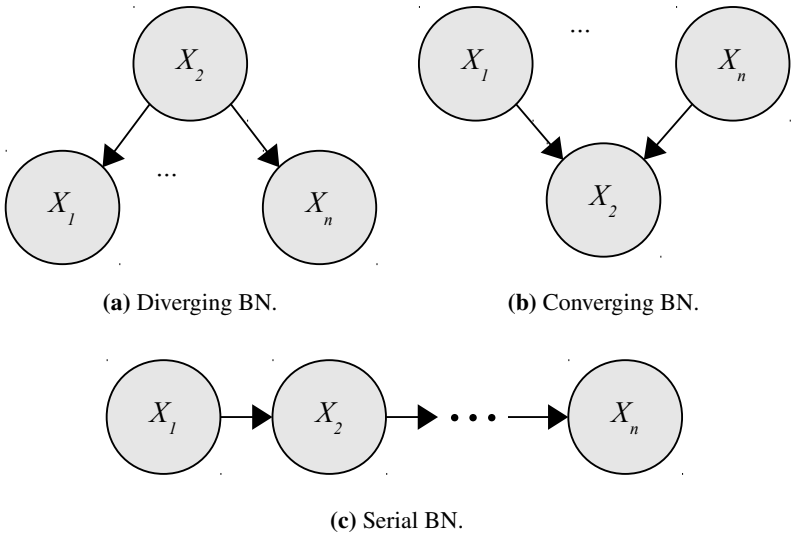
Consider the BN illustrated in Figure 2.7. In this BN, variable $B$ is statistically independent of variable $A$. The joint distribution is given as

$$p(A,B,C) = p(A|C)p(B|C)p(C). \tag{2.49}$$

A practical example of the BN in Figure 2.7 is a reasoning on a disease and its symptoms. The variable $C$ represents the disease. Variables $A$ and $B$ represent symptoms of the disease. The model could be analysed in two different ways depending on what information is given. The probability of exhibiting symptoms $A$ and $B$ may be inferred for a given disease $C$. The probability of having a disease $C$ may be inferred if the symptoms $A$ and $B$ are given.

Three basic structures in BNs exist [16, 14]; the diverging model, the converging model and the serial model. These models are illustrated in Figure 2.8a, Figure 2.8b and Figure 2.8c respectively. The divergent BN describes a model representing a problem with a 'common cause'. The convergent BN describes a model representing a problem with a 'common effect'. The serial BN describes a model representing a series of causal influences. A BN will consist of a combination of some or all of the three basic structures.

$D$-separation or Directed-graph-separation is a set of rules that may be used to determine conditional dependency between two variables in a BN by inspection of the graph. The definition of $D$-separation




**(a)** Diverging BN. **(b)** Converging BN.

**(c)** Serial BN.

**Figure 2.8:** The three basic structures found in the BN.

is provided as follows [16]:

> **Definition 2** *Two distinct variables A and B in a BN are D-separated if for all paths between A and B, there is an intermediate variable V (distinct from A and B) such that either*
>
> 1. *the connection is serial or diverging and V is instantiated (observed) or*
> 2. *the connection is converging and neither V nor any of V's descendants are observed.*

The first condition in Definition 2 refers to BNs described by Figure 2.8a and Figure 2.8c. For these cases, $X_1$ is independent of $X_3, ...X_n$ given $X_2$. The second condition in Definition 2 refers to BNs described by Figure 2.8b. For this case, $X_1$ is dependent on $X_3, ..., X_n$ given $X_2$ or any of the descendants of $X_2$. If $X_2$ or any of the descendants of $X_2$ are not observed then $X_1$ is independent of $X_3, ..., X_n$.

### 2.2.3 A Methodology for the Bayesian Network

The methodology for constructing and analysing a BN may be described as follows:

1. Define the random variables that are to be modelled.
2. Specify the dependency relationships between the defined random variables.

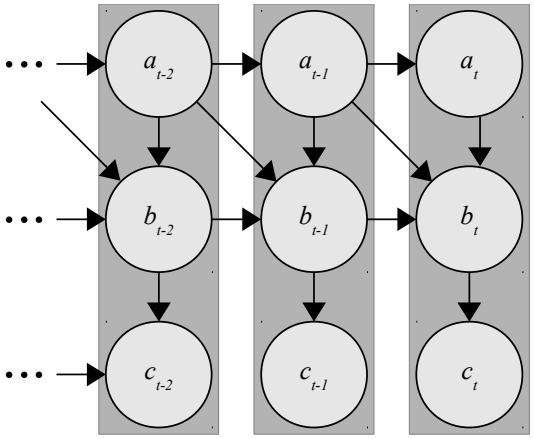



**Figure 2.9:** An example of a DBN with variables $a_t$, $b_t$ and $c_t$. The template BNs are constrained within the dark grey boxes.

3. Construct the BN graphical model. Each random variable is represented by a node. The links between the random variables are determined by the dependency between random variables
4. Specify the conditional probability distributions between variables and their parents.
5. Clamp observed random variables to their observed values.
6. Determine the probability distribution of particular random variables of interest using marginalisation.

### 2.2.4 The Dynamic Bayesian Network

The dynamic Bayesian network (DBN) is a temporal extension of the BN [14]. This provides the means to model a state of the world as it evolves over time. The general approach is to define a template BN for a particular time slice. The template BN consists of a set of $n$ random variables $x_t^1, ..., x_t^n$. The template is repeated at each discrete point in time. An example of a DBN with variables $a_t$, $b_t$ and $c_t$ is illustrated in Figure 2.9.

Random variables in a template BN at a particular time are conditioned on the random variables in a template at a previous time. For example, variable $a_t$ may be conditioned on variable $a_{t-1}$. The conditioning of $a_t$ on $a_{t-1}$ implies that there exists a link from $a_{t-1}$ to $a_t$. Owing to the nature of the DBN, this implies that each variable is conditioned on its predecessor. A series BN is thus formed from $a_0$ to $a_t$. This implies that $a_t$ is conditioned on all the variables $a_0$ to $a_{t-1}$ if none of them are observed. This property results in an increase in the complexity of the DBN as time increases. The Markov assumption is often applied to limit time dependency. In practical applications, the first order

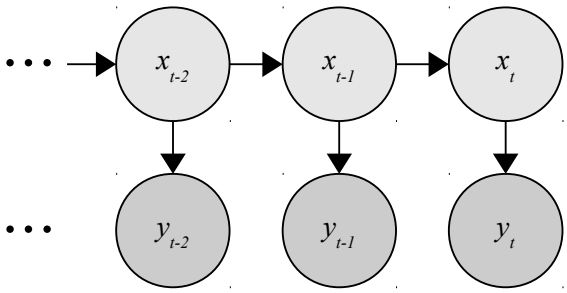



**Figure 2.10:** The hidden Markov model. The state variable $x_t$ is hidden and is thus unobservable. The observation variable $y_t$ is observable.

Markov assumption is often assumed [14].

An example of a DBN is the hidden Markov model (HMM). The HMM has been successfully applied to applications such as speech recognition [17]. The HMM is a Markov model with hidden and observable states. The HMM consists of a state variable $x_t$ and a observation variable $y_t$. The state variable $x_t$ is not observable and is thus considered hidden. The observation variable $y_t$ is emitted from the state variable $x_t$. The observation variable is observable. The structure of the HMM is illustrated in Figure 2.10.

### 2.2.5 Sampling from a Graphical Model

The *ancestral sampling* method is a method that may be used for sampling from a graphical model [18, 13]. Ancestral sampling is especially relevant to the BN and DBN. The ancestral sampling method begins at a root node $X_1$ in the graphical model. A sample $\hat{X}_1$ is drawn from $p(X_1)$. Paths from the root node(s) are followed down the graph. At each node in the graph, a sample $\hat{X}_i$ is drawn according to $p(X_i|pa(X_i))$, where $pa(X_i)$ is the set of parents of $X_i$. The set of samples $\hat{X}_1, ..., \hat{X}_n$ are considered synthetic data drawn from the distribution modelled by the graphical model.

## 2.3 SWITCHING LINEAR DYNAMIC SYSTEMS

### 2.3.1 The Linear Dynamic System

The linear dynamic system (LDS), also known as the dynamic state space model, is a continuous state system. The deterministic observed LDS is a system that contains a continuous state $v_t$ that changes

linearly over discrete time. The observed linear dynamical system is defined by the following linear equation [18]

$$h_t = \mathbf{A}h_{t-1} + \eta_h. \tag{2.50}$$

The variable $h_t$ is the state vector, $\mathbf{A}$ is the state transition matrix and $\eta_h$ describes Gaussian noise $\mathcal{N}(\eta_h|\mu,\Sigma)$ known as the plant noise. In this study, $\Sigma$ is considered to be time invariant and $\mu = 0$. The linear dynamical system is equivalent to a first order Markov model with transition probability given by [18]

$$p(h_t|h_{t-1}) = \frac{1}{|2\pi\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(h_t - \mathbf{A}h_{t-1})^T\Sigma^{-1}(h_t - \mathbf{A}h_{t-1})\right\}. \tag{2.51}$$

The latent LDS (or LLDS) includes a latent or hidden vector $h_t$. The latent vector describes the linear dynamics of the system. The observation vector $v_t$ is a linear function of the latent vector $h_t$. The linear function that emits the observation vector may model a sensor that observes the system. The latent LDS models an unknown linear system that is only observable through the observation vector. The latent linear dynamical system is defined by the following two linear equations [18]

$$h_t = \mathbf{A}h_{t-1} + \eta_h, \qquad \eta_h \sim \mathcal{N}(\eta_h|\bar{h}_t,\Sigma_h), \tag{2.52}$$

$$v_t = \mathbf{B}h_t + \eta_v, \qquad \eta_v \sim \mathcal{N}(\eta_v|\bar{v}_t,\Sigma_v). \tag{2.53}$$

The matrix $\mathbf{A}$ is the transition matrix and $\mathbf{B}$ is the emission matrix. The variables $\bar{h}_t$ and $\bar{v}_t$ describe the hidden and output biases respectively. The transition and emission probabilities are given by the following two equations [18, 19]

$$p(h_t|h_{t-1}) = \frac{1}{|2\pi\Sigma_h|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(h_t - \mathbf{A}h_{t-1})^T\Sigma_h^{-1}(h_t - \mathbf{A}h_{t-1})\right\}, \tag{2.54}$$

$$p(v_t|h_t) = \frac{1}{|2\pi\Sigma_v|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(v_t - \mathbf{B}h_t)^T\Sigma_v^{-1}(v_t - \mathbf{B}h_t)\right\}. \tag{2.55}$$

The LDS is a first order Markov model with a joint probability distribution given by [18]

$$p(h_{0:T}, v_{0:T}) = p(h_0)p(v_0|h_0)\prod_{t=1}^{T} p(h_t|h_{t-1})p(v_t|h_t). \tag{2.56}$$

The DBN model of the LDS is illustrated in Figure 2.11. The structure is identical to the HMM structure illustrated in Figure 2.10. The HMM contains nodes with discrete variables whereas the LDS contains nodes with continuous variables.

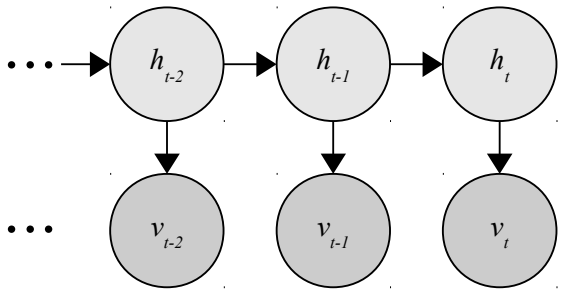



**Figure 2.11:** The linear dynamic system model. The state variable $h_t$ is hidden and is thus unobservable. The observation variable $v_t$ is observable.

The LDS has been successfully applied to various applications. The Kalman filter has been successfully applied for tracking targets in space. The Kalman filter is a method that infers the LDS parameters [14]. The inference of the parameters provides a means to predict the next state in time of the system. Predicting the next state in time provides the means to track the system through time.

### 2.3.2 Filtering in the LDS and HMM

Filtering in the LDS is the operation by which the hidden state variable $h_t$ is inferred at time $t$ given the current and previous measurements $v_{1:t}$ [20]. That is, filtering is the task of computing $p(h_t|v_{1:t})$. The on-line filtering algorithm is a recursive algorithm that computes the filtered result at the current time $t$ using the model parameters and the filtered result at the previous time step $t-1$. To represent the filtered result in a recursive form, the following five steps are taken [20, 18]:

1. Marginalise over the previous hidden variable:

$$p(h_t|v_{1:t}) = \int_{h_{t-1}} p(h_t, h_{t-1}|v_{1:t}).$$

2. Divide up the evidence:

$$p(h_t|v_{1:t}) = \int_{h_{t-1}} p(h_t, h_{t-1}|v_t, v_{1:t-1}).$$

3. Apply Bayes rule:

$$p(h_t|v_{1:t}) = \int_{h_{t-1}} \zeta p(v_t|h_t, h_{t-1}v_{1:t-1}) p(h_t, h_{t-1}|v_{1:t-1}).$$

4. Apply the chain rule of probability:

$$p(h_t|v_{1:t}) \propto \int_{h_{t-1}} p(v_t|h_t, h_{t-1}v_{1:t-1})p(h_t|h_{t-1}, v_{1:t-1})p(h_{t-1}|v_{1:t-1}).$$

5. Apply the Markov property:

$$p(h_t|v_{1:t}) \propto \int_{h_{t-1}} p(v_t|h_t)p(h_t|h_{t-1})p(h_{t-1}|v_{1:t-1}).$$

In step 3, $\zeta$ is a normalisation constant. The result in step 5 describes the filtering result in the form of a recursion. The first factor is the emission probability. The second factor is the transition property. The third factor is the filtering result for the previous time step at time $t-1$. In the discrete case of the HMM, the integral $\int_{h_{t-1}}$ is a summation $\sum_{h_{t-1}}$ over $h_{t-1}$. In larger networks, $D$-separation may applied in step 5 rather than the Markov property. The Markov property is a special case of $D$-separation.

### 2.3.3 The Switching Linear Dynamic System

The LDS provides a means to model kinematic motion. The LDS is not sufficient to model complex kinematic behaviour. Different behavioural activities will have different motion models. The switching linear dynamic system (SLDS) provides the means to include various linear dynamic system models in a single DBN. The SLDS extends the LDS to include a discrete switching processes state variable $s$. This variable is a variable that switches between linear dynamic models. The ability to switch between linear dynamic models provides the means to model kinematic behaviour.

The SLDS is defined by the following equations [18]

$$h_t = \mathbf{A}(s_t)h_{t-1} + \eta_h(s_t), \qquad \eta_h(s_t) \sim \mathcal{N}(\eta_h(s_t)|\bar{h}(s_t), \Sigma_h(s_t)), \tag{2.57}$$

$$v_t = \mathbf{B}(s_t)h_t + \eta_v(s_t), \qquad \eta_v(s_t) \sim \mathcal{N}(\eta_v(s_t)|\bar{v}(s_t), \Sigma_v(s_t)). \tag{2.58}$$

The transition and emission probabilities are given by the following two equations [18]

$$p(h_t|h_{t-1}, s_t) = \mathcal{N}(h_t|\mathbf{A}(s_t)h_{t-1} + \bar{h}(s_t), \Sigma_h(s_t)). \tag{2.59}$$

$$p(v_t|h_t, s_t) = \mathcal{N}(v_t|\mathbf{B}(s_t)h_t + \bar{v}(s_t), \Sigma_v(s_t)). \tag{2.60}$$

The SLDS is a first order Markov model with a joint probability distribution given by [18]

$$p(h_{0:T}, v_{0:T}, s_{0:T}) = \prod_{t=1}^{T} p(s_t|h_{t-1}, s_{t-1})p(h_t|h_{t-1}, s_t)p(v_t|h_t, s_t) \tag{2.61}$$

The initial distributions at time $t = 0$ , $p(s_0)$, $p(h_0|s_0)$ and $p(v_0|h_0, s_0)$ are Gaussian prior distributions. The DBN model of the SLDS is illustrated in Figure 2.12.

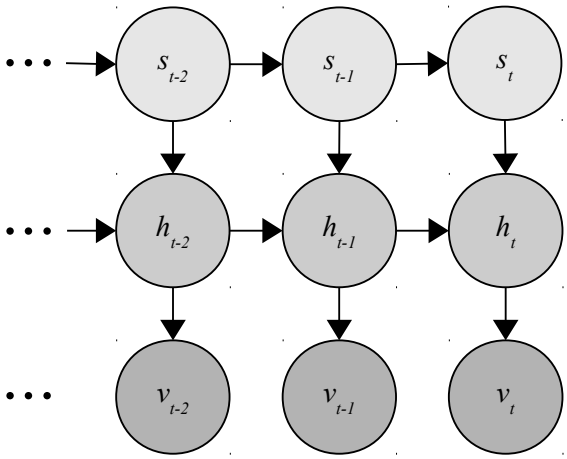



**Figure 2.12:** The switching linear dynamic system model. The variable $s_t$ is the switching state, $h_t$ is the hidden vector and $v_t$ is the observation vector.

## 2.4 CLOSING REMARKS

Theoretical foundations for probability theory, graphical models and the SLDS are set in Chapter 2. In following chapters, a DBN behavioural model is constructed upon this foundation. This DBN is applied for data generation and for classification.

# CHAPTER 3

# LITERATURE STUDY

A SURVEY OF THE LITERATURE is presented in this chapter. The objective of the survey is to provide context of this study with regards to the literature. The survey is focused on maritime piracy and applications involving the DBN.

## 3.1 MARITIME PIRACY

### 3.1.1 Maritime Piracy Definition and Description

**Definition 3** *The definition of piracy consists of any of the following acts*

1. *any illegal acts of violence or detention, or any act of depredation, committed for private ends by the crew or the passengers of a private ship or a private aircraft, and directed-*
   (a) *on the high seas, against another ship or aircraft, or against persons or property on board such ship or aircraft;*
   (b) *against a ship, aircraft, persons or property in a place outside the jurisdiction of any State;*
2. *any act of voluntary participation in the operation of a ship or of an aircraft with knowledge of facts making it a pirate ship or aircraft;*
3. *any act of inciting or of intentionally facilitating an act described in subparagraph (a) or (b).*

This definition is accepted by the International Chamber of Commerce's International Maritime Bureau (ICC-IMB) [2] and the International Maritime Organization (IMO) [21]

Pirate attacks generally occur in one of three categories [22]

1. petty theft,
2. planned Robberies or
3. permanent ship and cargo hijackings.

Permanent ship and cargo hijacking is considered to be the most serious offence in maritime piracy. Pirates hijack vessels and kidnap the crew. The hijacked ship and crew are taken to a pirate village. The vessel and crew are ransomed for between five-hundred-thousand and one million dollars [23]. This form of piracy has significant consequences in economic, humanitarian and environmental aspects. Economically, the cost is high due to factors such as loss of cargo, property, ransom costs, increased insurance premiums and re-routing costs [1]. Humanitarian consequences include health, psychological effects, physical injury and death of seafarers [24, 1]. Piracy hijackings have the potential to cause significant environmental and pollution damage. Oil or chemical leaks may be caused by pirates firing on a tanker or by a collision of an unsupervised drifting vessel [25]. A vessel may be left unsupervised if it is abandoned by pirates or under attack by pirates.

### 3.1.2 A Brief History and Current Trends

Maritime piracy has been a concern for many centuries. It is believed that Julius Caesar was ransomed by pirates in the year 78 BC [26]. The 17th and 18th centuries are considered to be the prime era for maritime piracy. Britain's first piracy act was incorporated in 1698. Further piracy acts were incorporated in 1721, 1837 and 1850 [27]. Whereas over the nineteenth and twentieth centuries, maritime piracy decreased, after the cold war it began to increase [25, 28].

A significant portion of maritime pirate activity is found off the shores of Somalia, in Gulf of Aden and in the Red Sea. This region is exclusively considered in this study. Somalia has been in a state of unrest since the civil war which lead to the collapse of the Siad Barre administration in 1991 [29]. The political unrest has left the country with economic, political and social problems. It is out of these problems that maritime piracy has flourished in this region. The lack of governance and authority in the country has left little or no negotiable traction for the international community. Warships from countries such the U.S, the U.K, Canada, Turkey, Germany, Denmark, France and Russia have been deployed to patrol the region [30]. In late 2008 the three counter-piracy missions were deployed; The EU's Operation 'Atlanta', NATO's Operation 'Ocean Shield' and the US-led Combined Task

Force-151 [5]. Signs of improvement have been noted in the Somali region; however piracy is on the increase in West Africa.

The IMB constructed a Piracy Reporting Centre (PRC) in October 1992 [2]. The services provided by this centre are as follows:

- Daily report status on piracy issued to ships via broadcasts
- Report piracy incidents to law enforcement and the International Maritime Organization (IMO)
- Help local law enforcement to apprehend and prosecute pirates
- Assist ship-owners and crew members whose vessels have been attacked
- Provide piracy activity updates via the internet
- Publish quarterly and annual reports detailing piracy statistics

The reports provided by the IMB are based on incidents reported by victims. Many ship-owners are reluctant to report piracy incidents. This is due to the cost caused by increased insurance premiums and delays caused by incident investigations. The number of reports provided by the community is thus considered to be a fraction of the actual number of incidents that occurred [31].

The 2011 annual report provided by the IMB [2] indicates that 439 reported incidents of piracy and armed robbery occurred in 2011. A total of 263 attacks occurred in 2007. A significant increase in the number of attacks occurred between 2007 and 2011. A global map of reported locations of the attacks of 2011 is provided in Figure 3.1. Most of the attacks occur in the Somali and the South-East Asian regions. Of the 439 incidents, 105 incidents were attempted attacks, pirates boarded the ships in 176 of the incidents, ships were fired upon in 113 of the incidents and ships were hijacked in 45 of the incidents. The attack locations in the 2011 IMB annual report are used as the primary dataset in this study.

## 3.2 BACKGROUND AND RELATED WORK

### 3.2.1 The DBN and the SLDS

The Bayesian network (BN) [32] is a probabilistic model that models causal relationships between random variables. The dynamic Bayesian network (DBN) [33] is the temporal extension of the BN. The DBN provides a means for modelling sequential data [33, 18, 34, 14, 16]. The DBN has found application in a wide variety of fields. These include computer vision based human motion analysis

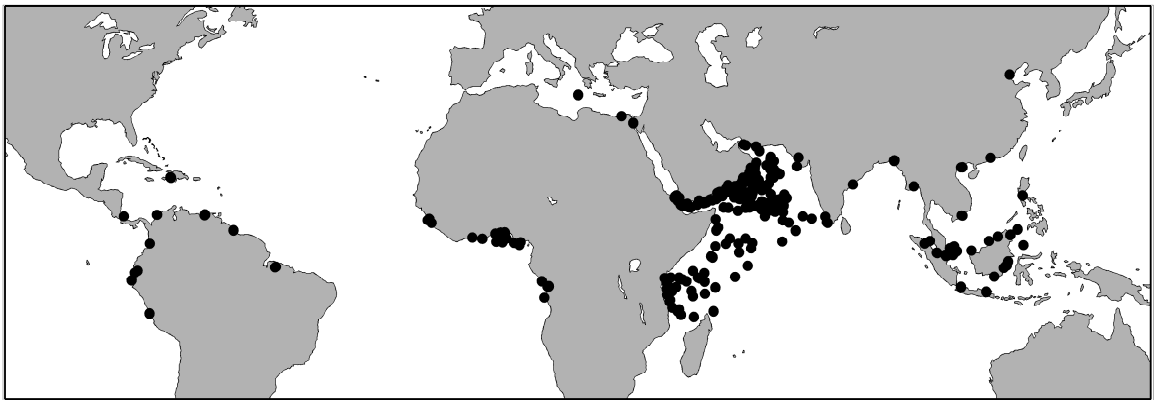


**Figure 3.1:** Global map with markers indicating reported pirate attack locations in 2011 [2].

[35], learning from observation [36] , vehicle detection and tracking [37], fault detection in dynamic systems [38] and situation awareness [39]. The application of the DBN for classifying maritime vessels has not been found in literature. The influence diagram (ID) is considered as a BN supplemented with utility functions and decision variables [40]. The higher level nodes in the generative model proposed in this study may be argued as a form of an ID. Poropudas and Virtanen have applied IDs for simulation data analysis [41]. This work has extended by applying DBNs in the application of simulation [42, 43]. In these applications, what-if-analysis is performed and time evolution is studied. The simulation approach of the DBN is used in the applications of simulated air combat and server queuing. The application of expert knowledge to assemble the DBN is suggested as an extension to their work. The structure of the generative behavioural model proposed in this study is informed by expert knowledge.

The linear dynamic system (LDS), also known as the dynamic state space model, may be considered as a continuous variable DBN [44]. The switching linear dynamic system (SLDS) [45, 34, 18] is formed by including a discrete switching state variable to the LDS. The SLDS is a hybrid DBN that consists of both discrete and continuous variables. The switching state variable describes various kinematic states of a modelled system. The SLDS has various alternative names such as the switching state space model and the switching Kalman filter [34]. The SLDS has been successfully utilised in various applications that include econometrics [46], human motion modelling in computer vision [47] and speech recognition [48, 19]. The SLDS is generally applied to model piecewise linear dynamical systems. The SLDS is intended as a modelling framework and does not provide a direct means for classification. The DBN proposed in this study consists of a SLDS that has been extended to model various forms of kinematic behaviour of several maritime vessel classes. The classifier DBN proposed

in this study includes variables that provide the means to perform classification.

A variety of methods for inference have been proposed for the LDS. The Kalman filter [49] is a particularly well known method for performing inference in the SDLS. In terms of the graphical model structure, the LDS is identical to the hidden Markov model (HMM). The forward-backward algorithm is a notable method for inference in the HMM. Inference in the SLDS is NP-hard [34, 18]. This is discussed in section 6.2. Methods such as the Kalman filter and the forward-backward algorithms cannot be directly applied to the SDLS. Various approximate inference methods for the SLDS have been proposed.

A survey on inference algorithms for the SLDS is provided in [19]. Inference methods include the assumed density filtering (ADF) algorithm [50, 51, 52], the approximate variational inference algorithm [53, 54], the approximate Viterbi inference algorithm [47], the generalised pseudo Bayesian (GPB) approximation algorithm [45] and the Gaussian sum filtering (GSF) algorithm [55, 18]. The GPB and GSF algorithms may be considered to fall within the category of 'merging Gaussians' algorithms. The 'merging Gaussians' algorithm is applied when the probability distribution increases in complexity over time. The probability distribution is represented using a mixture of Gaussians. At each time iteration, the probability distribution increases in complexity by an increase in the number of mixture components. The 'merging Gaussian' algorithm manages this increase in complexity by collapsing the Gaussian mixture into a smaller mixture with a specified number of Gaussians [54]. The 'merging Gaussians' algorithm is similar to the particle filter algorithm [18]. The Gaussian mixture components relate to the particles in the particle filter algorithm. The collapsing of the Gaussian mixture relates to the re-sampling step in the particle filter algorithm.

Nonparametric algorithms exist for performing inference in the SLDS. The Rao-Blackwellised particle filter is proposed for performing filtering in Dynamic Bayesian Networks [56]. The Sigma-Point Kalman Filter is applied for inference in the LDS [57]. Deterministic Annealing [54] and the Markov chain Monte Carlo (MCMC) methodology [19, 58] have been applied for inference in the SLDS. The analysis of such methods is reserved for future research for the development of a smoothing algorithm for the proposed classifier DBN model.

### 3.2.2 Related Fields and Applications

Multi-agent systems have found application in various fields. These fields include social sciences, robotics, computer games, simulation, military and econometrics [59, 60]. Multi-agent based simulation (MABS) is a framework that is relatively new. The MABS framework is generally applied to simulate and model entities in a particular environment [61]. In general, agents are considered to be autonomous and independent. Agents may have the ability to interact with other agents as well as their environment [62, 61]. The MABS application in this study is closely related to the approach of military applications. Military applications intend to support decision making and enhance training [63]. A survey of MABS applications in the military is provided in [64]. No application of the DBN to the problem of MABS has been found in literature.

The work provided in this study is considered to fall within the framework of the information fusion or data fusion field. Information fusion is a field that combines data from various sources to achieve inferences that cannot be obtained from a single data source [65, 40]. This study is concerned with maritime surveillance and DBN applications. Information fusion techniques have been applied in various maritime surveillance applications. A coastal surveillance simulation test-bed has been developed [66]. The simulation test-bed is used for the study of dynamic resource and network configuration management, distributed fusion, and self-synchronising units and agents. Various BNs have been applied in information fusion for maritime domain awareness [67, 68] and maritime security [69, 70, 71, 72]. The BN is generally applied for inference and decision making. A DBN is proposed for multi-sensor information fusion applications [73]. It is proposed that data from various sensors such as imaging sensors, radar sensors and acoustic sensors be fused using the DBN. The DBN has been applied for data fusion in a driver fatigue recognition application [74]. A discrete variable DBN is proposed for the purpose of inferring a level of driver fatigue. The model is application-specific and only consists of discrete variables. A human-computer interface application using the DBN has been proposed [75]. The study is a PhD thesis that considers various forms of DBNs including the hybrid DBN. The application proposed in the study is a hybrid DBN that performs hand gesture recognition. The DBN that is proposed is similar to the SLDS. This DBN is also expanded for visual tracking through images. The model that is proposed exclusively addresses the gesture recognition application.

The behavioural models presented in this study are statistical models that model on various behavioural states. The state transition data fusion (STDF) model is a general model that may be applied

in information fusion systems [76, 77]. A STDF model is proposed to provide unification between sensor and higher level data fusion [78]. The STDF model is applied for maritime domain awareness. A set of states of the world are defined. As time progresses, the world is understood as a set of transitions between states. The proposed STDF model may be applied at levels zero to three of the JDL data fusion model. The model does apply probability theory, however not in the form of a DBN.

In this study, behaviour is significantly influenced by contextual elements such as the environment. Context modelling applications model and incorporate contextual information. A survey on context modelling has been conducted in [79]. A context modelling and reasoning in pervasive computing has been conducted by Bettini et al. [80]. Context-based information fusion has been applied in various fields. Applications include video indexing [81], natural language processing [82] and computer vision [83]. Context-based information fusion methods have been applied to various maritime threat and situation assessment applications [84, 85, 86]. The DBN is not used in any of these applications. A discrete DBN has been proposed for context based information fusion in a location estimation application [87]. The DBN is applied to fuse information from various sources. The context attributes are represented using fuzzy logic. The application has been extended for imprecise contextual reasoning using fuzzy logic [88, 89]. The DBN is particularly applied for sensor information fusion. Modelling of the tracked entities is not included in the DBN.

Several methods exist for classifying maritime vessels. A simple method is to identify a vessel class given Automatic Identification System (AIS) data transmitted by the vessel [90]. The AIS is discussed in more detail in Section 4.2. Smaller vessels are not required to transmit AIS data and vessels have been known to disable their AIS. Radar, infrared and optical imagery are data sources that are commonly used for classifying maritime vessels. Radar imagery obtained using synthetic aperture radar (SAR) or inverse synthetic aperture radar (ISAR) methods are commonly used [91]. Particularly for SAR, a large body of literature exists [92, 93, 94]. Low resolution radar has also been considered for classification [95]. Various approaches of vessel classification using infrared imagery have been published [96, 97, 98]. Vessel classification using land based and ship based optical imagery in harbours has been proposed [99, 100]. High resolution satellite imagery has also been successfully applied to vessel classification [101]. The satellite optical imagery provides a higher level of detail than that of radar imagery. The optical imagery however is more susceptible to weather conditions and has lower region coverage than that of radar imagery [101, 93]. Infrared imagery is applicable during night time, whereas optical imagery is not. Vessel classification methods based on imagery

are not based on behaviour but rather physical features of the vessels. Furthermore, these methods are often not suitable for vessel sizes under $10 - 15$ meters [93]. The classification method presented in this study does not require any physical features of the vessel and is not limited by vessel size. The only information required is a tracked trajectory of the vessel.

Several website based applications for situation awareness are available on-line. An on-line maritime piracy risk assessment and data visualisation tool is available [102]. The Blue Hub for maritime surveillance data gathering has been developed by the European Commission [103]. A maritime piracy awareness platform is currently being developed for the Blue Hub.

### 3.2.3 Related Maritime Piracy Applications

Various counter-piracy applications have been proposed in literature. Game theory has been applied for the optimisation of counter piracy strategies. A game theoretic application has been proposed to suggest routes that avoid pirate vessels [104, 105]. A game theory approach for counter piracy patrolling strategy optimisation has been proposed [106]. Risk analysis applications have been proposed to assist ship captains in risk management during a pirate attack [107, 108]. These applications do not attempt to identify or detect maritime pirates.

### 3.2.4 Maritime Piracy Simulation and Modelling

A multi-agent state-based simulation environment has been developed for simulating behaviour of maritime entities [7]. Long-haul shipping, patrolling behaviour and piracy behaviour are simulated by the system. Vessel behaviour is simulated using finite state machines. Long-haul shipping behaviour is defined such that ships sail along routes that minimise cost, travel time and security. Pirate behavioural activities include discovering, approaching and attacking vessels. Patrol vessels are located according to a risk map and their deterrence potential. A set of routes that maximizes pirate deterrence is determined for patrol vessels. The simulation method is a heuristic method and does not consider the DBN.

A method for simulating pirate prowling behaviour has been proposed [109]. The model is defined such that pirates motor out from a home base in skiffs. The skiffs venture out to a predetermined location where they can wait for a target vessel. The skiffs drift in the predetermined region until food and water supplies are depleted. The pirates return to their home base to refresh their supplies.

The drift trajectory of the pirate skiffs is simulated using oceanographic and meteorological forecasts. The method is limited to the modelling of the kinematic motion of pirate vessels. Other contextual and behavioural information is not modelled.

### 3.2.5 Maritime Piracy Detection and Identification

Maritime piracy detection involves detecting and identifying pirates in the maritime domain. Counter piracy rely heavily upon naval warship patrols in high risk regions. Institutions such as the IMB and IMO have proposed various procedures for vessels in the case of a pirate attack. A proposed ship security plan is as follows [22]:

1. Detect threats as early as possible
2. Deny or restrict access to part or all of the ship
3. Communicate to others the nature of a threat
4. Protect the crew and passengers
5. Protect cargo
6. Outline crew actions in the event of hostile boarding

Early detection of possible threats is often the most effective deterrent [21]. Early detection allows for the sending distress signals such as flares and radio communications. This may allow for assistance from a nearby patrol. Early detection allows for measures to be taken such as evasive manoeuvres and on-board security plans.

An approach of localization and detection of pirates through monitoring satellite communications is proposed in [110]. Satellite communications require GPS information for beam spotting. This information is however not easily obtainable from the service provider. The alternative method of radio spectrum monitoring is proposed for localization. The Thuraya satellite telephone operates in the 1626.5-1660.5MHz frequency band with a maximum output power of 2W. The proposed method of detection is to utilise a directional parabolic antenna for detection of transmissions. The location of the transmission can be inferred by the directionality of the antenna and the dB drop of the signal. If two antennas are utilised, the location could be triangulated.

Pirate detection through classification of small craft using infra-red (IR) imagery has been proposed [96]. It is proposed that an onshore IR camera be placed in an optimal location for surveillance. The

surveillance footage is then monitored by a proposed classification system that detects and classifies small craft. The system is intended for a harbour environment.

A computer vision method of identifying and tracking vessels, such as pirate vessels, is presented in [111]. A background subtraction method is used for identifying the foreground in an image. The outline of moving vessels is demarcated using a real-time approximation of level-set-based curve evolution method. The proposed method is able to detect and track small targets.

Fourier space analysis has been proposed for identifying small craft in imagery [112]. The application is suggested for the use of detecting small pirate vessels. Frequency domain techniques are used to characterise sea state. The characterisation of the sea state allows for the reduction of apparent motion in the image due to the sea.

The three applications discussed above constitute all of documented pirate detection methods that could be identified at the time of investigation. The quantity and the variety of the methods is limited. This indicates that there is a deficiency of research in this area within the literature. This study is intended to provide a contribution to the literature in this deficient area. The model proposed in this study provides the means to fuse contextual information from various sources. Kinematic behaviour is directly modelled in the proposed DBN. This provides a more complete and robust methodology for identifying maritime pirates.

## 3.3 CLOSING REMARKS

Through a literature survey, an opportunity for contributing to the literature is identified. The opportunity involves the development of methods for modelling and classification of vessels in a maritime piracy environment. Such methods are proposed and discussed in following chapters.

# CHAPTER 4

# PRIOR KNOWLEDGE

PRIOR KNOWLEDGE is required to formulate the behavioural models for this study. Knowledge of pirate behaviour and tactics is used to inform the structure of the DBN that is proposed in this study. In application of the models, data sources and sensors are required to monitor vessels. Behaviour may be extracted from the data provided by data sources and sensors.

## 4.1 PIRACY BEHAVIOUR AND TACTICS

Pirate activity is considered to be opportunistically in nature. Pirates generally select high traffic regions and wait for a suitable target. Many attacks occur within sight of land during the night [22]. Pirates have however been reported to venture over 200 nautical miles offshore [113]. The trend for pirates is to use a fishing vessel as a mobile base or mothership. Small high speed craft such as skiffs are towed by the mothership. When a target is identified, one or more skiffs are deployed to engage in a swift attack on the target vessel. Each skiff will generally hold between five and ten pirates. Ships are boarded using grappling hooks or ladders. It is approximated that the time between pirate sighting and being boarded by pirates is fifteen minutes [1], [23]. The crew is taken hostage once the pirates have boarded the ship. The ship is sailed to a local pirate village from which ransom negotiations are commenced [114].

### 4.1.1 Pirate Appearance

Pirates generally capture or hijack a medium sized fishing vessel such as dhows, sailing yachts and trawlers. These vessels function as motherships for the pirates [23]. Skiffs are often towed and dispatched from the motherships in an attack. The skiffs are generally in poor condition, made of wood

or fibreglass and are powered by an outboard motor [115]. Pirates and pirate vessels are disguised as fishing vessels. This disguise causes difficulty in detection [114].

### 4.1.2 Pirate Kinematic Characteristics

Various models have been proposed for the kinematic behaviour of pirates. Models are generally based on the opportunistic nature of the pirates. A proposed model indicates that the pirates venture out from a mothership or land base in skiffs [109]. The pirates wait for a potential target in the skiffs. The skiff is left to drift with the ocean currents and winds while the pirates wait. Pirates return to their base to refresh depleted supplies such as food and water. Reports have indicated that pirates also search for targets from their motherships. Once a target has been identified, the mothership is abandoned and skiffs are deployed for an attack [1]. Motherships are ransomed or abandoned if the supplies are depleted or if the ship is suspected by naval forces [116]. There is evidence that suggests some pirates use Automatic Identification System (AIS) data transmitted from vessels for target selection [30].

### 4.1.3 Communication Methods and Technology

Reports have indicated that pirates utilise mobile communications to communicate with ground based pirates [110]. Standard cellular devices can only be used near the coast. The maximum range of the cellular devices is approximately twenty to thirty kilometres from land. Alternatively, HF/VHF communications are able to provide superior range at sea. HF/VHF communication is monitored by authorities and criminals are generally weary of using it. Satellite communication is the preferred medium of communication due to its wide spatial range and lack of monitoring by authorities [110], [1]. Other forms of technology utilised by pirates include GPS devices and advanced money counting machines [23].

### 4.1.4 Weapons

Maritime pirates possess weapons for the purpose of attacking targets, holding target crews and defending themselves against naval patrols. Reported weapons include knives, pistols, machine guns, explosive devices, high calibre rifles, Strella surface to Air Missiles (SAMs), Anti-ship Mines, hand-held mortars and rocket propelled grenades (RPGs) [25], [114].

### 4.1.5 Pirate Offensive Tactics

As discussed previously, maritime pirates generally attack using high speed craft such as skiffs [22]. Pirates carry weapons, ladders and grappling hooks for boarding target vessels. Once the target has been identified, a speedy approach on the target is made by one or more skiffs. Targets may be fired upon for the purpose of slowing the target down. Pirates attempt to board moving vessels alongside the stern. Vessels at anchor are generally boarded from the bow or the stern. Reports have often noted a dhow or fishing vessel to be in the vicinity of the attack [2]. These dhows and fishing vessels are most likely pirate motherships.

### 4.1.6 Target Characteristics

Prime targets for pirates are lightly manned, slow moving ships with easily accessible freeboards [23], [22]. High traffic regions are preferred. Pirates do not seem to make any distinction between different flags [117]. Cruise ships are generally not targeted due to high range acoustic crowd control devices on board. These devices have a range of 0.4 to 16km [6]. Of the 439 reported incidents in 2011, 100 bulk carriers, 71 chemical tankers, 62 container ships and 61 general cargo ships were targeted [2].

### 4.1.7 Meteorological and Oceanographic Influences

The small and medium sized pirate vessels are susceptible to weather and oceanographic conditions. Pirates are therefore expected to avoid seasonal conditions such as high waves, high winds, high ocean currents and monsoon seasons [116]. Droughts may indirectly drive piracy through poverty [113]. The ICC-IMB 2011 annual report provides a monthly comparison of incidents [2]. The highest frequency of pirate attacks is over the period of January to May. The highest number of attacks occurs in January. Attacks decrease over the monsoon season of June to September. The lowest number of attacks occurs over the months of November and December. A histogram of piracy incidents over the calendar months of 2011 is provided in Figure 4.1. Pirates generally select specific geographic regions where oceanographic and meteorological conditions are favourable. The U.S. naval oceanographic office has developed the Piracy Performance Surface [109]. This product forecasts wind and seas for the purpose of mapping locations that are favourable for pirate activity.

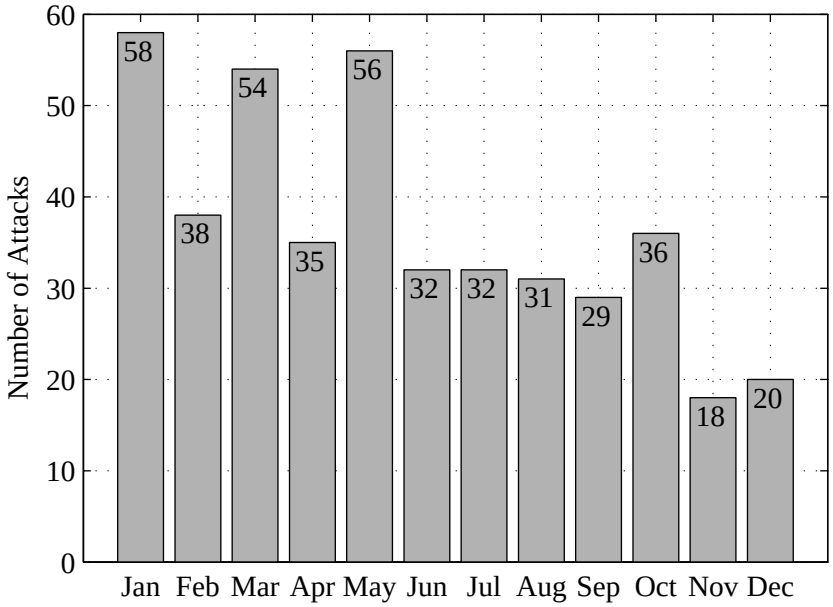



**Figure 4.1:** A histogram describing the number of attacks that occurred in each month of 2011 [2].

## 4.2 MARITIME DATA SOURCES AND ENTITY BEHAVIOUR

Maritime domain awareness systems require sensors for monitoring the domain. Such sensors include Radar systems (coastal and air), patrolling, self-reporting, visible and infrared imaging, video, Radar imagery, satellite communications and LF/MF/HF/VHF communications [110], [118], [119]. Behavioural attributes of maritime entities may be extracted from the data provided by these sensors.

A set of various sensor types may be used to monitor a particular domain. Data fusion methods integrate, fuse and filter data between sensors. The data fusion methods are required to handle unsynchronised data and missing data.

The Automatic Identification System (AIS) system is a self-reporting system. As of 2005, the International Maritime Organization (IMO) specifies that all ships greater than three hundred tons are required to transmit AIS data [120, 90]. The AIS was originally intended as a collision detection system and thus the range of the system is limited. AIS messages are automatically transmitted with a frequency proportional to the vessel speed. The messages contain information such as the Maritime Mobile Service Identity (MMSI), location (longitude, latitude), ground speed, rate of turn, vessel name, vessel type, vessel dimensions, cargo type and destination [121]. AIS data can be incorrectly

configured or disabled [122]. Criminals engaged in illegal activity are likely to disable the AIS on their vessels. Maritime pirates may use AIS data for target selection. Vessels smaller than three hundred tons may disable the AIS to avoid pirates.

Behavioural attributes of entities may be extracted from sensor data. Behaviour may be extracted from atomic events and long term behaviour [123]. Atomic level information may describe the current state of the vessel with respect to its environment. Long term behaviour considers sequences of events and activities. Behavioural attributes can be extracted from sources such as kinematic and communication information. Various factors that affect behaviour such as geographical, oceanographic, meteorological, seasonal and political factors may be considered for behaviour modelling.

Vessel location and tracks provide information of kinematic behaviour. Vessels generally travel along the most economical route [124]. The most economical route is the shortest route that incurs the lowest cost. Routes are constrained by factors such as landmass, water depth, traffic separation schemes, weather and exclusion zones. A vessel's journey will generally begin and end at a port. Behavioural models may be constructed based on such constraints.

Vessel communication provides behavioural information on vessel activities. Self reporting systems such as AIS may be monitored. Reporting should be according to the specification of the self reporting system. A model of normal activity may be constructed according to the specification of the self reporting system [124]. For example, vessels that report incorrect information or vessels not reporting information may be considered as behavioural anomalies. Evidence indicates that criminals who participate in activities such as piracy and illegal immigration utilise satellite communication [110]. Information of a communication such as the time of day, season, location, duration and frequency may be considered for behavioural analysis.

Various situational entities and forms of behaviour are abstract and latent [118]. Intent is an example of an entity that is latent. That is, intent cannot be directly observed. Abstract latent entities and behaviour variables may only be inferred.

## 4.3 CLOSING REMARKS

Prior behavioural knowledge is acquired in this chapter. In following chapters, this prior knowledge is used to inform the structure and parameters of behavioural models that are developed.

# CHAPTER 5

# GENERATIVE MODEL

A GENERATIVE MODEL models the distribution of the inputs as well as the outputs of a system [13]. Graphical models may be applied as generative models [13]. By sampling from a generative model, synthetic data may be generated.

In this chapter, a generative model is formed to generate synthetic maritime data in a piracy situation. A set of vessels of various classes are initialised on a map. The kinetic behaviour of each vessel is modelled by the generative model. The model generates a set of synthetic tracked coordinates through discrete time for each vessel. Transport vessels, fishing vessels and pirate vessels are modelled. Behaviour such as sailing, target acquisition, and attacking are modelled. The structure of the DBN is informed by prior knowledge of the problem. The DBN is constructed as an extension of the SLDS. Data is generated by applying ancestral sampling to sample data from the DBN.

## 5.1 SWITCHING LINEAR DYNAMIC SYSTEM ADAPTION

The classic SLDS is adapted for this study. A control function, a control vector and a set of sensors are appended to the SLDS model. The adaption of the classical SLDS model that is proposed is described by the following state space equations

$$h_t = \mathbf{A}(s_t)h_{t-1} + \mathbf{C}(s_t)u_t + \eta_h(s_t), \qquad \eta_h(s_t) \sim \mathcal{N}(\eta_h(s_t)|\bar{h}(s_t), \Sigma_h(s_t)), \tag{5.1}$$

$$v_t^m = \mathbf{B}^m(s_t)h_t + \eta_v^m(s_t), \qquad \eta_v^m(s_t) \sim \mathcal{N}(\eta_v^m(s_t)|\bar{v}^m(s_t), \Sigma_v^m(s_t)). \tag{5.2}$$

and

$$u_t = f_u(h_{t-1}). \tag{5.3}$$

The variables in (5.1) describe the state vector $h_t$, the control vector $u_t$, the system matrix $\mathbf{A}(s_t)$, the

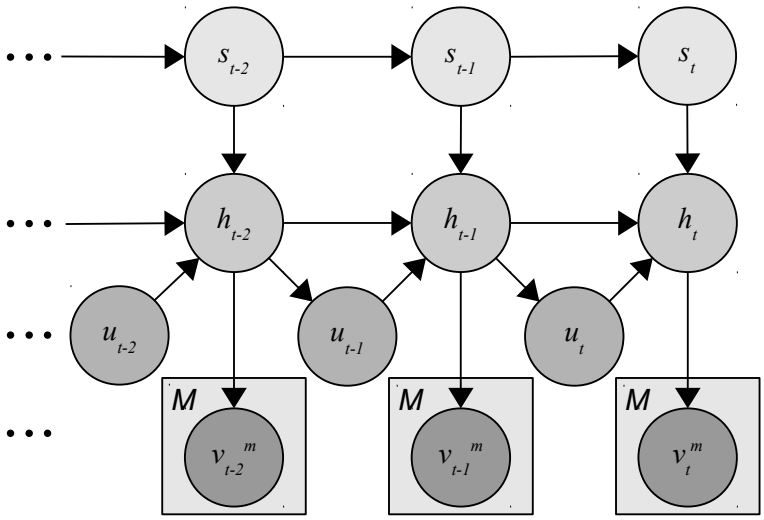



**Figure 5.1:** The adapted switching linear dynamic system model. The variable $s_t$ is the switching state, $h_t$ is the hidden vector, $u_t$ is the control vector and $v_t^m$ is the $m^{\text{th}}$ observation vector.

input matrix $\mathbf{C}(s_t)$ and the state noise process $\eta_h(s_t)$. The variables in (5.2) describe the observed measurement $v_t^m$, the observation matrix $\mathbf{B}^m(s_t)$ and the measurement noise $\eta_v^m(s_t)$. In (5.2), $v_t^m$ describes the measurement vector of the $m^{\text{th}}$ sensor that is selected from a set of $M$ sensors. The box around the $v_t^m$ variable is plate notation to indicate multiple variables of the same type. Equations (5.1) and (5.2) describe a standard LDS with a control vector [44]. The function $f(h_{t-1})$ in (5.3) is the control function. The control function transforms a vessel's previous state vector, $h_{t-1}$ to a control vector $u_t$. The parameter $s_t$ in (5.1) and (5.2) is the switching processes state. The switching process state $s_t$ is assumed to follow a first order Markov process [125]. The linear dynamic systems parameters change as $s_t$ changes. The SLDS described by (5.1), (5.2) and (5.3) is illustrated as the DBN in Figure 5.1.

The following joint probability distribution describes the DBN model in Figure 5.1

$$\begin{aligned} &p(\bar{y}_{0:T}, h_{0:T}, u_{0:T}, s_{0:T}) \\ &\quad = \prod_{t=0}^{T} \prod_{m=1}^{M} p(v_t^m|h_t) p(h_t|s_t, h_{t-1}, u_t) p(s_t|s_{t-1}, h_{t-1}). \end{aligned} \tag{5.4}$$

## 5.2 PROPOSED MODEL

The switching process state $s_t$ is proposed to model maritime vessel behaviour. To model a variety of behaviours, switching process state $s_t$ is proposed to be expanded as a DBN. The DBN is re-

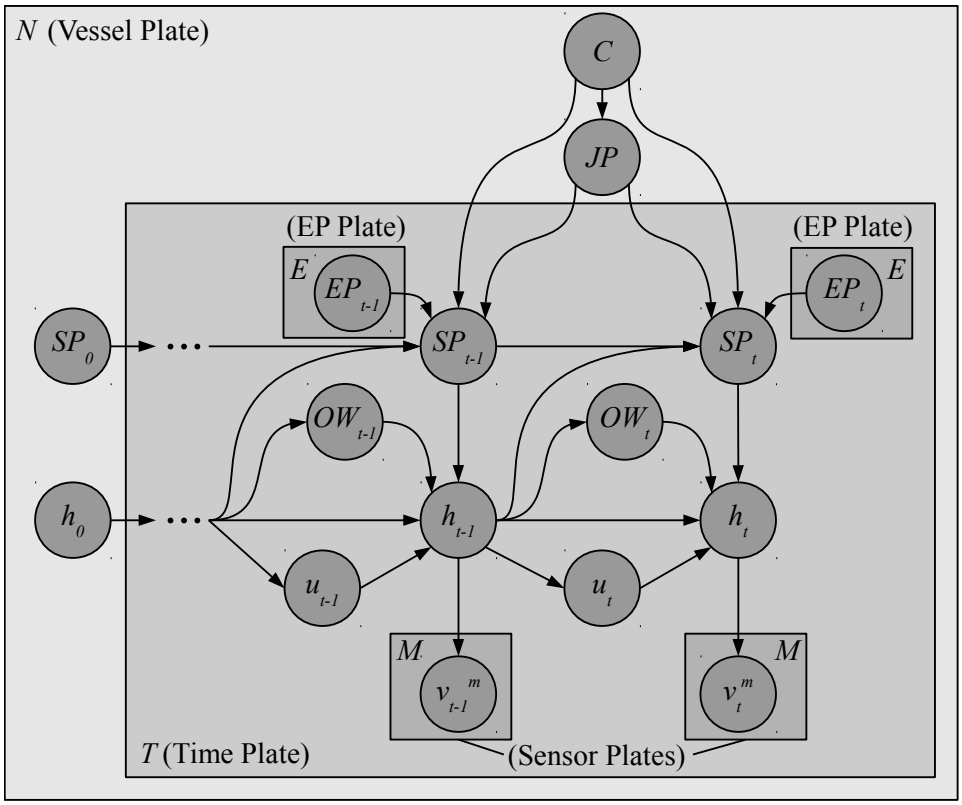



**Figure 5.2:** Dynamic Bayesian Network (DBN) behavioural model for vessels in a maritime piracy situation.

quired to include various classes of vessels, various vessel behaviours and contextual elements that influence behaviour. The DBN that is proposed for modelling vessel behaviour in a maritime piracy situation is illustrated in Figure 5.2. The DBN that forms the switching process state $s_t$, includes the class ($C$), journey parameters ($JP$), external parameters ($EP$) and the state parameter ($SP$) random variables. The $h_t$, $v_t^m$ and the $u_t$ variables are the LDS variables as described in section 5.1. The ocean and weather conditions ($OW$) variable is included to model ocean and weather effects on vessel motion.

The DBN model illustrated in Figure 5.2 is represented in plate notation. A vessel plate represents one of $N$ vessels in the environment. A set of $T$ discrete simulation time steps are represented by a set of $T$ time plates. All DBN nodes that lie within the time plate are considered to be time dependent variables. The $EP$ plate represents a set of $E$ external parameters. The set of $M$ sensors are contained in the sensor plates.

### 5.2.1 Class Variable ($C$)

The $C$ variable describes the class of a particular maritime vessel. In this study, the $C$ variable may represent either a transport vessel, a pirate vessel or a fishing vessel. Pirate vessels include pirate motherships and small pirate craft such as skiffs. A single vessel labelled as a pirate vessel represents the mothership as well as its associated skiffs. The transport class is a class that considers commercial vessels that transport goods. The transport class is a general class that describes commercial vessels. The transport class may represent vessels such as oil tankers, cargo ships, container ships and cruise ships. The fishing class describes fishing vessels that fish near the coastline and in specified fishing regions. The fishing class may include private or commercial fishing vessels.

The transport vessel and fishing vessel class probability distributions may be determined by considering ocean traffic statistics or ship registers. Such data may be obtained from sources such as Lloyd's register of ships [126]. The probability distribution for pirate vessels may be estimated from statistics extracted from piracy reports provided by the International Maritime Bureau (IMB).

The class variable is a discrete random variable that takes on a finite set of values. The number of values in the set is equal to the number of categories in the class.

### 5.2.2 Journey Parameters Variable ($JP$)

The $JP$ variable determines the home location, the route path and the destination location for a particular vessel. Home locations are defined as commercial, fishing or pirate ports. Destination locations may be commercial ports, fishing zones or pirate zones. A set of locations and routes are predefined. The $JP$ variable selects from the predefined list according to a probability distribution.

The $JP$ variable is a discrete variable that may be considered to consist of two values. The first value is the home location of the vessel. The second value is the destination location of the vessel. The path may be considered as an implicit parameter. Both the home and destination locations are selected from a discrete distribution of ports or destinations.

The $JP$ variable is conditionally dependent on the $C$ variable. This conditioning restricts the set of locations that may be associated with a given class. Transport vessels may only select commercial ports as home and destination locations. Fishing vessels may only select fishing ports as home loca-

tions and fishing zones as destination locations. Pirate vessels may only select pirate ports as home locations and pirate zones as destination locations.

#### 5.2.2.1 Fishing and Transport Vessel Ports and Destinations

The probability distribution for commercial ports could be inferred from commercial port rankings. Commercial port rankings may be obtained from sources such as the American Association of Port Authorities (AAPA) [127]. Fishing ports may be inferred from known fishing villages, ports and fishing vessel traffic data. Fishing zones may be inferred from specified legal fishing zones or from fishing vessel traffic data.

#### 5.2.2.2 Pirate Ports

Various locations of pirate ports have been suspected. At the time of writing, these locations are available for viewing on Google Earth [128].

#### 5.2.2.3 Gaussian Mixture Model Fitting and Pirate Zones

The pirate zones are modelled using a Gaussian mixture model (GMM). The GMM is fitted to a dataset consisting of pirate attack locations. Pirate attack locations provided in IMB published attack reports.

The GMM is a probability density function that is comprised of a set $g$ Gaussian density functions aggregated into a single distribution. The GMM is parameterised by the set of $g$ mean values and $g$ covariance matrices that correspond to each Gaussian mixture component. Each Gaussian component has a weight assigned to it. The GMM is useful in modelling clustered data [129]. Pirate attack locations are generally clustered in various pirate zones. A GMM is fitted to a set of $n$ pirate attack locations. The GMM provides an estimation of the probability density function of the pirate zones. The GMM mean and covariance values describe the centre and width of the pirate zones respectively. The weight of a mixture describes the probability of the corresponding pirate zone.

Let the vector of the pirate attack locations be represented by $\bar{b} = (\bar{b}_1^T, ..., \bar{b}_n^T)^T$. Each element $\bar{b}_j$, $j = 1, ..., n$ describes the latitude and longitude of the $j^{\text{th}}$ referenced pirate attack location. That is,

$\bar{b}_j = [\text{Lat}_j, \text{Lon}_j]^T$. The $g$-component GMM that describes the dataset is given as follows [129]

$$q(\bar{b}_j; \psi_r) = \sum_{i=1}^{g} \pi_i \phi_i(\bar{b}_j; \mu_i, \Sigma_i). \tag{5.5}$$

The parameters of the GMM fitted to the real-world dataset are described by the variable $\psi_r$. The $i^{\text{th}}$ GMM component is denoted by $\phi_i()$. The parameters of the $i^{\text{th}}$ Gaussian mixture component include $\mu_i$, $\Sigma_i$ and $\pi_i$. Variable $\mu_i$ describes the mean and $\Sigma_i$ describes covariance of the $i^{\text{th}}$ Gaussian mixture component. The mixture weight of the $i^{\text{th}}$ Gaussian mixture component is described by the variable $\pi_i$.

The likelihood for the model parameters $\psi_r$ is determined from the observed data. The likelihood is given by [129]

$$\mathscr{L}(\psi_r) = \prod_{j=1}^{n} q(\bar{b}_j; \psi_r). \tag{5.6}$$

The likelihood is often more conveniently represented as the log-likelihood. The log-likelihood is given by [129]

$$\log \mathscr{L}(\psi_r) = \sum_{j=1}^{n} \log q(\bar{b}_j; \psi_r). \tag{5.7}$$

The expectation maximization (EM) algorithm is used to fit the GMM to the pirate attack dataset. The EM algorithm is generally used to find the maximum likelihood parameters for a statistical model with latent variables. The GMM is fitted to the pirate attack dataset by determining the maximum likelihood GMM parameters given the dataset. For the latent variables required by the EM algorithm, the data vector $\bar{b}$ is considered to be incomplete. The associated set of component-label vectors $\bar{l} = (\bar{l}_1, ..., \bar{l}_n)^T$ are considered to be missing from the data vector. The component-label vectors are the latent variables required by the EM algorithm. A component-label vector $\bar{l}_j$ is a $g$-dimensional vector consisting of indicator variables. That is, $\bar{l}_j = [l_{1j}, ..., l_{ij}, ..., l_{gj}]$. The indicator variable indicates from which Gaussian mixture component an element $\bar{b}_j$ arose from. The indicator variable $l_{ij} \in \bar{l}_j$ is assigned the value 1 if the element $\bar{b}_j$ arose from the $i^{\text{th}}$ mixture component and 0 otherwise ($i = 1, ..., g; j = 1, ...n$). The complete data vector is defined as

$$\bar{b}_c = (\bar{b}^T, \bar{l}^T)^T. \tag{5.8}$$

The complete-data log likelihood for $\psi_r$ is given as [129]

$$\log \mathscr{L}_c(\psi_r) = \sum_{i=1}^{g} \sum_{j=1}^{n} l_{ij} \left( \log \pi_i + \log q(\bar{b}_{cj}; \psi_r) \right). \tag{5.9}$$

The conditional expectation $\log \mathscr{L}_c(\psi_r)$ given $\bar{b}$ is computed in the E-step of the EM algorithm. The expectation at the $k^{\text{th}}$ iteration of the algorithm is given as follows [129]

$$\begin{aligned} Q(\psi_r; \psi_r^{(k)}) &= \mathbb{E}_{\psi_r^{(k)}} \left( \log \mathscr{L}_c(\psi_r) | \bar{b} \right) \\ &= \sum_{i=1}^{g} \sum_{j=1}^{n} \tau_i(\bar{b}_j; \psi_r^{(k)}) \left( \log \pi_i + \log q(\bar{b}_{cj}; \psi_r) \right). \end{aligned} \tag{5.10}$$

The expectation of $l_{ij}$ with respect to the observed data $\bar{b}$ is described by the variable $\tau_i(\bar{b}_j; \psi_r^{(k)})$. This expectation is given as [129]

$$\tau_i(\bar{b}_j; \psi_r^{(k)}) = \tau_{ij}^{(k)} = p_{\psi_r^{(k)}}(l_{ij} = 1 | \bar{b}_j) = \frac{\pi_i \phi(\bar{b}_j; \mu_i, \Sigma_i)}{\sum_{h=1}^{g} \pi_h \phi(\bar{b}_j; \mu_h, \Sigma_h)}. \tag{5.11}$$

The global maximization of $Q(\psi_r; \psi_r^{(k)})$ with respect to $\psi_r$ is computed in M-step of the EM algorithm. For Gaussian components, a closed form expression for the maximization exists. The expression is expressed in terms of the mixture component mean vectors and covariance matrices. For the $k^{\text{th}}$ iteration, the update for the mean of the $i^{\text{th}}$ mixture component is given as follows [129]

$$\mu_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} \bar{b}_j}{\sum_{j=1}^{n} \tau_{ij}^{(k)}}. \tag{5.12}$$

For the $k^{\text{th}}$ iteration, the update for the covariance matrix of the $i^{\text{th}}$ mixture component is given as follows [129]

$$\Sigma_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} (\bar{b}_j - \mu_i^{(k+1)})(\bar{b}_j - \mu_i^{(k+1)})^T}{\sum_{j=1}^{n} \tau_{ij}^{(k)}}. \tag{5.13}$$

The update for the mixture weight in the $k^{\text{th}}$ iteration is given by [129]

$$\pi_i^{(k+1)} = \sum_{j=1}^{n} \tau_i(\bar{b}_j; \psi_r^{(k)})/n. \tag{5.14}$$

The EM algorithm iteratively alternates between the E- and M-steps until convergence. The algorithm may be considered to have converged when $\log \mathscr{L}_c(\psi_r^{(k+1)}) - \log \mathscr{L}_c(\psi_r^{(k)}) < \varepsilon$, where $\varepsilon$ is a small arbitrary value [129].

#### 5.2.2.4 Route Paths

Route paths are defined by way-points. A route consists of a set of straight lines between route points. A route begins at the home location, it passes through a set of way-points and it ends at a destination location. The set of way-points on a route are selected such that the shortest path between the home

location and destination is followed. The shortest route is determined using the $A*$ algorithm. The $A*$ algorithm is a best-first graph search algorithm that finds the shortest path between two nodes in a graph [130, 131, 20]. The $A*$ algorithm transverses through a graph following the expected path of lowest cost. A cost function is used to determine the cost of a path. The cost function combines a cost to reach the node $n$ and the cost to reach the destination node from node $n$. The cost to reach the node $n$ is denoted by $g(n)$. The cost to reach the destination node is denoted by $h(n)$. The expected cost of the path through node $n$ is given by [130, 131, 20]

$$k(n) = g(n) + h(n). \tag{5.15}$$

The value of the heuristic $g(n)$ may be determined as a sum of costs of traversing through the previously determined set of nodes to reach the current node $n$. The value of the function $h(n)$ is the distance from the current node $n$ to the goal node. The function $h(n)$ may be described by any admissible heuristic. An admissible heuristic is a heuristic that never overestimates the cost to reach the goal [20]. The Euclidean distance is a commonly used heuristic in path planning [132]. For path planning, the set of ports and way-points form nodes in the graph. The ports and way-points are linked according to a set of specified conditions. The specifications will specify paths that avoid obstacles. The $A*$ algorithm determines the shortest path between two given nodes. The pseudocode for the $A*$ algorithm is presented in Algorithm 1.

### 5.2.3 External Parameters (*EP*)

The external parameters variable represents context based external factors that influence vessel behaviour. Several external parameters may be included in the model. External factors that influence vessel behaviour may include time, date, season, relative location, weather conditions and ocean conditions. Each of the $EP$ variables included in the model is required to be observable. The $EP$ variables represent data that is provided to the model. Data may be obtained from data sources or from models.

Climatic and ocean conditions influence maritime vessel behaviour. Surface marine data is provided in the international comprehensive ocean atmosphere data set (ICOADS). The dataset contains spans from the year 1662 to the year 2007. Data observations are recorded from platforms such as buoys and ships. The dataset includes information such as sea surface temperature, air temperature, wind, humidity, air pressure, wave data and cloud cover data. Each of these listed elements may function as an external parameter. Elements may also be combined into single variables if they are related.

**Algorithm 1** The $A*$ algorithm for vessel route planning [20, 132].

**Require:** The graph containing the set of nodes and links to be evaluated.

**Require:** The costs $g(n)$ for each link in the graph.

**Require:** The start node and the destination node.

1: Define an empty set of closed nodes $\mathscr{C}$.
2: Define an empty set of open nodes $\mathscr{O}$.
3: Set $n$ as the start node.
4: Set the cost value $k(n) = h(n)$.
5: Add $n$ to $\mathscr{O}$.
6: **while** $n$ is not the destination node, **do**
7: **for** each neighbour node $m \notin \mathscr{C}$ of node $n$, **do**
8: Calculate the heuristic $h(m)$ using the specified heuristic.
9: Calculate the cost value $k(m)$ using (5.15).
10: Add $m$ to $\mathscr{O}$.
11: **end for**
12: Set $n$ as the node in $\mathscr{O}$ with the minimum cost value.
13: Move $n$ from $\mathscr{O}$ to $\mathscr{C}$.
14: Note from which node $n$ came from {This is used for retracing the optimal path in step 16}
15: **end while**
16: Retrace the optimal path.

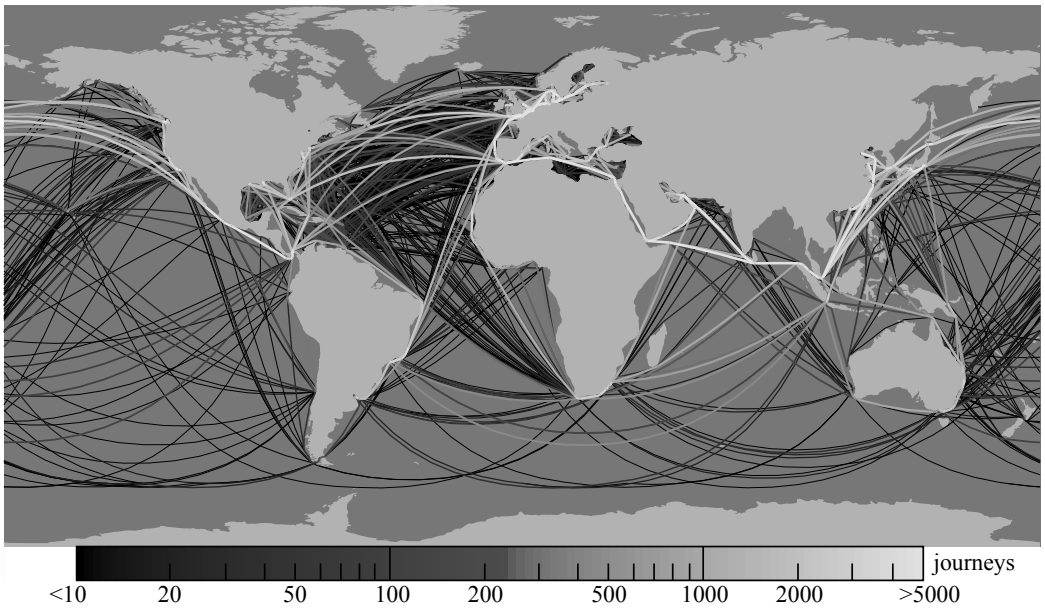



**Figure 5.3:** Commercial shipping routes for vessels larger than 10 000GT during 2007. [133]. (With permission from the Journal of The Royal Society Interface.)

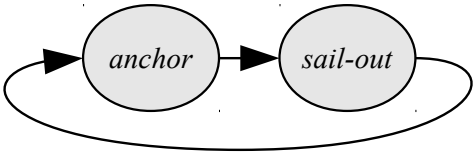
Time influences maritime vessel behaviour. For example, pirate behaviour is significantly influenced by time. Pirates generally attack during hours of darkness [22]. Pirates generally avoid conditions such as high waves, high winds, strong ocean currents and monsoon seasons [116, 2].

Behaviour may depend on the geographic location. Fishing vessels will prefer to fish in specified fishing zones. Fishing zones may be located according to fish habitat or according to fishing laws and regulations. Transport vessels tend to follow particular shipping routes. An illustration of shipping routes obtained from AIS data is illustrated in Figure 5.3 [133]. Pirate vessels prefer to operate near high traffic regions [134].

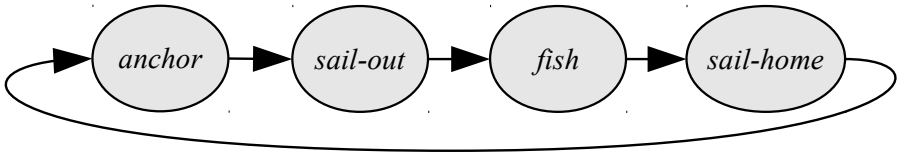
The *EP* variables are discrete random variables. The distribution of the variable will depend on the parameter that the variable represents. The distribution may be in the form of a discretised continuous distribution. Examples include time that is discretised into hours or space discretised into regions.

### 5.2.4 State Parameters Variable (*SP*)

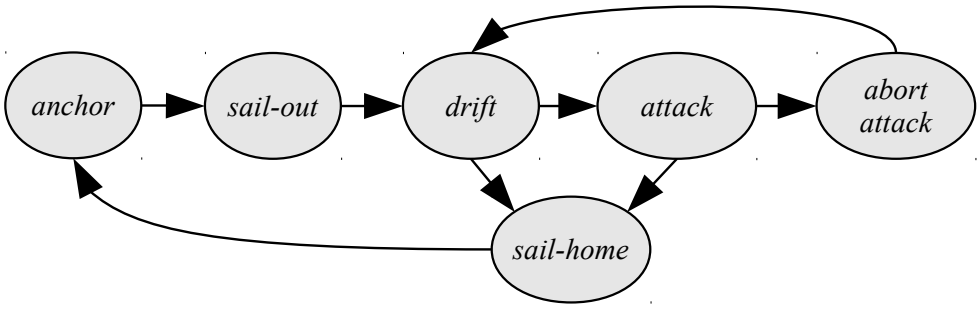
The *SP* variable describes the nature of the kinematic activity of a particular vessel. The kinematic activity or behaviour is represented by a set of states that the vessel may be in at different times. The




**(a)** Transport vessel state diagram.




**(b)** Fishing vessel state diagram.




**(c)** Pirate vessel state diagram.

**Figure 5.4:** Behavioural State transition diagrams that describe the state-parameters variable for each class.

set of states are defined as the *anchor*, *sail-out*, *sail-home*, *fish*, *drift*, *attack* and *abort-attack* states. The *SP* variable is conditioned on the *C* variable. This dependence places a constraint on which *SP* states a particular vessel class may assume. The state parameter states and their class associativity are illustrated by the state transition diagrams in Figure 5.4.

The *SP* variable is conditionally dependent on the *EP* variable. The transition probabilities between states are dependent on external parameters. This provides the ability to model the dependence of behaviour on external factors.

The transport vessel state transition diagram is illustrated in Figure 5.4a. Transport vessels are assumed to travel along the most economical route [124, 7]. The vessel assumes the *anchor* state when it is located at its home location. From the *anchor* state, *SP* variable transitions to the *sail-out* state. In the *sail-out* state, the vessel sails to its destination port. The vessel returns to the *anchor* state when it reaches its destination port.

The fishing vessel state transition diagram is illustrated in Figure 5.4b. Fishing vessels are modelled to prefer fishing at dawn and dusk. Fishermen often prefer these times as dawn and dusk are considered feeding times for many fish [135]. Fishing vessels, like pirate vessels, are susceptible to ocean and weather conditions. Conditions such as monsoons are avoided. A fishing vessel assumes the *anchor* state when it is located at its fishing port. The vessel transitions to the *sail-out* state to sail out to a fishing zone. The fishing vessel transitions to the *fish* state when it reaches the fishing zone. When the fishing period is over, the fishing vessel transitions to the *sail-home* state. When the fishing port is reached, the vessel returns to the *anchor* state.

The pirate vessel state transition diagram is illustrated in Figure 5.4c. The pirate behavioural model is based on the pirate behavioural model proposed in [7] and [134]. A pirate vessel will transition from the *anchor* state to the *sail-out* state. The pirate vessel sails from its pirate port to a pirate zone. On reaching the pirate zone, the pirate vessel transitions to the *drift* state. In the *drift* state, the pirates wait for a target [109]. If a target is detected, the pirate vessel transitions to the *attack* state and approaches the target. In the *attack* state the pirates may attack the target with high speed boats such as skiffs [22, 1]. If the pirate vessel is able to reach the target within a specified time, the attack is considered successful. The pirates sail home with the hijacked vessel and ransom it. The pirate mothership is left abandoned at the location of the attack. If the pirates cannot reach the target in the specified time the attack is considered unsuccessful. In this case, the pirates transition to the *abort-attack* state. In the *abort-attack* state the pirate skiffs return to the mothership. Once the skiffs have returned to the mothership, the *drift* state is assumed and another target is sought. If the ocean, weather or daylight conditions become unfavourable, the pirate vessel will transition to the *sail-home* state.

It should be noted that a pirate vessel is required have a perception of surrounding vessels when in the *drift* state. The $SP$ variable is thus conditionally dependent on the measurement vector $v_t^m$ of other vessels. This implies a conditional dependence between the set of $N$ vessel plates.

The $SP$ variable is a discrete random variable. The variable consists of a set of states that describe various activities. The states are linked together in a transition state diagram. The transitions are configured as to describe behaviour. The $SP$ variable transitions between states according to predefined state transition probabilities. These probabilities are influenced by the $C$ and $EP$ variables.

### 5.2.5 Ocean/Weather Conditions Variable (*OW*)

The *OW* variable describes ocean and weather conditions. The variable provides a means to model the effect of weather and ocean conditions on the motion of a vessel. Ocean and weather conditions particularly affect vessel motion when vessels are in the *drift* or *fish* states. The *OW* variable is conditioned on $h_{t-1}$. In particular, the *OW* variable is location dependent. The dependence on location provides the means to provide localised weather and ocean conditions.

The *OW* variable is considered to be observable. The variable may be obtained from oceanographic and climatic models or data. The drifting behaviour of objects in the ocean may be modelled using an autoregressive process [136]. The first order Markov process is used to model the perturbations of the position $x$ of the drifting object. The perturbations are modelled by a random Gaussian variable $\varepsilon$ with zero mean. External forces such as those due to wind and current may be modelled by the function $F(x,t)$. The trajectory of the object is given by the differential equation [136]

$$dx = F(x,t)dt + d\varepsilon. \tag{5.16}$$

The pirate and fishing vessels will tend to remain in the vicinity of their selected location. That is, they will not drift away from the selected location where fish or target vessels are located. If there is an external force such as wind or current, the pirate or fishing vessel will counteract this force as to remain in the specified region. The external forces $F(x,t)$ are thus set to zero in this study. Only the random Gaussian variable $\varepsilon$ is considered in the drifting model. The drifting model is a random walk model. The *OW* variable is thus a continuous random variable that is sampled from a random walk model. In computer vision and image processing, the random walk model has been successfully applied to the problem of loitering people detection [137]. The drifting behaviour of vessels may be considered as a form of loitering.

### 5.2.6 Linear Dynamic System Variables

The linear dynamic system variables include the state vector $h_t$, the observed measurement vector $v_t^m$ and the control vector $u_t$. These vectors are continuous random variables that are related by the state space equations provided in (5.1), (5.2) and (5.3). The $h_t$ and $v_t$ variables are specified to be vectors that contain the position and velocity of a vessel. The $h_t$ variable is defined such that $h_t = [x_t, y_t, \Delta x_t, \Delta y_t]^T$, where $(x_t, y_t)$ denote the coordinates of the vessel at time $t$.

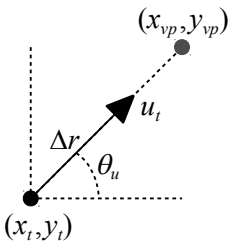



**Figure 5.5:** The control vector $u_t$ for a particular vessel at time $t$ is given by (5.17). The vector is computed using the current coordinates of the vessel $(x_t, y_t)$ and the coordinates of the destination way-point $(x_{vp}, y_{vp})$. The variable $\Delta r$ describes the vessel speed. The variable $\theta_u$ describes the angle of direction of the vessel.

The control vector $u_t$ is computed by the control function $f_u(h_t)$ such that the vessel sails through the way-points specified by the $JP$ variable. As illustrated in Figure 5.5, the control function $u_t = f_u(h_t)$ may compute the control vector using a proportional controller as follows

$$u_t = f_u(h_t) = \begin{bmatrix} \Delta r \cos(\theta_u) + \eta_u & \Delta r \sin(\theta_u) + \eta_u \end{bmatrix}^T. \tag{5.17}$$

The variable $\eta_u$ is a Gaussian random variable with zero mean. This variable introduces a small amount randomness to the motion of the vessel. The vessel travels with a speed of $\Delta r$. The angle of direction from the vessel location to the way-point is given by

$$\theta_u = \arctan\left(\frac{y_{vp} - y_t}{x_{vp} - x_t}\right). \tag{5.18}$$

The components $(x_t, y_t)$, denote the Cartesian coordinate location of a vessel at time $t$. The components $(x_{vp}, y_{vp})$, denote the Cartesian coordinate location of a way-point or port that the vessel is sailing towards.

### 5.2.7 Variable Summary

Table 5.1 provides a summary of the generative model variables. The variable type (discrete or continuous) and a possible data source is provided.

### 5.2.8 Ancestral Sampling in the Proposed Model

The ancestral sampling method described in Section 2.2.5 may be used to sample from the generative model. The process involves sampling for each vessel on the map at each simulation time step. The

**Table 5.1:** Summarising description of the generative model variables and their data sources.

| Varible | Type | Data Source |
|---|---|---|
| $C$ | Discrete | Ocean traffic statistics such as Lloyd's register of ships [126] |
| $JP$ | Discrete | Transport vessel ports: Port ranking statistics. |
| | | Transport vessel routes: Ocean traffic statistics. |
| | | Fishing ports: Ocean traffic statistics and known fishing villages. |
| | | Fishing zones: Ocean traffic statistics and known fishing regions. |
| | | Pirate ports: Suspected pirate ports [128]. |
| | | Pirate zones: GMM fitting to pirate attack data. |
| $EP$ | Discrete | Climatic/oceanographic: Models or data sources such as ICOADS |
| | | Times and seasons: Modelled through the simulation time steps. |
| | | Region Data: Land mass locations extracted from maps. |
| | | Ocean traffic statistics describe preferred sailing regions. |
| $SP$ | Discrete | Expert knowledge and/or tracking data source of modelled entity |
| $OW$ | Continuous | Ocean/weather data source or model |
| $h_t$, $v_t$ and $u_t$ | Continuous | Expert knowledge and/or tracking data source of modelled entity |

**Algorithm 2** Ancestral Sampling method for the proposed generative DBN.

**Require:** The generative graphical DBN and its associated parameters.

**for** each vessel on the map **do**
  Sample the class ($C$) variable for the vessel.
  Sample the journey parameters $JP$.
**end for**
**for** each time step $t$ and each sampled vessel **do**
  **for** each vessel on the map **do**
    Sample each of the $E$ external parameters $EP_t$.
    Sample the state parameter variable $SP_t$.
    Sample the Ocean/Wind variable $OW_t$.
    Sample the control variable $u_t$.
    Sample the state vector $h_t$.
    Sample each of the $M$ observation vectors $v_t^m$.
  **end for**
**end for**

sampling process samples from the vessel class down to the measurement vector. The measurement vector simulates sensors that observe the vessel. The generative model may be considered to generate simulated sensor data. The Ancestral sampling algorithm for the proposed generative DBN model is provided in Algorithm 2.

## 5.3 EVALUATION METHODOLOGY

The evaluation of the proposed model is performed by comparing generated data with real-world pirate data. The set of pirate attacks that are considered are limited to the cluster of attacks in the region surrounding the Gulf of Aden. A set of 235 attack locations are obtained from the 2011 IMB annual piracy report [2]. The set of 235 attack locations serves as the real-world dataset to which the generative model data is compared. The generative model is used to run a set of simulations. During each simulation, pirate attack locations are logged. A simulation is complete when at least 235 pirate attack locations have been logged. The set of logged pirate attack locations serve as the simulated pirate attack dataset. The simulated datasets are compared with the real-world dataset.

Four methods of evaluation are employed to evaluate and verify the model. In summary, the methods

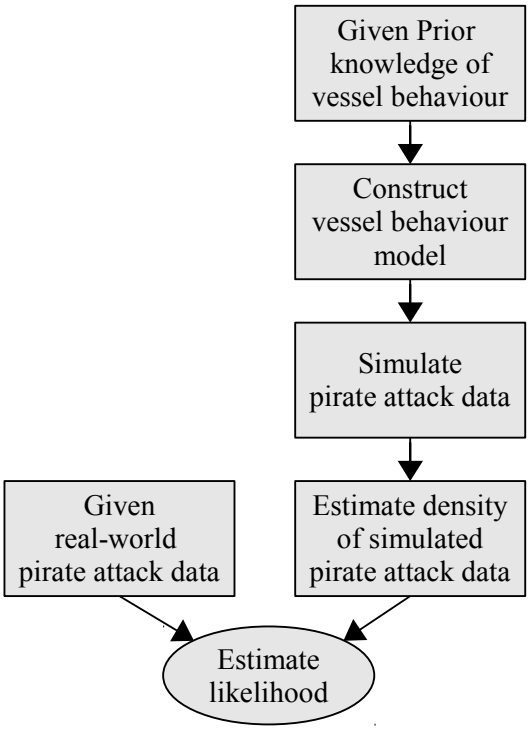



**Figure 5.6:** Block diagram of the evaluation process of the proposed model. The computed likelihood provides an indication of the capability of the generative model to generate real-world-like pirate data.

are described as follows:

1. **Likelihood based evaluation:** describes the likelihood that the generative model generated real-world-like pirate attack data.
2. **Information fusion system evaluation:** considers the generative model as an information fusion system and evaluates it's effectiveness according to robustness, quality and information gain.
3. **Cross validation based verification:** validates the generative model results using the method of cross validation and likelihood measures.
4. **Temporal evaluation:** The Bhattacharyya coefficient is used to measure the similarity between a real-world temporal pirate attack distribution and a distribution generated by the model.

### 5.3.1 Likelihood Based Evaluation

The evaluation process of the generative model is described by the block diagram illustrated in Figure 5.6. Prior knowledge of vessel behaviour is used to construct the generative model of transport,

fishing and pirate vessel behaviour. Simulated pirate attack data is generated by the generative model. A probability density function of the simulated pirate attack data is estimated. The probability density function estimate is used in computing the likelihood of the real-world dataset given the simulated dataset. The computed likelihood provides an indication of the capability of the generative model to generate real-world-like pirate data. The computed likelihood is used for the evaluation of the generative model. The generative model effectiveness is evaluated according to robustness, quality and information gain. The method of cross validation is applied to demonstrate the generalisability of the model.

#### 5.3.1.1 Simulated Data Gaussian Mixture Model

A GMM is fitted to the set of $n$ simulated pirate attack locations. The GMM the method for fitting the GMM to a dataset is described in Section 5.2.2.3. The GMM provides an estimation of the probability density function of the simulated pirate attack dataset. Let the vector of the simulated pirate attack locations be represented by $\bar{d} = (\bar{d}_1^T, ..., \bar{d}_n^T)^T$. This vector is in the same format as vector $\bar{b}$ described in section 5.2.2.3. Each element $\bar{d}_j$, $j = 1, ..., n$ describes the latitude and longitude of the $j^{\text{th}}$ referenced pirate attack location. That is, $\bar{d}_j = [\text{Lat}_j, \text{Lon}_j]^T$. The number of simulated pirate attack locations $n$ is defined as the number of real-world pirate locations in a given dataset. The $g$-component GMM that describes the simulated dataset is given as follows [129]

$$q(\bar{d}_j; \psi_s) = \sum_{i=1}^{g} \pi_i \phi_i(\bar{d}_j; \mu_i, \Sigma_i). \tag{5.19}$$

The parameters of the GMM fitted to the simulated dataset are described by the variable $\psi_s$.

#### 5.3.1.2 Model Likelihood

As described in the introduction to Section 5.3, a likelihood function of the model is required. The likelihood function should describe the likelihood of the real-world pirate dataset given the simulated pirate dataset. (Note that this likelihood should not be confused with the likelihood associated with the EM algorithm for fitting the GMM described in Section 5.2.2.3). In general, a likelihood function is defined according to a set of observations originating from a distribution with parameters $\psi$ [129]. The likelihood function that is required for evaluation purposes may thus be described according to the real-world dataset and the parameters of the probability distribution of the simulated dataset. The parameters of the simulated dataset distribution are parameters of the GMM model $\psi_s$

described in (5.19). Let the $n$ pirate attack locations of the 2011 dataset be described by the vector $\bar{b} = (b_1^T, ..., b_n^T)^T$. The log-likelihood function of the real-world dataset with respect to the simulated dataset is given as

$$\log \mathscr{L}(\psi_s) = \sum_{j=1}^{n} \log q(\bar{b}_j; \psi_s). \tag{5.20}$$

Distribution $q(\bar{b}_j; \psi_s)$ is the GMM of the simulated dataset that is evaluated at the locations given by $\bar{b}_j$, $j = 1...n$. That is, the GMM forms a 'probability surface' over the map. The set $\bar{b}$ consists of a set of points (pirate attack locations) scattered over the map. The likelihood of a single sample $\bar{b}_j$ is the value of the GMM (height of the 'probability surface') evaluated at that location of the sample on the map. The log-likelihood of the set $\bar{b}$ is the log-sum of the likelihoods of all the samples in the set. This log-likelihood, described by (5.20), provides a measure of the likelihood that the real-world data was produced by the generative model. This relation provides the means to evaluate the generative model.

### 5.3.2 Information Fusion System Evaluation

An information fusion system effectiveness may be described according to the system's robustness, quality and information gain [138, 76]. Robustness measures the consistency of the information fusion system. Quality measures the performance of the information fusion system. Information gain measures the ability of the information fusion system to provide improvement. Each of these measures is used to evaluate the generative model.

The Bhattacharyya coefficient and distance provide a measure of the similarity between two discrete distributions. By comparing the real world dataset with the simulated dataset, the Bhattacharyya coefficient and distance may be used to measure quality. The Bhattacharyya coefficient between two discrete distributions $p_1$ and $p_2$ is defined as [139, 140]

$$\rho(p_1, p_2) = \sum_{x} \sqrt{p_1(x) p_2(x)}. \tag{5.21}$$

The Bhattacharyya coefficient may be argued as a form of the cosine of the angle between two unit vectors, $\left[\sqrt{p_1(1)}, \ldots, \sqrt{p_1(\kappa)}\right]^T$ and $\left[\sqrt{p_2(1)}, \ldots, \sqrt{p_2(\kappa)}\right]^T$ from histograms with $\kappa$ bins [139].

The Bhattacharyya distance between two discrete distributions $p_1$ and $p_2$ may be computed as [139, 140]

$$D_B(p_1, p_2) = \sqrt{1 - \rho(p_1, p_2)}. \tag{5.22}$$

The information gain is measured as the information that is required to be gained for the generated distribution to match the real-world data distribution. Information may be described using entropy. Cross entropy may be used for comparing different probabilistic models [141]. In language processing, cross entropy is used to compare a sequence of words or parts of speech with a particular model [141]. In this study, cross entropy is used to compare the real world dataset with the proposed model. Suppose the real world dataset is generated by some natural probability distribution $p(y)$. The distribution $q$ denotes the GMM model of the simulated dataset. The cross entropy between $p(y)$ and $q(y;\psi_s)$, over some variable $y$, is given as [141]

$$H(p,q) = -\int p(y)\log(q(y;\psi_s))dy. \tag{5.23}$$

Given the set of real world data $\bar{b}$, the cross entropy is given by [141]

$$H(p,q) = \lim_{n\to\infty} -\frac{1}{n}\sum_{j}^{n} p(\bar{b}_j)\log(q(\bar{b}_j;\psi_s)). \tag{5.24}$$

The Shannon-McMillan-Breiman theorem dictates that, for a stationary ergodic process, the cross entropy may be written as follows [141]

$$H(p,q) = \lim_{n\to\infty} -\frac{1}{n}\sum_{j}^{n} \log(q(\bar{b}_j;\psi_s)). \tag{5.25}$$

This equation may be argued to describe the information of the GMM distribution evaluated at the data points of the real world dataset. The information will be high if there is little correlation between the real-world dataset and the GMM distribution. That is, a higher amount of information is required to 'encode' the real-world dataset.

### 5.3.3 Cross Validation Based Verification

The method of cross validation provides a means to determine how well a model generalises a dataset [142, 18]. The basis of the cross validation method is to split a dataset into a training set and a validation set. The model is trained on the training set. The validation set is used to evaluate the model. This process is repeated over a number of cross validation folds. A new random training set and validation set is produced in each fold. This provides a means to evaluate how the model performs on unseen data [20].

The method of cross validation applied to a Bayes factor based evaluation of the proposed model is described in Algorithm 3. A graphical representation of this algorithm is illustrated in Figure 5.7. A dataset consisting of real world pirate attack data is split into a training ($D_T$) and validation ($D_V$) set.

The training set is used to determine the pirate zone parameters for the generative model. The model is used to generate a simulated set ($D_S$). A GMM is fitted to the training set and a GMM is fitted to the simulated set. The likelihood of the each validation set sample is computed given the training set GMM with parameter $\psi_T$. The likelihood of the each validation set sample is computed given the simulation set GMM with parameter $\psi_S$. Bayes factor is computed given the likelihood values corresponding to each validation set sample. Bayes factor is given as follows [143]

$$K_i = \frac{q(D_V^{(i)};\psi_T)}{q(D_V^{(i)};\psi_S)}. \tag{5.26}$$

The variable $D_V^{(i)}$ is the $i^{\text{th}}$ validation set sample $i$. The numerator is the likelihood of the validation sample $D_V^{(i)}$ given the GMM of the real world data training set. The denominator is the likelihood of the validation sample $D_V^{(i)}$ given the GMM of the simulated dataset. In this application, the value of Bayes factor provides an indication of how many more times likely the real world GMM is to the simulated data GMM at the validation sample point. The median Bayes factor over all the samples is computed as follows

$$K = \text{median}\,(K_i) = \text{median}\left(\frac{q(D_V^{(i)};\psi_T)}{q(D_V^{(i)};\psi_S)}\right). \tag{5.27}$$

The median value is considered, as opposed to the mean value, as outlier samples may significantly skew the results. This process is repeated over 10 cross validation folds [18].

The Bayes factor provides a measure of how many more times likely the validation set fits the real world data model than that of the simulation data model. The closer the Bayes factor is to unity, the more similar the models are. The more similar the models are, the better the performance is of the generative model.

### 5.3.4 Temporal Evaluation

As discussed in Chapter 4, maritime pirate activity is dependent on time and season. The monthly number of attacks over the year 2011 is described by the histogram presented in Figure 4.1. The histogram is a discrete distribution that provides a description of the temporal aspect of pirate behaviour. The proposed model may be evaluated by comparing generated results to this temporal discrete distribution.

The state transition probabilities of the simulated vessels affect temporal behaviour. The pirate state transition probabilities may be adjusted such that the model generates results that are comparable to

**Algorithm 3** Cross validation based verification method for evaluating the proposed model.

**Require:** 2011 attack data consisting of the pirate attack locations.

1: **for** fold = 1 to 10 **do**
2: Sample (without replacement) 10% of the 2011 attack dataset to form the validation set $D_V$.
3: Set the remaining 90% remaining data as the training set $D_T$.
4: Generate a simulation dataset $D_S$ using pirate zone parameters determined from $D_T$.
5: Fit a GMM with parameters $\psi_T$ to $D_T$.
6: Fit a GMM with parameters $\psi_S$ to $D_S$.
7: **for** each validation sample $i$ **do**
8: Compute the likelihood the validation set sample given the GMM model of the training set $q(D_V^{(i)}; \psi_T)$.
9: Compute the likelihood the validation set sample given the GMM model of the simulation set $q(D_V^{(i)}; \psi_S)$.
10: Compute Bayes Factor for each of the two computed sample likelihoods using (5.26).
11: **end for**
12: Compute the median of the Bayes factor using (5.27).
13: **end for**
14: Plot the Bayes factor median for each fold.

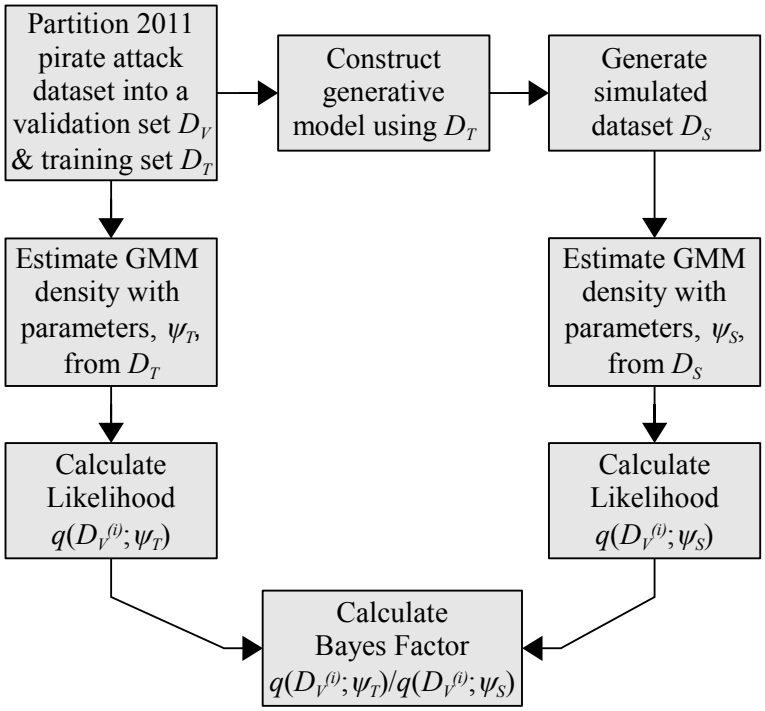



**Figure 5.7:** Cross validation based verification method for evaluating the proposed model.

Figure 4.1. The approach is to adjust the amount of time that a pirate vessel spends out at sea during each month. During peak months, the pirate vessels should spend more time out at sea searching for targets. During months or seasons where few pirate attacks occur, the simulated pirates should spend less time out at sea. To affect the time spent at sea, either the *anchor* to *sail-out* or the *drift* to *sail-home* transition probabilities may be configured. For example, if a low transition probability is associated with the *drift* to *sail-home* transition, the vessel will spend more time in the drift state.

The generative model may simulate a years worth of data. By accumulating the number of attacks over each month, a discrete distribution may be formed. This distribution may be directly compared to the temporal distribution describing the 2011 pirate attacks. The Bhattacharyya coefficient given by (5.21) may be used as a measure for comparing the distributions.

## 5.4 OPTIMISATION PROCEDURE

In general, there are two methods for parametrising the model variables. The variable may be parameterised from data or expert knowledge that describes the variable. Alternatively, the parameters of the variable may be inferred. In this study, all the parameters of the model are determinable from data

**Algorithm 4** The maximum likelihood brute force procedure for the optimising the generative model.

**Require:** The coordinates of the $n$ pirate attacks of 2011.

**Require:** The set of parameters to be optimised.

**Require:** The range and incremental value for each parameter to be optimised.

**for** each parameter to be optimised **do**
  Initialise the generative model.
  **for** each incremental value of the parameter **do**
    Run a simulation that generates a dataset of $n$ pirate attacks.
    Fit a GMM to the generated dataset.
    Determine the likelihood of the real-world data given the GMM density function.
  **end for**
**end for**
Select the set of parameters that result in the highest likelihood value.

and expert knowledge as described in Table 5.1 and Section 7.1. In some applications of the proposed model, data for defining the parameters may not exist and inference may be required. An optimisation method may be applied for inference. The optimisation algorithm should infer the parameters such that the model produces a desired result. Over a parameter space search, the outputs of the model may be compared to the desired result using a likelihood measure. A likelihood measure such as that described in section 5.3.1.2 is appropriate. The method thus searches for the parameters that produce the most likely result; which is the maximum likelihood method.

Algorithm 4 describes a brute force method for performing optimisation for parametrisation. This method is used in this study to both demonstrate the optimisation procedure and provide a sensitivity study on the parameters. The brute force method is however inefficient by nature. Algorithm 5 describes a gradient ascent based process for parameter optimisation. This algorithm is more efficient than the brute force method.

## 5.5 CLOSING REMARKS

A generative behavioural model is developed in Chapter 5. The results of this model are presented in Chapter 7. In Chapter 6, a DBN classification model is developed.

**Algorithm 5** The maximum likelihood gradient ascent procedure for the optimising the generative model.

**Require:** The coordinates of the $n$ pirate attacks of 2011.

**Require:** The set of parameters to be optimised.

**Require:** The range and incremental value for each parameter to be optimised.

Initialise the generative model.
**while** Maxima not found **do**
    Initialise the generative model with the new parameters $PARAM_{new}$.
    **for** each parameter to be optimised **do**
        Run a simulation that each generates a dataset of $n$ pirate attacks.
        Fit a GMM to the generated dataset.
        Determine the likelihood of the real-world data given the GMM density function ($\mathscr{L}_1$).
        Increase the parameter by a differential value ($\Delta x$)
        Run a simulation that each generates a dataset of n pirate attacks with the new parameter value.
        Fit a GMM to the generated dataset.
        Determine the likelihood of the real-world data given the GMM density function ($\mathscr{L}_2$)
        Compute the gradient using the likelihood of the first result, the likelihood of the differential result and the differential value: $GRAD_{param} = (\mathscr{L}_1 - \mathscr{L}_2)/\Delta x$
    **end for**
    Update the parameters according to $PARAM_{new} = PARAM_{old} + \eta GRAD_{param}$
**end while**

# CHAPTER 6

# CLASSIFICATION METHOD

INFERENCE or reasoning is the process of computing probabilities of interest conditioned on evidence [18]. The purpose of this chapter is to describe the method for inferring the class of a vessel given a sequence of tracked locations of the vessel. In Chapter 5, a generative DBN is discussed. Data is generated by sampling down through the DBN from the class root node to the sensor leaf nodes. The approach presented in Chapter 6 may be considered as the inverse of the sampling process described in Chapter 5. Inference is performed by propagating up the DBN from the sensor leaf nodes to the class root node. This provides the means to infer the class of a vessel given a sequence of tracked locations of the vessel. Classification of vessels is performed by inferring their class.

Chapter 6 begins by describing a behavioural classifier model. The Gaussian sum filtering (GSF) algorithm is used to perform inference on this model. A section is provided to present a derivation of the GSF algorithm for the behavioural classifier model. The chapter is concluded with a section on the methodology of evaluating this model and the classification results.

## 6.1 MODEL

The classical SLDS is extended to include nodes that represent contextual elements and the class. The DBN structure of the proposed behavioural classifier model is illustrated in Figure 6.1. A static class node $c$ is included in the model. The model includes set of $N_a$ contextual element nodes $a_t^n$, $n_a = \{1, ..., N_a\}$. Each of the $N_a$ contextual element nodes represents a particular contextual factor that influences behaviour. The contextual element nodes are assumed to be observable. These nodes are equivalent to the generative model $EP$ nodes described in Section 5.2.3. The $s_t$, $h_t$ and $v_t$ nodes form a SLDS. The behaviour of the system is encoded in the switching state $s_t$ variable. Observations or

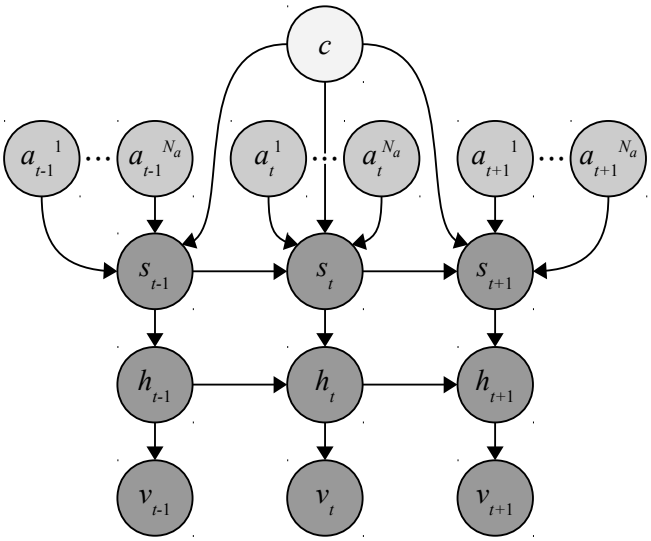



**Figure 6.1:** Proposed Inference Model

measurements of the dynamical system are described in the visible state variable $v_t$. The unobservable states of the LDS are described by the hidden state variable $h_t$. The observations $v_t$ are emitted from the hidden state. This model may be considered as a generalised form of the model proposed in Chapter 5.

The SLDS [34] of the behaviour classifier model are represented by following state space equations [18, 34]

$$h_t = \mathbf{A}(s_t, c, \bar{a}_t)h_{t-1} + \eta_t^h(s_t, c, \bar{a}_t), \tag{6.1}$$

$$v_t = \mathbf{B}(s_t, c, \bar{a}_t)h_t + \eta_t^v(s_t, c, \bar{a}_t). \tag{6.2}$$

The discrete contextual element nodes variable $\bar{a}_t$ refers to all $a_t^n$, $n_a = \{1,...,N_a\}$ at time $t$. Variables $\eta_t^h$ and $\eta_t^v$ describe the state noise process and the measurement noise respectively. These variables are assumed to be in the form of white Gaussian noise. The matrices **A** and **B** describe the state transition matrix and measurement matrix respectively. Equation (6.1) describes the transition probability distribution $p(h_t|h_{t-1}, s_t)$. Equation (6.2) describes the emission probability distribution $p(v_t|h_t)$.

The LDS variables in the proposed model are conditioned on the class $c$, the contextual elements $\bar{a}_t$ and the switching state $s_t$. LDS parameters may be defined for each combination of $c$, $\bar{a}_t$ and $s_t$. The class is inferred according to the parameters that provide the best modelling of the data. The data consists of the LDS observations and the contextual elements.

**Table 6.1:** Summarising description of the inference model variables.

| Varible | Type | Observability |
|---|---|---|
| $c$ | Discrete | Inferred |
| $\bar{a}_t$ | Discrete | Observable |
| $s_t$ | Discrete | Inferred |
| $h_t$ | Continuous | Inferred |
| $v_t$ | Continuous | Observable |

A summary of the model variables is presented in Table 6.1. The variable type (discrete or continuous) and the variables observability are described.

## 6.2 CLASSIFICATION VIA INFERENCE

Classification on the proposed model is performed by inferring the class variable $c$. Inferring the class variable given the provided data is equivalent to evaluating

$$p(c|v_{1:T}, \bar{a}_{1:T}). \tag{6.3}$$

Exact inference on an SLDS is NP-hard [34, 18]. As described in Section. 2.3.2, inference involves the marginalisation over previous unobservable (hidden) state variables. The unobservable variables for the SLDS include the switching state $s_t$ and the hidden state variable $h_t$. Marginalisation involves the integration over $h_{t-1}$ and the summation over $s_{t-1}$. In the trivial case, consider the initial filtered distribution at time $t = 0$ to be represented by a Gaussian distribution. At time $t = 1$, the integration over $h_0$ results in a Gaussian distribution. The summation over $s_0$ results in $S$ Gaussian distributions. At time $t = 2$, the summation over $s_1$ will result in $S^2$ Gaussian distributions. At time $t$,the summation over $s_{t-1}$ will result in $S^t$ Gaussian distributions. The computational complexity of exact inference increases exponentially over time.

The GSF algorithm has been proposed to perform approximate inference on the SLDS [55, 18]. This algorithm is applied to perform inference in the proposed model. In the GSF algorithm, the filtered probability distribution is represented by a mixture of Gaussians. Naturally, the number of mixture components increase exponentially over time as the DBN grows. This is addressed in the GSF algorithm by collapsing the mixture of Gaussians at each time step.

To perform inference, a forward recursion algorithm is to be formulated. The approach described in Section. 2.3.2 for formulating the filtering recursion is applied. The joint distribution of unobserved variables is marginalised over previous unobserved variables. Additionally, the evidence is split into the current and previous observations as follows

$$p(s_t, h_t, c|v_{1:t}, \bar{a}_{1:t}) = \sum_{s_{t-1}} \int_{h_{t-1}} p(s_t, h_t, s_{t-1}, h_{t-1}, c|v_t, v_{1:t-1}, \bar{a}_{1:t}). \tag{6.4}$$

Using Bayes' rule, (6.4) may be expressed as follows

$$\begin{aligned} p(s_t, h_t, c|v_{1:t}, \bar{a}_{1:t}) = \zeta \sum_{s_{t-1}} \int_{h_{t-1}} [p(v_t|s_t, h_t, s_{t-1}, h_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) \\ \cdot p(s_t, h_t, s_{t-1}, h_{t-1}, c|v_{1:t-1}, \bar{a}_{1:t})], \end{aligned} \tag{6.5}$$

where $\zeta$ is a normalising constant. By $D$-separation, $v_t$ in the first term is only conditioned on $h_t$. The other terms in the first factor are serially connected through $h_t$ to $v_t$. If $h_t$ is instantiated, the other terms are independent of $v_t$. The second term may be expanded by the chain rule of probability as follows

$$\begin{aligned} p(s_t, h_t, c|v_{1:t}, \bar{a}_{1:t}) = \zeta \sum_{s_{t-1}} \int_{h_{t-1}} [p(v_t|h_t) p(h_t|s_t, s_{t-1}, c, h_{t-1}, v_{1:t-1}, \bar{a}_{1:t}) \\ \cdot p(s_t|s_{t-1}, c, h_{t-1}, v_{1:t-1}, \bar{a}_{1:t}) p(s_{t-1}, h_{t-1}, c|v_{1:t-1}, \bar{a}_{1:t})]. \end{aligned} \tag{6.6}$$

$D$-separation (see Definition 2) may be applied to the last three factors in (6.6) by considering Figure 6.1as follows

**Second factor** $p(h_t|s_t, s_{t-1}, c, h_{t-1}, v_{1:t-1}, \bar{a}_{1:t})$**:**

The $s_t$ and $h_{t-1}$ nodes are directed into the $h_t$ node. The variable $s_{t-1}$ forms a serial connection with $h_t$ through $s_t$. The variable $h_t$ is independent of $s_{t-1}$ given $s_t$. Similarly, $h_t$ is independent of $h_{t-2}$ given $h_{t-1}$. The variables $v_{1:t-1}$ are connected to $h_t$ through diverging connections from $h_{1:t-1}$. These variables are however considered to be blocked by the instantiation of $h_{t-1}$. The variable $h_t$ is conditioned only on $s_t$ and $h_{t-1}$.

**Third factor** $p(s_t|s_{t-1}, c, h_{t-1}, v_{1:t-1}, \bar{a}_{1:t})$**:**

The $s_{t-1}$, $\bar{a}_t$ and $c$ nodes are directed into the $s_t$ node. The $h_{t-1}$ node is connected to the $s_t$ node through a diverging connection from $s_{t-1}$. The variable $s_t$ is independent of $h_{t-1}$ given $s_{t-1}$. The variable $v_{t-1}$ is connected to $s_t$ through a diverging connection through $s_{t-1}$. The variable $s_t$ is independent of $v_{t-1}$ given $s_{t-1}$. In general, the variables $v_{1:t-1}$ are blocked by $s_{t-1}$. The nodes $\bar{a}_{1:t-1}$ are blocked by the node $s_{t-1}$. The variable $s_t$ is conditioned only on $c$, $\bar{a}_t$ and $s_{t-1}$.

**Fourth factor** $p(s_{t-1}, h_{t-1}, c|v_{1:t-1}, \bar{a}_{1:t})$**:**

The node $s_{t-1}$ and $h_{t-1}$ form converging connections to node $\bar{a}_t$ through nodes $s_t$ and $h_t$ respectively. The variables $s_{t-1}$ and $h_{t-1}$ are only independent of $\bar{a}_t$ if the converging node or any of its children are instantiated. The children of $s_t$ and $h_t$ include $s_{t+1:T}$, $h_{t+1:T}$ and $v_{t:T}$. None of these variables are instantiated. The variables $s_{t-1}$ and $h_{t-1}$ are independent of the variables $\bar{a}_t$.

By $D$-separation, (6.6) may be thus simplified as follows

$$p(s_t,h_t,c|v_{1:t},\bar{a}_{1:t}) = \zeta \sum_{s_{t-1}} \int_{h_{t-1}} [p(v_t|h_t)p(h_t|s_t,h_{t-1})p(s_t|s_{t-1},c,\bar{a}_t) \cdot p(s_{t-1},h_{t-1},c|v_{1:t-1},\bar{a}_{1:t-1})]. \tag{6.7}$$

The first term is the emission probability. The second and third terms describe transition probabilities. The fourth term is the joint distribution at time $t-1$. The current joint distribution is thus a function of the previously determined joint distribution. This forms the desired recursion. The GSF algorithm performs inference by propagating $p(s_t,h_t,c|v_{1:t},\bar{a}_{1:t})$ forward using exact dynamics. This factor can be separated into continuous and discrete parts as follows

$$p(s_t,h_t,c|v_{1:t},\bar{a}_{1:t}) = p(h_t|s_t,c,v_{1:t},\bar{a}_{1:t}) \cdot p(s_t|c,v_{1:t},\bar{a}_{1:t}) \cdot p(c|v_{1:t},\bar{a}_{1:t}). \tag{6.8}$$

The first factor is a continuous distribution. The second and third factors are discrete distributions and are denoted by $\alpha_t$ and $\beta_t$ respectively. The second factor describes the switching state. The third factor describes the class. The following sections consider the continuous distribution, followed by the discrete distributions.

### 6.2.1 Continuous Distribution

The continuous factor in (6.8) is approximated by a mixture of $I$ Gaussians as follows

$$p(h_t|s_t,c,v_{1:t},\bar{a}_{1:t}) \approx \sum_{i_t=1}^{I} q(h_t|i_t,s_t,c,v_{1:t},\bar{a}_{1:t}) \cdot q(i_t|s_t,c,v_{1:t},\bar{a}_{1:t}). \tag{6.9}$$

The variable $i_t$ is the indicator variable that references the $i^{\text{th}}$ Gaussian mixture component. The variable $i_t$ is associated with the node $h_t$ in the DBN illustrated in Figure 6.1. Approximate distributions are denoted by $q$. The first factor in (6.9) describes the $i^{\text{th}}$ Gaussian mixture component. This mixture component is parameterised by mean $f(i_t,s_t,c)$ and covariance $F(i_t,s_t,c)$. The second factor in (6.9) describes the weight of the $i^{\text{th}}$ Gaussian mixture component.

At each time step, the $I$ mixture of Gaussians is expanded to an $I \times S$ mixture of Gaussians for each

class $c = 1,...,C$. This distribution is given as follows

$$q(h_t|s_t,c,v_{1:t},\bar{a}_{1:t}) = \sum_{i_{t-1},s_{t-1}} q(h_t|i_{t-1},s_{t-1},s_t,c,v_{1:t},\bar{a}_{1:t}) \cdot q(i_{t-1},s_{t-1}|s_t,c,v_{1:t},\bar{a}_{1:t}). \tag{6.10}$$

The first factor describes a Gaussian component and the second factor describes the weight of the Gaussian component.

#### 6.2.1.1 Gaussian Mixture Components Analysis

The linear dynamic system consists of the $h_t$ and $v_t$ nodes. To analyse the Gaussian, consider the following joint distribution that describes the linear dynamic system

$$q(h_t,v_t|i_{t-1},s_{t-1},s_t,c,v_{1:t-1},\bar{a}_{1:t}) = \int_{h_{t-1}} [q(h_t,v_t|h_{t-1},i_{t-1},s_{t-1},s_t,c,v_{1:t-1},\bar{a}_{1:t}) \cdot q(h_{t-1}|i_{t-1},s_{t-1},s_t,c,v_{1:t-1},\bar{a}_{1:t})]. \tag{6.11}$$

By $D$-separation (see Definition 2), $h_t$ and $v_t$ are independent of $s_{t-1}$ given $h_{t-1}$ and $s_t$. The variables $h_t$ and $v_t$ are independent of the indicator variable $i_{t-1}$ at time $t-1$. In the second term, $h_{t-1}$ is independent $s_t$ and $a_t$ given that $h_t$ and any of its descendants are not observed. Equation (6.11) is thus simplified as follows

$$q(h_t,v_t|i_{t-1},s_{t-1},s_t,c,v_{1:t-1},\bar{a}_{1:t}) = \int_{h_{t-1}} [q(h_t,v_t|h_{t-1},s_t,c,v_{1:t-1},\bar{a}_{1:t}) \cdot q(h_{t-1}|i_{t-1},s_{t-1},c,v_{1:t-1},\bar{a}_{1:t-1})]. \tag{6.12}$$

The second term corresponds to the Gaussian component in (6.9) at time $t-1$. This mixture component is parameterised by mean $f(i_{t-1},s_{t-1},c)$ and covariance $F(i_{t-1},s_{t-1},c)$. This Gaussian component may be propagated forward using the linear dynamics of the system. Lemma 1 in Appendix A may be utilised to evaluate (6.12) using this forward propagation [18]. Lemma 1 implies that (6.12) is a Gaussian with the following mean and covariance elements

$$\Sigma_{hh} = \mathbf{A}(s_t,c,\bar{a}_t)F(i_{t-1},s_{t-1},c)\mathbf{A}^T(s_t,c,\bar{a}_t) + \Sigma_h(s_t,c,\bar{a}_t), \tag{6.13}$$

$$\Sigma_{vv} = \mathbf{B}(s_t,c,\bar{a}_t)\Sigma_{hh}{}^T(s_t,c,\bar{a}_t) + \Sigma_v(s_t,c,\bar{a}_t), \tag{6.14}$$

$$\Sigma_{vh} = \Sigma_{hv}^T = \mathbf{B}(s_t,c,\bar{a}_t)\Sigma_{hh}, \tag{6.15}$$

$$\mu_v = \mathbf{B}(s_t,c,\bar{a}_t)\mathbf{A}(s_t,c,\bar{a}_t)f(i_{t-1},s_{t-1},c), \tag{6.16}$$

$$\mu_h = \mathbf{A}(s_t,c,\bar{a}_t)f(i_{t-1},s_{t-1},c). \tag{6.17}$$

The variance $\Sigma_h$ and the variance $\Sigma_v$ are defined in 5.1 and 5.2 respectively. With the extension of the SDLS, these variances are conditionally dependent on $c$ and $\bar{a}_t$ as well. To determine $q(h_t|i_{t-1}, s_{t-1}, s_t, c, v_{1:t-1}, \bar{a}_{1:t-1})$ in (6.10), (6.12) is conditioned on $v_t$ to provide the following

$$q(h_t|i_{t-1}, s_{t-1}, s_t, c, v_{1:t}, \bar{a}_{1:t-1}) = \mathcal{N}(h_t|\mu_{h|v}, \Sigma_{h|v}). \tag{6.18}$$

Lemma 2 in Appendix A is utilised to determine the values of $\mu_{h|v}$ and $\Sigma_{h|v}$ [18]. Lemma 2 indicates that the values of $\mu_{h|v}$ and $\Sigma_{h|v}$ are defined according to (6.13), (6.14), (6.15), (6.16) and (6.17) as follows

$$\mu_{h|v} = \mu_h + \Sigma_{hv}\Sigma_{vv}^{-1}(v_t - \mu_v), \tag{6.19}$$

$$\Sigma_{h|v} = \Sigma_{hh} - \Sigma_{hv}\Sigma_{vv}^{-1}\Sigma_{vh}. \tag{6.20}$$

#### 6.2.1.2 Gaussian Mixture Weights Analysis

The observations $v_{1:t}$ in the Gaussian mixture weights in (6.10) may be separated as follows

$$q(i_{t-1}, s_{t-1}|s_t, c, v_{1:t}, \bar{a}_{1:t}) = q(i_{t-1}, s_{t-1}|s_t, c, v_t, v_{1:t-1}, \bar{a}_{1:t}). \tag{6.21}$$

The values $v_t$, $s_t$ and $c$ are particular realisations of their respective random variables. Using Bayes' rule, (6.21) may be written as follows

$$q(i_{t-1}, s_{t-1}|s_t, c, v_{1:t}, \bar{a}_{1:t}) = \zeta q(v_t, s_t|i_{t-1}, s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) q(i_{t-1}, s_{t-1}|c, v_{1:t-1}, \bar{a}_{1:t}), \tag{6.22}$$

where $\zeta$ is a normalization constant. By the chain rule of probability, (6.22) may be expanded as follows

$$\begin{aligned} q(i_{t-1}, s_{t-1}|s_t, c, v_{1:t}, \bar{a}_{1:t}) \propto\ & q(v_t|s_t, i_{t-1}, s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) \\ & \cdot q(s_t|i_{t-1}, s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) \\ & \cdot q(i_{t-1}|s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t-1}) \\ & \cdot p(s_{t-1}|c, v_{1:t-1}, \bar{a}_{1:t-1}). \end{aligned} \tag{6.23}$$

By $D$-separation, the third and fourth factors are conditionally independent of the external factors given at time $t$. The variables $s_{t-1}$ and $i_{t-1}$ are connected to $\bar{a}_t$ through converging connections through $s_t$ and $i_t$ respectively. The variables $s_t$, $i_t$ and their children are not observed.

The first factor in (6.23) is a Gaussian with mean $\mu_v$ and variance $\Sigma_{vv}$. The second factor is the transition distribution between states. By $D$-separation (see Definition 2), $s_t$ in the second factor is conditionally independent of $v_{1:t-1}$ given $s_{t-1}$. Furthermore, $s_t$ in the second factor is conditionally independent of $i_{1-t}$. The second factor is thus given by

$$p(s_t|s_{t-1},c,\bar{a}_t). \tag{6.24}$$

The third factor in (6.23) contains the weights of the Gaussian mixture at time $t-1$. The fourth factor in (6.23) is the distribution of the switching state at time $t-1$ given the class, data and contextual elements. This fourth factor is denoted by $\alpha_{t-1}$ and is computed in the preceding step of the GSF algorithm.

#### 6.2.1.3 Collapsing the Gaussian Mixture Components

For each class, the mixture of $I \times S$ Gaussian mixtures described in (6.10) is collapsed to an $I$ Gaussian mixture described by (6.9). A method to collapse the Gaussian mixture is to retain the $I-1$ mixture components with the highest weights. The remaining mixture components are merged to a single Gaussian with the following parameters [55]

$$\mu = \sum_i p_i \mu_i, \quad \Sigma = \sum_i p_i \left(\Sigma_i + \mu_i \mu_i^T\right) - \mu\mu^T. \tag{6.25}$$

The values $p_i$, $\mu_i$ and $\Sigma_i$ are the weight, mean and variance of the $i^{\text{th}}$ Gaussian component that is merged.

### 6.2.2 Discrete Components

The first discrete component defined in (6.8) may be expressed as follows

$$\alpha_t = p(s_t|c,v_{1:t},\bar{a}_{1:t}) = \sum_{i_{t-1},s_{t-1}} q(i_{t-1},s_{t-1},s_t|c,v_t,v_{1:t-1},\bar{a}_{1:t}). \tag{6.26}$$

The values $v_t$ and $c$ are particular realisations of their respective random variables. Using Bayes' rule, the term within the summation in (6.26) may be written as

$$\begin{aligned} q(s_t,i_{t-1},s_{t-1}|c,v_t,v_{1:t-1},\bar{a}_{1:t}) =& \zeta q(v_t|s_t,i_{t-1},s_{t-1},c,v_{1:t-1},\bar{a}_{1:t}) \\ & \cdot q(s_t,i_{t-1},s_{t-1}|c,v_{1:t-1},\bar{a}_{1:t}), \end{aligned} \tag{6.27}$$

where $\zeta$ is a normalization constant. This result may be expanded using the chain rule of probability to provide an expression for the first discrete component, $\alpha_t$:

$$\begin{aligned}\alpha_t = p(s_t|c, v_{1:t}, \bar{a}_{1:t}) \propto \sum_{i_{t-1}, s_{t-1}} & [q(v_t|s_t, i_{t-1}, s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) \\ & \cdot q(s_t|i_{t-1}, s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) \\ & \cdot q(i_{t-1}|s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t-1}) \\ & \cdot p(s_{t-1}|c, v_{1:t-1}, \bar{a}_{1:t-1})]. \end{aligned} \tag{6.28}$$

Each of these factors are determined in the calculation of the mixture weights in (6.23). As in (6.23), the third and fourth factors are $D$-separated from the external factors given at time $t$.

The second discrete component defined in (6.8) may be expressed as

$$\beta_t = p(c|v_{1:t}, \bar{a}_{1:t}) = \sum_{i_{t-1}, s_{t-1}, s_t} q(i_{t-1}, s_{t-1}, s_t, c|v_t, v_{1:t-1}, \bar{a}_{1:t}). \tag{6.29}$$

The values $v_t$ and $c$ are particular realisations of their respective random variables. Using Bayes' rule, the term within the summation in (6.29) may be written as

$$\begin{aligned} q(i_{t-1}, s_{t-1}, s_t, c|v_t, v_{1:t-1}, \bar{a}_{1:t}) = & \zeta q(v_t|s_t, i_{t-1}, s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) \\ & \cdot q(s_t, i_{t-1}, s_{t-1}, c|v_{1:t-1}, \bar{a}_{1:t}), \end{aligned} \tag{6.30}$$

where $\zeta$ is a normalization constant. This result may be expanded using the chain rule of probability to provide an expression for the second discrete component, $\beta_t$:

$$\begin{aligned}\beta_t = p(c|v_{1:t}, \bar{a}_{1:t}) \propto \sum_{i_{t-1}, s_{t-1}, s_t} & [q(v_t|s_t, i_{t-1}, s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) \\ & \cdot q(s_t|i_{t-1}, s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t}) \\ & \cdot q(i_{t-1}|s_{t-1}, c, v_{1:t-1}, \bar{a}_{1:t-1}) \\ & \cdot p(s_{t-1}|c, v_{1:t-1}, \bar{a}_{1:t-1}) \\ & \cdot p(c|v_{1:t-1}, \bar{a}_{1:t-1})]. \end{aligned} \tag{6.31}$$

The first four factors in (6.31) are determined in the calculation of the mixture weights in (6.23). As in (6.23), the third and fourth factors are $D$-separated from the external factors given at time $t$. In the fifth factor, $c$ and $\bar{a}_t$ are $D$-separated given that the converging node $s_t$ between $c$ and $\bar{a}_t$ and the children of $s_t$ are not instantiated.

Equations (6.23), (6.28) and (6.31) are determined up to a normalisation constant. After each iteration, $\alpha_t$ and $\beta_t$ are thus normalised to form probability density functions. The algorithm for the GSF procedure on the proposed model is provided in Algorithm 6.

## 6.3 EVALUATION METHODOLOGY

Standard classification evaluation methods are applied to evaluate the results of the inference method. Recall and precision have been recommended for measuring the confidence of situation assessment systems [76]. The recall, precision, F-score and accuracy are used. The values for these measures are computed using the confusion matrix. The confusion matrix $\mathbf{\Lambda} = [\Lambda(j,k)]$ is defined such that an element $\Lambda(j,k)$ contains the number of elements whose true class label is $j$ and are classified to the class with label $k$ [144]. The rows of the confusion matrix describe the true class. The columns of the confusion matrix describe the inferred class. In this study the three classes of pirate, transport and fishing vessels are considered. The confusion matrix is a $3 \times 3$ matrix.

The recall is calculated from the confusion matrix for class $j$ as follows [144, 145]

$$R_j = \frac{\Lambda(j,j)}{\sum_{k=1}^{C} \Lambda(j,k)}. \tag{6.32}$$

Recall may be described as the probability that a vessel is classified with the class label $j$, given that the true label of the vessel is $j$. Recall may also be interpreted as a sensitivity measure.

The precision is calculated from the confusion matrix for class $j$ as follows [144, 145]

$$P_j = \frac{\Lambda(j,j)}{\sum_{k=1}^{C} \Lambda(k,j)}. \tag{6.33}$$

Precision may be described as the probability that the true label of a vessel is $j$, given that the vessel is classified with the class label $j$. Precision may also be interpreted as a confidence measure.

The F-score describes the harmonic mean between the recall and precision. The F-score is given as follows [145]

$$F_j = \frac{2R_jP_j}{R_j + P_j}. \tag{6.34}$$

The overall accuracy is calculated from the confusion matrix as follows [144]

$$Accuracy = \frac{\sum_{j=1}^{C} \Lambda(j,j)}{\sum_{j=1}^{C} \sum_{k=1}^{C} \Lambda(j,k)}. \tag{6.35}$$

**Algorithm 6** Gaussian sum filtering (GSF) algorithm for the proposed inference DBN model.

**Require:** the sample data, the contextual element data, the linear dynamic system parameters and the switching transition probabilities.

At $t = 1$, initialise $\alpha_1$, $\beta_1$ and the Gaussian mixture model.
**for** $t = 2 \rightarrow T$ **do**
  {Recursion for inference at each time step}
  **for** $c = 1 \rightarrow C$ **do**
    **for** $s_t = 1 \rightarrow S$ **do**
      {Recursion for Gaussian mixture components:}
      **for** $i_{t-1} = 1 \rightarrow I$ and $s_{t-1} = 1 \rightarrow S$ **do**
        Calculate $\Sigma_{hh}$, $\Sigma_{vv}$, $\Sigma_{vh}$, $\Sigma_{hv}$, $\mu_v$ and $\mu_h$ using (6.13) through (6.17).
        Calculate $\mu_{h|v}$ and $\Sigma_{h|v}$ using (6.19) and (6.20) respectively.
        Set $q(v_t|i_{t-1}, s_{t-1}, s_t, c, v_{1:t-1}, \bar{a}_{1:t})$ in (6.23) as a Gaussian distribution parameterised by $\mu_v$ and $\Sigma_{vv}$.
        Calculate the unnormalised mixture weights given by (6.23).
      **end for**
      Collapse the $(I \times S)$ component Gaussian mixture to an $(I)$ component Gaussian mixture.
      Calculate the unnormalised $\alpha_t$ in (6.26) by marginalizing (6.28) over $i_{t-1}$ and $s_{t-1}$.
    **end for**
    Normalise $\alpha_t$.
    Calculate the unnormalised $\beta_t$ in (6.29) by marginalizing (6.31) over $i_{t-1}$, $s_{t-1}$ and $s_t$.
  **end for**
  Normalise $\beta_t$.
**end for**

As an information fusion system, the model may be considered to be applied at levels one and two of the JDL data fusion model [146]. As for the generative model, the effectiveness of the inference method as an information fusion system, may be according to quality, robustness and information gain [138, 76].

## 6.4 CLOSING REMARKS

In Chapter 6 a behavioural model is developed for the purpose of classification. The GSF algorithm is applied to this model for performing inference. In Chapter 7, behavioural data is generated using the generative model described in Chapter 5. In Chapter 8, the classification method is applied to the generated data and the results are presented.

# CHAPTER 7

# GENERATIVE MODEL RESULTS

A GENERATIVE MODEL is described in Chapter 5. The generative model may be considered as a simulation engine. Vessels are simulated in a maritime pirate situation. Vessels are established on a map. The vessels move around on the map exhibiting kinematic behaviour. Fishing vessels sail out to fish. Transport vessels sail between ports as they transport products. Pirate vessels sail out in search of vessels to hijack. Pirate vessels attack and hold vessels for ransom.

Chapter 7 presents the results of the generative model. The parameter selection and model initialisation is discussed. Model optimisation and the results are presented. The model and the results are evaluated using data fusion system metrics.

## 7.1 PARAMETER SELECTION AND MODEL INITIALISATION

The parameter selection for the DBN generative model generally involves the selection of the conditional distributions between parameters. The conditional distributions are often estimated using relative frequencies. In the case of this study, knowledge of these distributions is limited. Selection of the parameters thus involves intuitive approximations based on prior knowledge of the problem. Parameter optimisation is performed as to adjust the parameters to correspond with real-world data.

### 7.1.1 Scale

Variables and parameters are initialised according to a map of the region considered. A pixel scale was initially chosen for ease of use and for visualization purposes. In the spatial scale, one pixel represents and area of approximately 5km $\times$ 5km. In the time scale, one simulation time step represents

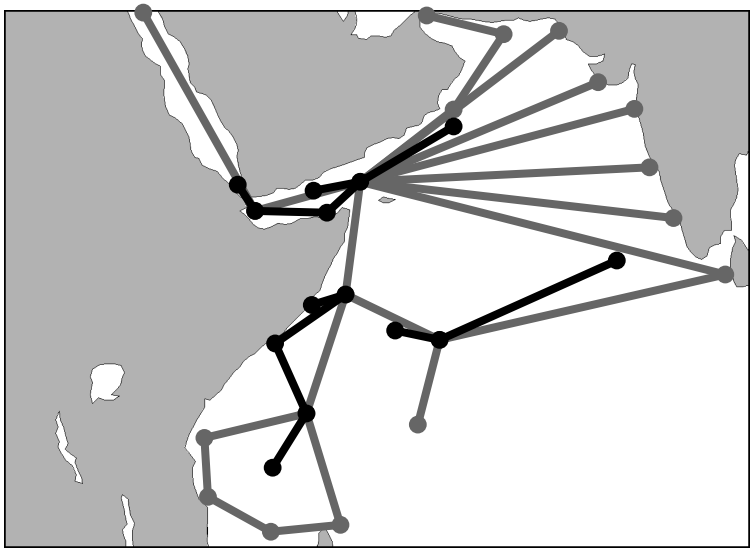


**Figure 7.1:** Map of the Gulf of Aden with vessel routes, ports and way-points. Way-points and ports are indicated by circular nodes. Links between nodes indicate routes. Grey links are routes for transport and pirate vessels. Black links are routes that extend from pirate ports and pirate zone centres.

approximately four minutes.

### 7.1.2 Class Variable ($C$)

The set of three classes of vessels $\{pirate, transport$ and $fishing\}$ is considered. The number of pirate and fishing vessels in the ocean at a particular time is unknown. The number of transport vessels in the ocean could be approximated from AIS reports. The number of transport vessels is however irrelevant if the relative number of pirate and fishing vessels are unknown. The probability distribution of the vessel classes is thus required to be estimated and optimised. The distribution of vessels is chosen as $p(transport) = 0.7$, $p(fishing) = 0.25$ and $p(pirate) = 0.05$.

### 7.1.3 Journey Parameters Variable ($JP$)

The journey parameters consists of the selection of a home port, a destination port and a sailing route. A set of ports and routes on the map of the gulf of Aden are established as illustrated in Figure 7.1. The locations of the ports and routes are established according to the spatial distribution of the 2011 pirate attacks. The locations of the 2011 pirate attacks are illustrated in Figure 7.2. The nodes and

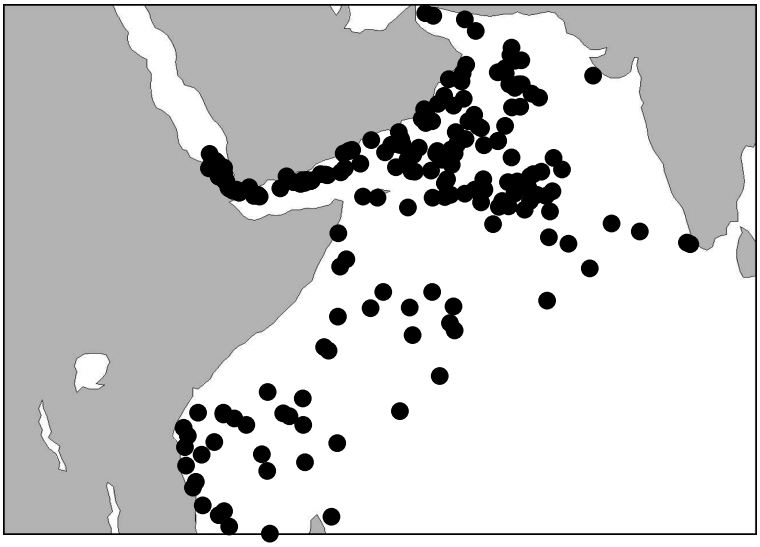


**Figure 7.2:** Pirate attack locations of 2011 indicated by the black markers [2].

links illustrated in Figure 7.1 form the graph that is used by the $A*$ algorithm for route planning.

Pirate zones are represented by a GMM as described in Section 5.2.2.3. The GMM is fitted to the 2011 pirate attack dataset. To fit the GMM to the dataset, the number of mixture components (pirate zones) must be specified. It is proposed that the likelihood of the data given number of components be used to select the optimal number of mixture components ($g$). The likelihood is a self-likelihood that is determined as follows

$$\log \mathscr{L}(\psi_r|g) = \sum_{j=1}^{n} \log q(\bar{b}_j; \psi_r|g). \tag{7.1}$$

That is, for a given number of mixture components, the GMM is fitted to the 2011 pirate attack data. The likelihood is then computed from the real world data, given the GMM. This may be performed for a range of values of $g$. A plot of the likelihood for a range of $g$ is presented in Figure 7.3. It should be noted that a single likelihood value has no meaning by itself. Its meaning is demonstrated when compared to other likelihood values.

As illustrated in Figure 7.3, the likelihood increases as the number of components increase. The problems of under-fitting and over-fitting become evident. For a single mixture component, the likelihood is lowest. This is due to under-fitting. The single Gaussian component is not able to describe the structure of the distribution sufficiently. If the number of components is set to the number of pirate attacks, each mixture component will lie at the location of a pirate attack. This describes the extreme over-fitting case. The suggested number of components is the value that lies at the "knee" of the curve

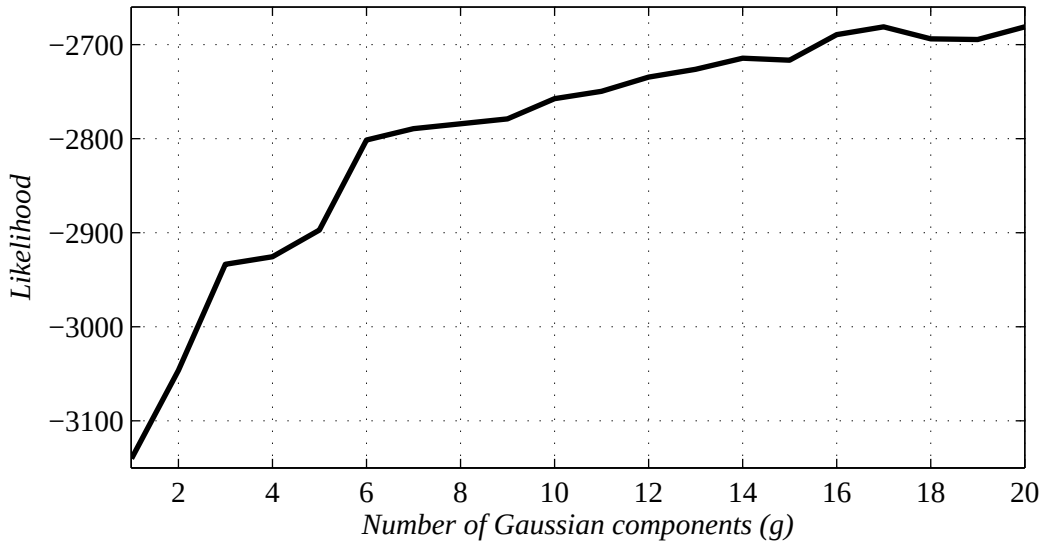



**Figure 7.3:** The self-likelihood (7.1) is plotted over a range of the number of GMM components ($g$). For each value of $g$, a GMM is fitted to the 2011 pirate attack dataset and the self likelihood is computed. The optimal number of components is the value that corresponds to the "knee" of the curve.

[147]. The knee of the curve lies at a value of $g = 6$ mixture components.

An illustration of a pirate zone GMM is provided in Figure 7.4. A pirate zone is selected according to the GMM weight parameter. A pirate vessel will sail from the pirate port to the centre of the selected pirate zone. A random location within the pirate zone is selected by sampling from the pirate zone Gaussian distribution mixture. From the centre of the distribution, the pirate vessel sails to the random location. The pirate vessel drifts at this location, waiting for a target.

### 7.1.4 External Parameters (*EP*)

The external parameters include the season of the year, the time of the day, weather and ocean conditions. Each of these parameters influences vessel behaviour. Each of the parameters is included to provide an indication of sailing conditions. With the common objective, the parameters are merged into a single variable that describes the sailing conditions. The sailing conditions are described as *favourable*, *adequate* or *poor*. The reduction of the parameters reduces the complexity of the DBN. This results in a more efficient implementation. The sailing conditions are described in Table 7.1. The sailing conditions are directly related to the weather/ocean state. That is, if the weather/ocean

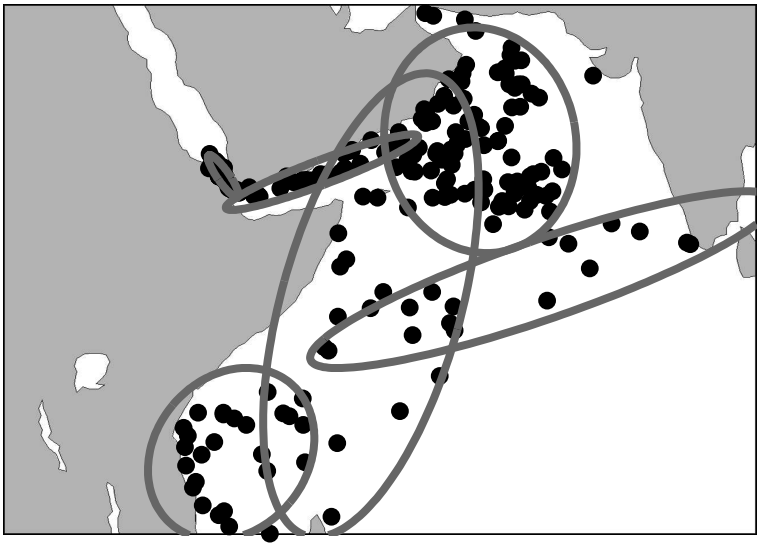


**Figure 7.4:** Pirate zones according to a GMM fitted to the 2011 pirate attack dataset. Each pirate zone is represented by a Gaussian mixture component. The grey ellipses describe the standard deviation of each Gaussian mixture component. The 2011 pirate attack locations are plotted as black markers.

**Table 7.1:** Sailing conditions according to various external parameter states.

| Sailing Conditions | Season | Period of day | Weather/Ocean |
|---|---|---|---|
| Favourable | Jan-May | Night/Dusk/Dawn | Favourable |
| Adequate | Jun-Oct | Night/Dusk/Dawn | Adequate |
| Poor | Nov-Dec | Night/Dusk/Dawn | Poor |
| Poor | * | Day | * |

state is poor, the sailing conditions are always assumed to be poor. Furthermore, if the period of the day is daytime, the sailing conditions are always assumed to be poor.

The seasonal sailing conditions are determined according to the monthly piracy reports of 2011 as illustrated in Figure 4.1. Fishermen prefer dusk and dawn for fishing. Pirates prefer dark hours for hijacking vessels. Dark hours are considered to include night, dusk and dawn. Operability of the relatively small skiffs used by pirates is highly affected by ocean and weather conditions. Poor weather or ocean conditions may include high wind, high waves, poor visibility and heavy rain. The weather and ocean conditions may be modelled as a Markov chain [148]. The state transition diagram for the *poor*, *adequate* and *favourable* states is illustrated in Figure 7.5. The stationary distribution

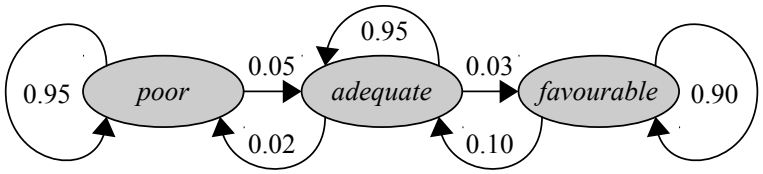



**Figure 7.5:** Ocean and weather state transition diagram. Ocean and Weather conditions transition between *poor*, *adequate* and *favourable* states.

**Table 7.2:** Transport vessel transition probabilities for the various states. '*Stay*' denotes the action to remain in the referenced state. 'Dest.' describes an indication of whether the vessel has reached its destination or not.

**(a)** *Anchor* state

| ***Stay*** | ***Sail-out*** |
|---|---|
| 0.3 | 0.7 |

**(b)** *Sail-out* state

| **Dest.** | ***Stay*** | ***Anchor*** |
|---|---|---|
| *no* | 1 | 0 |
| *yes* | 0 | 1 |

(defined in Definition 1) of the ocean and weather states are determined as $\pi = [0.23, 0.59, 0.18]^T$. This distribution indicates that as $t \to \infty$, $p(poor) = 0.23$, $p(adequate) = 0.59$ and $p(favourable) = 0.18$. These values approximately correspond to the monthly attacks illustrated in Figure 4.1. Three of the twelve months contain more than 50 attacks per month. These correspond to *favourable* conditions that occur 25% of the year. Seven of the twelve months contain between 29 and 38 attacks per month. These correspond to *adequate* conditions that occur 58% of the year. Two of the twelve months contain 20 or less attacks per month. These correspond to *poor* conditions that occur 17% of the year.

### 7.1.5 State Parameters Variable (*SP*)

The configuration of the state parameter variable involves the selection of the transition probabilities between behavioural states. The state transition diagrams are described in Figure 5.4. The transport, fishing and pirate vessel state transition probabilities are presented in Table 7.2, Table 7.3 and Table 7.4 respectively.

**Table 7.3:** Fishing vessel transition probabilities for the various states. '*Stay*' denotes the action to remain in the referenced state. 'Oc./We.' describes the ocean and weather external parameter. 'Dest.' describes an indication of whether the vessel has reached its destination or not.

**(a)** *Anchor* state

| **Oc./We.** | ***Stay*** | ***Sail-out*** |
|---|---|---|
| *favourable* | 0.1 | 0.9 |
| *adequate* | 0.5 | 0.5 |
| *poor* | 0.7 | 0.3 |

**(b)** *Sail-out* state

| **Dest.** | ***Stay*** | ***Fish*** |
|---|---|---|
| *no* | 1 | 0 |
| *yes* | 0 | 1 |

**(c)** *Fish* state

| **Oc./We.** | ***Stay*** | ***Sail-home*** |
|---|---|---|
| *favourable* | 0.9 | 0.1 |
| *adequate* | 0.3 | 0.7 |
| *poor* | 0.01 | 0.99 |

**(d)** *Sail-home* state

| **Dest.** | ***Stay*** | ***Anchor*** |
|---|---|---|
| *no* | 1 | 0 |
| *yes* | 0 | 1 |

### 7.1.6 Ocean Current/Weather Variable (*OW*)

The *OW* variable is modelled by a Markov process with perturbations modelled by a Gaussian process. The Gaussian process is a zero mean process with a covariance given by $\Sigma_{OW}$. The covariance is set as $\Sigma_{OW} = 2.5$km.

### 7.1.7 Linear Dynamic System Variables

The linear dynamic system variable configuration involves the specification of the system matrix $A(s_t)$, the observation matrix $B^m(s_t)$, the input matrix $C(s_t)$ and the noise parameters. The LDS parameters are configured according to the discrete white noise constant velocity model [149, 150]. The system matrix, with reference to a simulation time step of $\Delta t = 1$, is given by [149, 150]

$$A(s_t) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

**Table 7.4:** Pirate vessel transition probabilities for the various states. '*Stay*' denotes the action to remain in the referenced state. 'Oc./We.' describes the ocean and weather external parameter. 'Dest.' is an indication of whether the vessel has reached its destination or not. 'Targ.' is an indication of whether a target is in range or not. 'Fail' is an indication of whether an attack has failed or not. 'Capt.' is an indication of whether an a vessel is captured or not.

**(a)** *Anchor* state

| **Oc./We.** | ***Stay*** | ***Sail-out*** |
|---|---|---|
| *favourable* | 0.01 | 0.99 |
| *adequate* | 0.3 | 0.7 |
| *poor* | 0.99 | 0.01 |

**(b)** *Sail-out* state

| **Dest.** | ***Stay*** | ***Drift*** |
|---|---|---|
| *no* | 1 | 0 |
| *yes* | 0 | 1 |

**(c)** *Drift* state

| **Targ.** | **Oc./We.** | ***Stay*** | ***Attack*** | ***Sail-Home*** |
|---|---|---|---|---|
| *yes* | *favourable* | 0 | 1 | 0 |
| *yes* | *adequate* | 0.1 | 0.8 | 0.1 |
| *yes* | *poor* | 0.1 | 0.1 | 0.8 |
| *no* | *favourable* | 1 | 0 | 0 |
| *no* | *adequate* | 0.9 | 0 | 0.1 |
| *no* | *poor* | 0 | 0 | 1 |

**(d)** *Attack* state

| **Fail.** | **Capt.** | ***Stay*** | ***Abort*** | ***Sail-Home*** |
|---|---|---|---|---|
| *no* | *no* | 1 | 0 | 0 |
| *yes* | *no* | 0 | 1 | 0 |
| *no* | *yes* | 0 | 0 | 1 |

**(e)** *Abort* state

| **Complete** | ***Stay*** | ***Drift*** |
|---|---|---|
| *no* | 1 | 0 |
| *yes* | 0 | 1 |

**(f)** *Sail-home* state

| **Dest.** | ***Stay*** | ***Anchor*** |
|---|---|---|
| *no* | 1 | 0 |
| *yes* | 0 | 1 |

Though the model provides a means to model multiple sensors, only one sensor is required for this study. The single sensor is modelled to measure the location of a vessel. The observation matrix for this sensor is thus given as

$$B(s_t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The control function controls the position of the vessel. The control function does not consider the velocity of the vessel. The input matrix is thus given as

$$C(s_t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The control vector parameters include the speed $\Delta r$ and the Gaussian noise parameter $\eta_u$. The speed is set as $\Delta r = 20$knots for pirate and fishing vessels and $\Delta r = 15$knots for transport vessels. The Gaussian noise parameter is a zero mean random variable with a covariance of $\Sigma_u = 5$km. The state noise process covariance is set as $\eta_h = 0.05$km. The measurement noise is set as $\eta_v = 0.005$km.

## 7.2 MODEL OPTIMISATION DEMONSTRATION

In this study, the parameters of the model are determinable. The parameters are determined from data sources and expert knowledge. Parameter optimisation is not necessary. However, a demonstration of the optimisation procedure may provide insight into

1. the optimisation procedure itself,
2. parameter sensitivity in the model and
3. model evaluation.

For the purpose the second and third items in the above list, the brute force optimisation method described in Algorithm 4 will be applied.

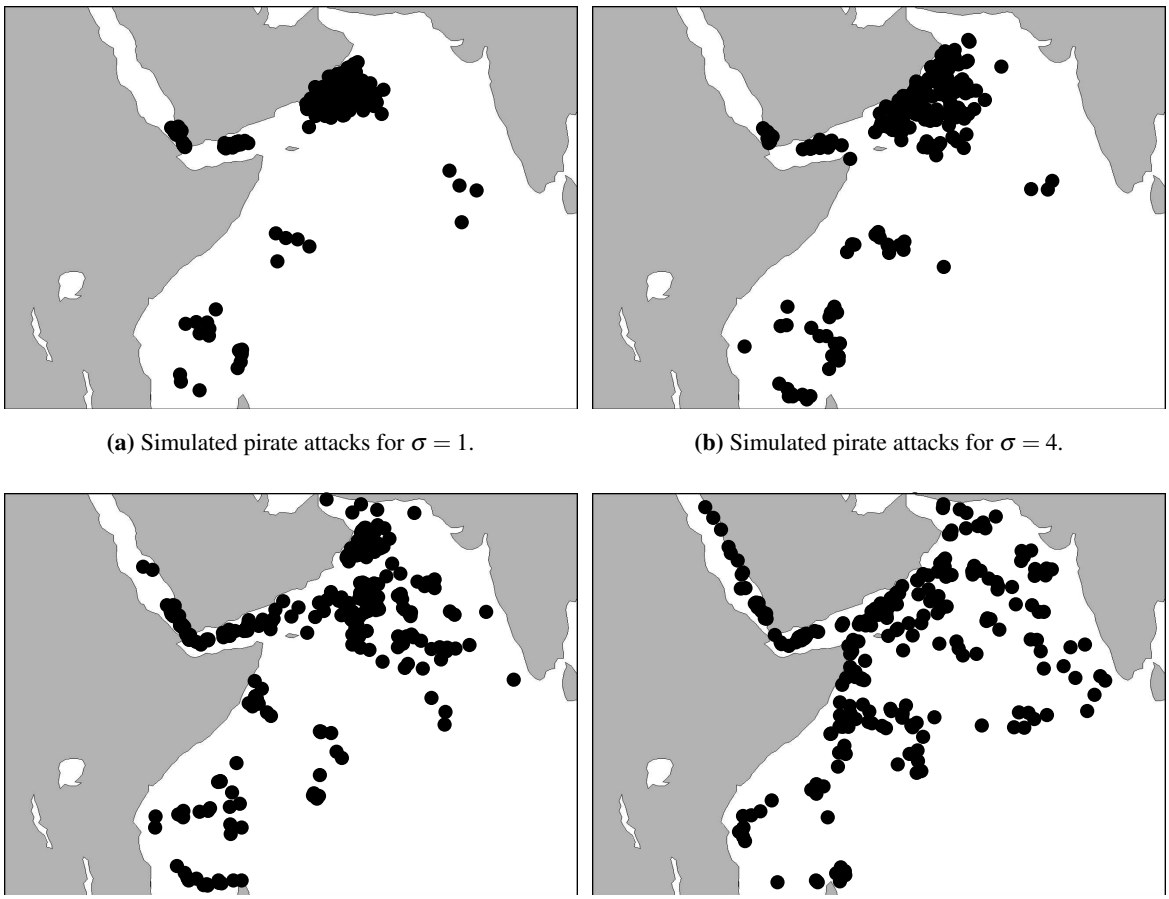
The optimisation procedure is demonstrated by determining the pirate zone parameters through optimisation. In particular, the pirate zone size parameter is optimised. Six pirate zones are defined on the map as illustrated in Figure 7.4. The sizes and locations of the pirate zones are defined according to the 2011 attack data. A probability is associated with each pirate zone. The pirate zone probability

is computed according to the number of pirate attacks that fall within the zone and the relative size of pirate zone. Each pirate zone is represented by a bivariate Gaussian distribution. The physical location of the pirate zone corresponds to the mean value of the Gaussian distribution. The physical size of the pirate zone corresponds to the covariance of the Gaussian distribution.

For the purpose of the model optimisation demonstration, the covariances of the Gaussian distributions are varied. A preselected covariance is associated with each Gaussian distribution. The set of covariance matrices of the Gaussian distributions are scaled according to a single scaling factor, $\sigma$. The optimisation process considers the set of scaling factors given by $\sigma = \{1,2,3,4,5,6,7,8,9\}$. For each value of $\sigma$, a simulation is performed. The generated set of simulated pirate attacks for $\sigma = 1$ is illustrated in Figure 7.6a. Pirate attack locations are condensed into small clusters. As a comparison, Figure 7.2 illustrates the pirate attack locations of 2011. The dataset for $\sigma = 1$ seems to demonstrate little correlation with the real-world data. The simulated pirate attacks for $\sigma = 4$ and $\sigma = 6$ are illustrated in Figure 7.6b and Figure 7.6c respectively. These results for these values seem to demonstrate a higher correlation with the 2011 pirate attack data. The generated pirate attack locations for $\sigma = 9$ are illustrated in Figure 7.6d. The distribution of the attacks for $\sigma = 9$ seems to be a more uniform distribution.

For the purpose of evaluation, a GMM is fitted to each of the simulation results that are generated for the set of the scaling factors. The GMMs are fitted to the generated data using the EM algorithm described in section 5.2.2.3. Initial parameters for the GMM are required by the EM-algorithm. The number of Gaussian mixture components was set as $g = 6$. This value is set according to the number of preselected pirate zones. The initial mean values of the mixture components were initialised with values equal to the preselected pirate zone Gaussian distribution means. The Gaussian mixture component covariance matrices were initialised as diagonal matrices. The elements on the diagonal were all set as the variance of the real-world pirate attack dataset. The initial mixture weights were set uniformly. The GMM that is fitted to the generated dataset for $\sigma = 6$ is illustrated in Figure 7.7.

The likelihood described by (5.20) is used in the optimisation procedure. The likelihood values for the datasets generated for the set of $\sigma$ values are provided in Table 7.5. The optimum value is the maximum likelihood value. The maximum likelihood corresponds to a scaling factor of $\sigma = 6$.



**(a)** Simulated pirate attacks for $\sigma = 1$.

**(b)** Simulated pirate attacks for $\sigma = 4$.

**(c)** Simulated pirate attacks for $\sigma = 6$.

**(d)** Simulated pirate attacks for $\sigma = 9$.

**Figure 7.6:** Simulated pirate attacks for various values of the pirate zone variance $\sigma$.

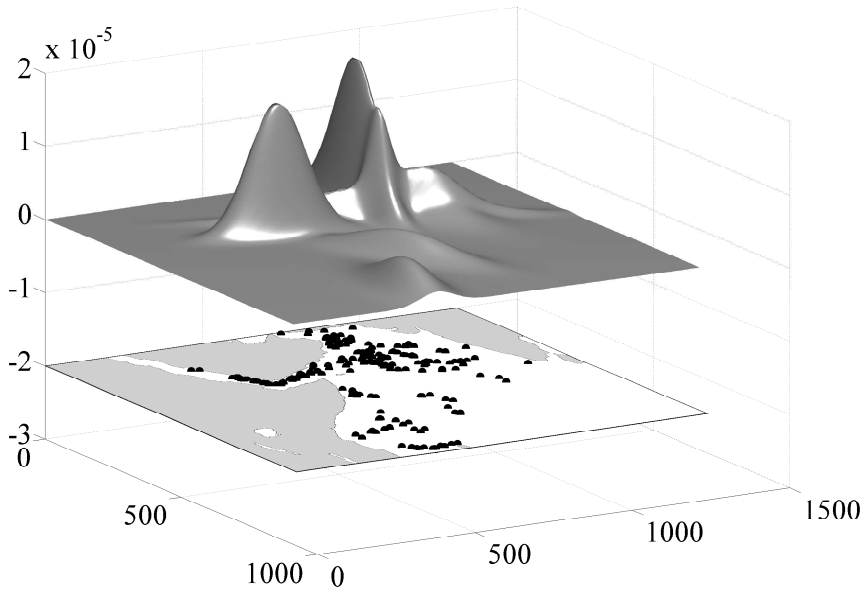



**Figure 7.7:** GMM for the simulated pirate attack data for $\sigma = 6$. The surface illustrates the GMM probability density function above the map of the region considered.

**Table 7.5:** Log-likelihood values for the pirate attack data of 2011 given the simulation data. The log-likelihood is provided for each of the simulations over the set $\sigma = \{1,2,3,4,5,6,7,8,9\}$. The row containing the maximum likelihood is presented in bold font. (It should be noted that the values presented have no meaning when considered in isolation. Their meaning is expressed when they are compared with each other.)

| $\sigma$ | $\log \mathscr{L}(\psi_s)$ |
|---|---|
| 1 | -3752 |
| 2 | -3817 |
| 3 | -3460 |
| 4 | -3358 |
| 5 | -3206 |
| **6** | **-3139** |
| 7 | -3290 |
| 8 | -3324 |
| 9 | -3195 |

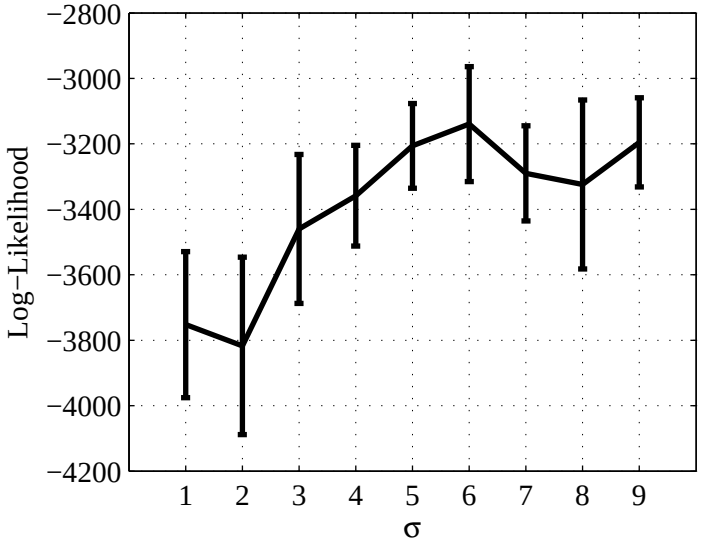



**Figure 7.8:** An error-bar plot for a set of four simulations over the set $\sigma = \{1,2,3,4,5,6,7,8,9\}$. The mean value for each $\sigma$ value is plotted as a line. The error bars describe the standard deviation of the results for each $\sigma$ value.

The brute force optimisation method of Algorithm 4 is inefficient. The bottleneck of the algorithm is the step in which the generative model is run and a dataset is generated. It takes an average of two hours to generate a dataset with the required number of pirate attacks on an Intel i7 core processor. This is however expected as the dataset describes a set of sparse events. This bottleneck is present in the more efficient gradient descent based Algorithm 5 as well. Algorithm 5 however reduces the number of overall iterations required by searching the parameter space more effectively.

## 7.3 VALIDATION OF THE RESULTS

To demonstrate some form of statistical significance, the simulations described in section 7.2 are repeated. A set of four simulations are run for each of the values in the set $\sigma = \{1,2,3,4,5,6,7,8,9\}$. The set of simulations require a considerable amount of time to complete. The number of simulation sets is thus limited to the four sets. An error-bar plot of the repeated simulation results is illustrated in Figure 7.8. The line plot illustrates the mean values of the log-likelihood for each value of sigma over the four simulations. The error bars illustrate the standard deviation values of the log-likelihood for each value of sigma over the four simulations. The results confirm that the maximum log-likelihood occurs for $\sigma = 6$.

It may noted that the log-likelihood does not seem to decrease significantly as $\sigma$ increases beyond $\sigma = 6$. By increasing $\sigma$, the pirate zones are increased in size. It is expected that the pirate attack locations will be more distributed over the region. The distribution of pirate attacks is however constrained by the structure of the transport vessel routes. Pirate attacks will only occur in regions where transport vessels sail. This constraint is demonstrated by the results for $\sigma = 9$, illustrated in Figure 7.6d. The distribution of pirate attacks has a more uniform distribution, however only over the regions where transport vessels sail. The transport routes are illustrated in Figure 7.1. The transport route constraint places a constraint on the likelihood results.

## 7.4 MODEL EFFECTIVENESS

### 7.4.1 Model Robustness

In this study a robust model is a model that is able to generalise the data effectively. The generated pirate attack locations should not necessarily be identical to the attack locations in the real-world dataset. A robust model is required to maintain the structure of the spatial distribution. A level of uncertainty on the locations of attacks is required.

The error-bar results illustrated in Figure 7.8 provide an indication of the robustness of the generative model. The error bars illustrate the standard deviation of the results for each value of $\sigma$. The log-likelihood values seem to vary more over the range of $\sigma$ values than over different simulation instances with the same $\sigma$ value. This is especially true for $\sigma \leq 5$. For example, the model with $\sigma = 6$ is more statistically significant than the model with $\sigma = 2$ since their error bars do not overlap. An overlap in the error bars indicates an overlap in the range of possible outcomes. If the error bars for two selected parameters overlap, it implies that the two parameters could produce identical results. The results for the two parameters are thus not statistically significant. It may be noted that the values of the standard deviation are similar between the various model parameters. The implication of this observation is that the model is affected more by parameter selection than the model uncertainty.

### 7.4.2 Model Quality

The quality of the model may be measured by the similarity between the real-world data and the simulated data. An indication of the similarity between the datasets is provided by the likelihood results

presented in section 7.2. The model with highest quality is indicated by the model that produces the maximum likelihood value.

It may be noted that the same data that is used for informing some of the model parameters is used for testing the model. This may result in a level of bias in the results. The cross validation based verification method discussed in Section 5.3.3 naturally addresses this problem. In cross validation, a training set is used for informing the model parameters. A separate validation set is used for evaluating the model. The cross validation based method results are presented in Section 7.5.

A simpler and more intuitive quality measure than the likelihood is Bhattacharyya distance. The Bhattacharyya distance is however a less rigorous measure in the case of this application. The results for the Bhattacharyya measurements are presented in Section 5.3.2. A smaller value of the Bhattacharyya distance indicates similarity between the considered distributions.

Two dimensional histograms are used to estimate the discrete distributions of pirate attack locations. The map is divided into square cells that form histogram bins. The distribution $p_1$ is the histogram computed from the simulated data. The distribution $p_2$ is histogram computed from the 2011 pirate attack data. The Bhattacharyya coefficients and distances, according to various histogram bin sizes, are presented in Table 7.6. The distributions appear unrelated for small histogram bin sizes. The distributions are more similar for large histogram bin sizes. The results demonstrate the scale at which the generative model performance becomes acceptable. The model performance becomes acceptable in the region of 250km$\times$250km. This is in relation to the 6400km$\times$4600km map size. This model performance range is partly constrained by the limited number of data points in the real-world dataset.

### 7.4.3 Information Gain (Cross Entropy)

Information gain is described by cross entropy. The cross entropy (5.25) values for the set of simulation parameters are presented in Table 7.7. The values of the cross entropy are presented in nats. A smaller value of the cross entropy indicates a higher accuracy of the model [141]. In terms of information, lower cross entropy indicates that a lower amount of information is required to be gained for the simulation data to match the real-world data. The simulation parameter that produces the highest information gain is the parameter that produces the minimum cross entropy value. The minimum cross entropy value corresponds to the covariance value for $\sigma = 6$. This result agrees with the maximum

**Table 7.6:** Pirate attack location spatial distribution comparisons between the simulated results and the 2011 attacks. The Bhattacharyya coefficient (5.21) and the Bhattacharyya distance (5.22) are provided for various histogram bin sizes in kilometres. The bin sizes are discretised in pixels and converted to approximated distance measures. As a frame of reference, the dimensions of the map are approximately 6400km×4600km. Values of $\rho(p_1, p_2) = 1$ and $D_B(p_1, p_2) = 0$ indicate optimal similarity.

| Bin Size | $\rho(p_1, p_2)$ | $D_B(p_1, p_2)$ |
|---|---|---|
| 25km×25km | 0.0358 | 0.9819 |
| 100km×100km | 0.2702 | 0.8543 |
| 250km×250km | 0.6684 | 0.5758 |
| 400km×400km | 0.7988 | 0.4485 |
| 500km×500km | 0.8632 | 0.3699 |

likelihood value displayed in table 7.5.

## 7.5 CROSS VALIDATION

Cross validation is performed over ten folds [18] using the method described in Section 5.3.3 and in Algorithm 3. The validation set ($D_V$) consists of 23 samples from the 2011 pirate attack dataset. The training set ($D_T$) consists of the 212 the remaining samples. A simulation set ($D_S$) of 212 sample points is generated using the generative model. A set of ten cross validation folds are performed. In each cross validation fold, a new validation, training and simulation set are created.

The log-likelihood of the validation dataset, given the models, plotted over the ten cross validation folds, is illustrated in Figure 7.9. The black curve is the likelihood of the validation dataset given the training dataset model $q(D_V; \psi_T)$. The grey curve is the likelihood of the validation dataset given the simulated dataset model $q(D_V; \psi_S)$. The trends of the two curves are similar and the deviation between the curves is low. This provides an inclination that the generative model performs well.

Bayes factor (5.26) is used to compare the simulated results with the real world dataset. The Bayes factor results for the set of ten folds is illustrated in Figure 7.10. The median value of the Bayes factor over the simulation remains at a value that is near unity. The closer the Bayes factor value is to unity, the more similar the models are. The average Bayes factor over the set of folds is 1.3.

**Table 7.7:** Cross entropy (5.25) in nats for the set of simulations over various values of $\sigma$. The values indicate the amount of information (in nats) that is required to be gained for the simulation data to match the real-world data.

| $\sigma$ | $H(p,q)$ |
|---|---|
| 1 | 18.5 |
| 2 | 19.1 |
| 3 | 17.2 |
| 4 | 14.2 |
| 5 | 13.2 |
| **6** | **12.2** |
| 7 | 13.1 |
| 8 | 12.7 |
| 9 | 12.5 |

The Bayes factor values are slightly greater than unity as the likelihood of the real world model is naturally greater than the likelihood of the simulated data model.

Box plots of the Bayes factor results are presented in Figure 7.11. The central mark on each box is the median value. The box edges are the lower and upper quartiles ($25^{\text{th}}$ and $75^{\text{th}}$ percentiles respectively). These quartiles remain within a range of 0.57 and 8.6. The box plot whiskers extend to the extreme values of the Bayes factor, excluding the outlier values. The upper extreme values remain below a value of 21. Most of the extreme values remain within a Bayes factor of 5. In relation to the application, this implies that in the extreme case, the validation sample given the real-world data GMM is only 5 times more likely than the validation sample given the simulated data GMM. The outlier values are plotted as grey '+' markers. The Bayes factor of the outliers is limited to a value of 22 in the plot. The outliers are validation samples (pirate attack locations) that the proposed model does not generalise well. The maximum outlier has a Bayes Factor of $1.3 \times 10^5$. The location of this outlier pirate attack is illustrated in Figure 7.12. It is evident that the location of this attack lies on the outskirts of the map where few other pirate attacks have occurred. This outlier is a natural outlier of the data.

In the ten cross validation folds, the simulation dataset contained 212 sample points as in the training set. By increasing the number of samples in the simulation set, the GMM estimate of the simulation

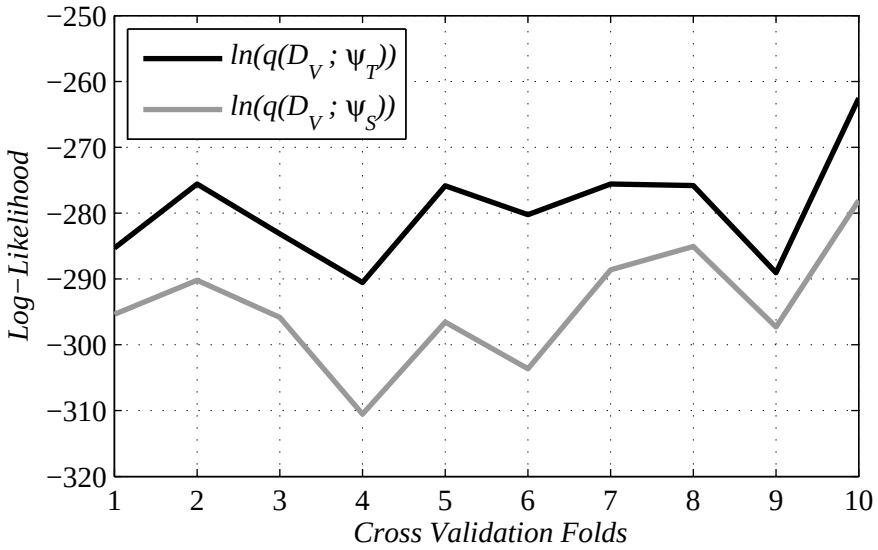



**Figure 7.9:** The log-likelihood of the validation dataset given the real-world model $q(D_V;\psi_T)$ and the simulation model $q(D_V;\psi_S)$. The curves are similar in value and in structure which indicates that the generative model performs well.

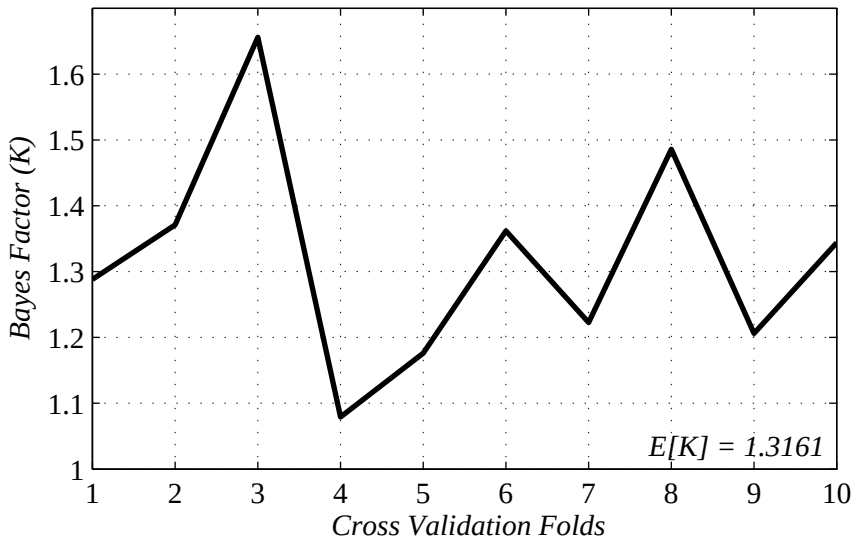



**Figure 7.10:** Median value of the Bayes factor given by (5.27) is plotted for each validation set over ten folds. The closer the Bayes factor value is to unity, the more similar the models are.

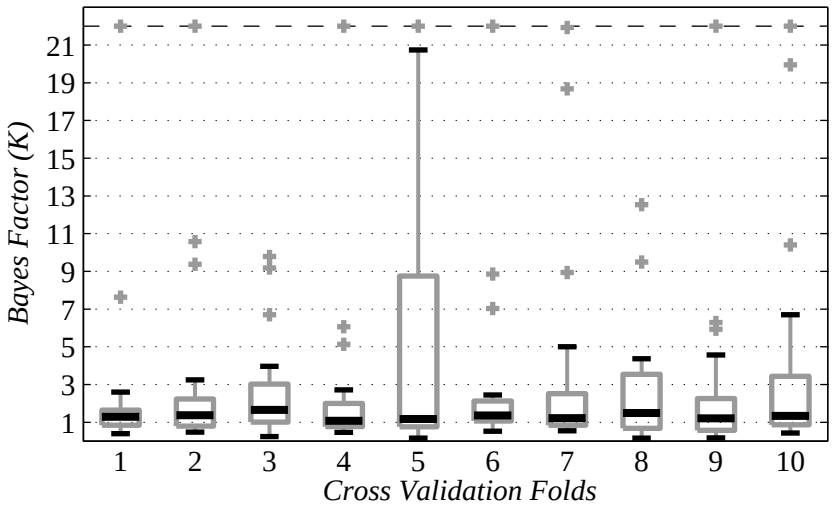



**Figure 7.11:** Box plots of Bayes factor for 23 validation samples over 10 cross validation folds.

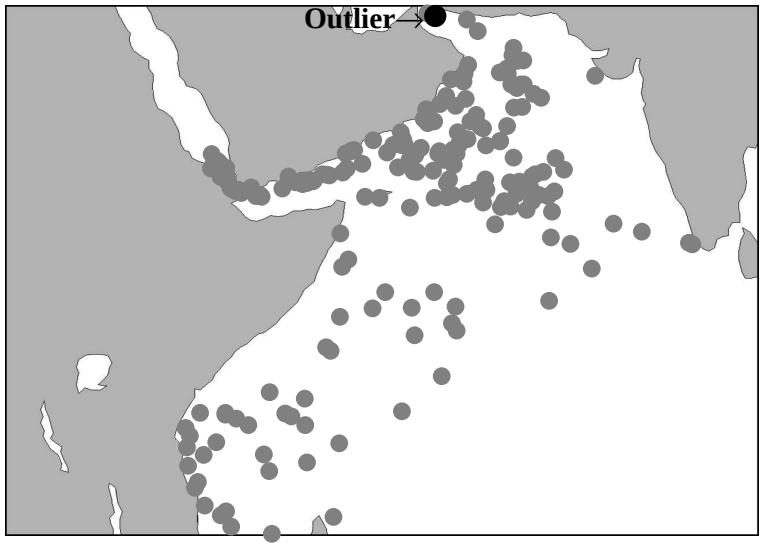



**Figure 7.12:** The labelled black marker at the top of the map describes the location of the maximum valued outlier sample in the validation dataset.

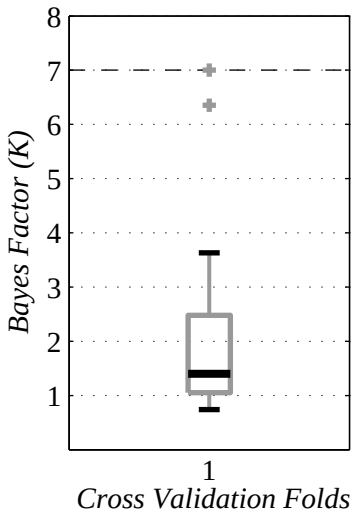



**Figure 7.13:** Box plot of Bayes factor for a simulation set with 1176 samples. This figure may be compared to Figure 7.11 where the simulation set for each cross validation fold consists of 212 samples.

samples may be improved. To demonstrate this, a set of 1176 samples are generated for the simulation set. The Bayes factor of the 23 validation samples are computed. The box plot is illustrated in Figure 7.13. The maximum outlier is reduced to a Bayes factor value of 28.8. The range of the quartiles is compressed to a range of 1.05 and 2.47. The median value is 1.4. A significant decrease in the variance of the Bayes factor results is observed.

The results are computed from a validation set that the generative model was not modelled from. The Bayes factor for these results remains near a value of unity. This indicates that the proposed model generalises the 2011 pirate attack data set well.

## 7.6 TEMPORAL RESULTS

A set of simulations are conducted to demonstrate the temporal modelling capability of the proposed model. A simulation is performed to simulate vessel behaviour over the duration of a year. The number of pirate attacks are accumulated for each month. A discrete distribution may be formed that describes the number of attacks over a set of twelve months. This distribution may be compared to the 2011 pirate attack temporal distribution. This distribution is illustrated by the histogram presented in Figure 4.1. For convenience, this figure is duplicated in Figure 7.14. The real-world temporal

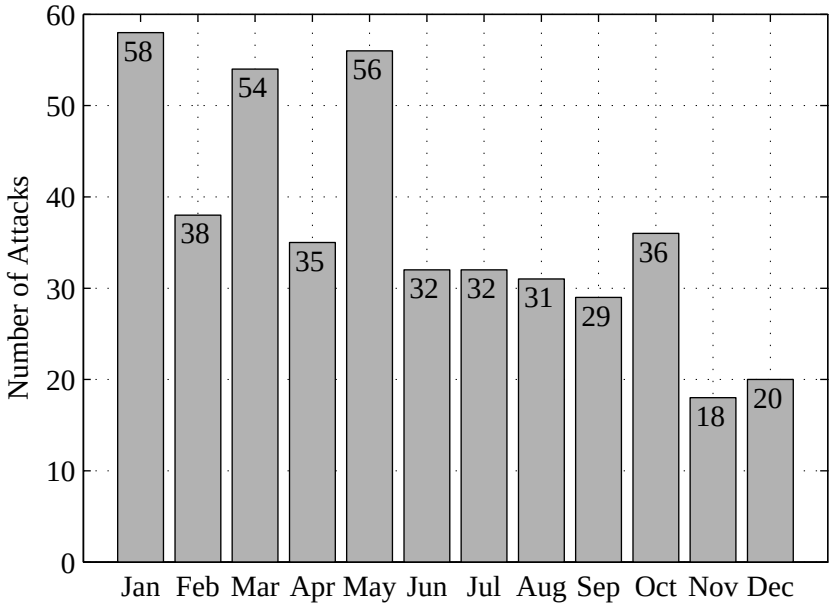



**Figure 7.14:** A bar graph describing $p_{RW}$, the number of attacks that occurred in each month of 2011 [2]. The generative model temporal results are compared to this figure.

distribution is given by

$$p_{RW} = \{58, 38, 54, 35, 56, 32, 32, 31, 29, 36, 18, 20\}. \tag{7.2}$$

This distribution describes the number of pirate attacks corresponding to each month of the year of 2011. This set consists of all the pirate attacks that occurred in 2011, across the world.

As described in Section 5.3.4, the *drift* to *sail-home* transition probability is used for incorporating the desired temporal behaviour. The *drift* to *sail-home* transition probability of a pirate vessel is configured according to the histogram in Figure 7.14. The probability for for the transition from the *drift* to *sail-home* states, for month $i \in \{1, \ldots, 12\}$ is given by

$$p(\textit{sail-home}|\textit{drift}, i) = \frac{p_{RW}(i)}{\max(p_{RW}) + \min(p_{RW})}. \tag{7.3}$$

The value in the denominator effects the total number of attacks that occur throughout the year. It does not influence the temporal distribution.

The Monte Carlo approach is adopted for generating the temporal distribution. A set of 10 simulations are performed. The average value of attacks for each month is computed over the set of simulations. This results in the discrete distribution $p_{MC}$. An element in the set $p_{MC}$ is the average value of the

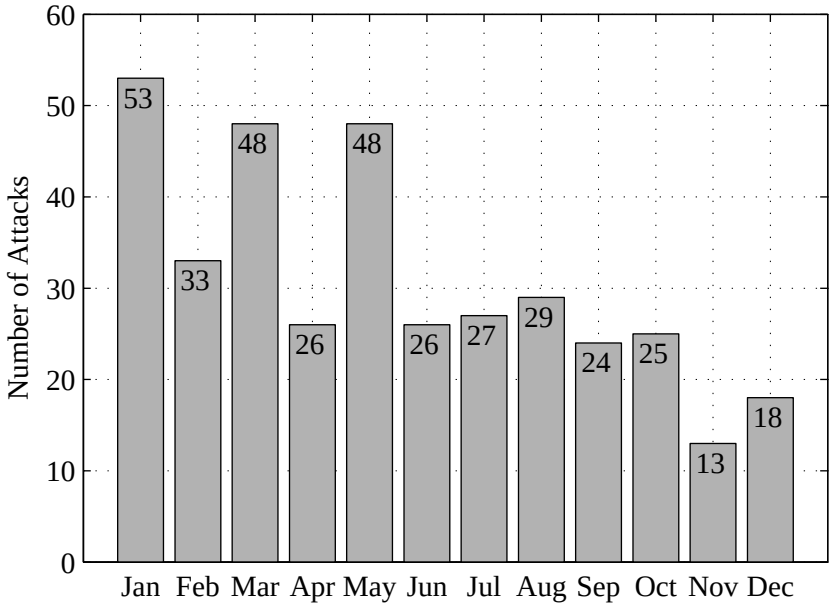



**Figure 7.15:** A histogram describing $p_{MC}$, the number of simulated attacks that occurred in each month, averaged over 10 simulations. This figure may be compared to Figure 7.14.

number of attacks over the 10 simulations, for the corresponding month. The histogram of $p_{MC}$ is presented in Figure 7.15. This figure may be compared to Figure 7.14.

The Bhattacharyya coefficient for each simulation is plotted in Figure 7.16. The values reside within a range of 0.993 and 0.997. The Bhattacharyya coefficient values are near unity indicating a high level of similarity between the distributions. Using the Monte Carlo approach, the similarity is improved. The Bhattacharyya coefficient between the Monte Carlo set, $p_{MC}$ and the 2011 pirate attack set, $p_{RW}$ is computed as

$$\rho(p_{MC}, p_{RW}) = 0.9991. \tag{7.4}$$

The value is near unity which indicates a high level of similarity between the distributions.

## 7.7 PARAMETER VARIATION AND SENSITIVITY

By varying parameters, either the spatial or the temporal distribution is affected. The class, external parameters, state parameter, ocean current/weather parameter and the linear dynamic system variables all affect the temporal distribution. The journey parameter affects the spatial distribution.

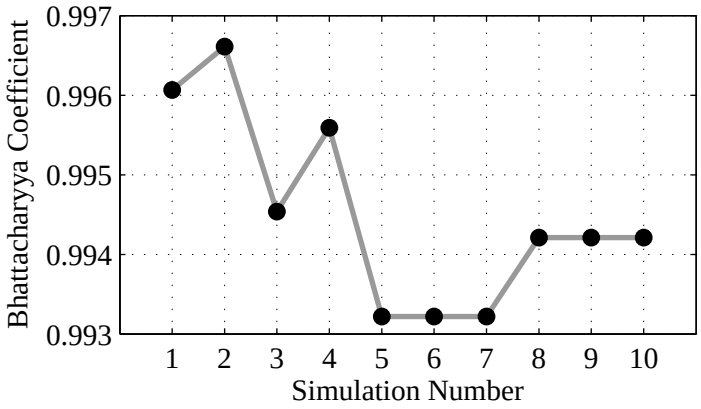
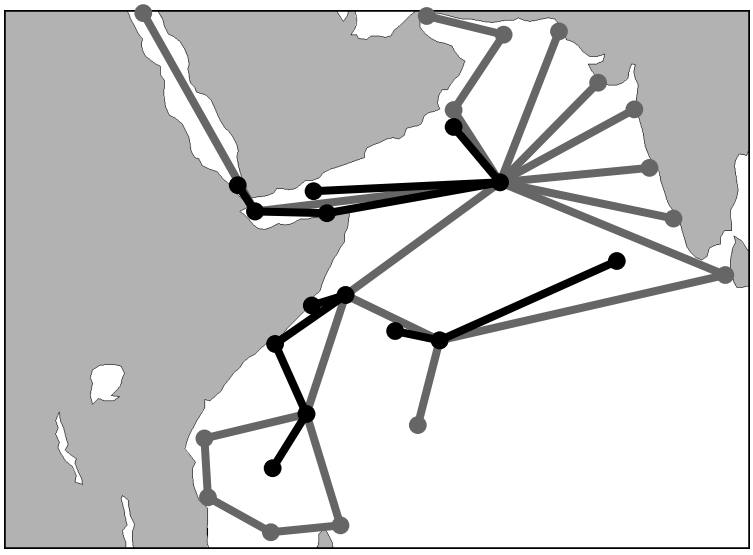



**Figure 7.16:** The Bhattacharyya coefficient comparing the 2011 temporal distribution and the 10 simulated temporal distributions. The values are near unity, indicating a high level of similarity between the distributions.

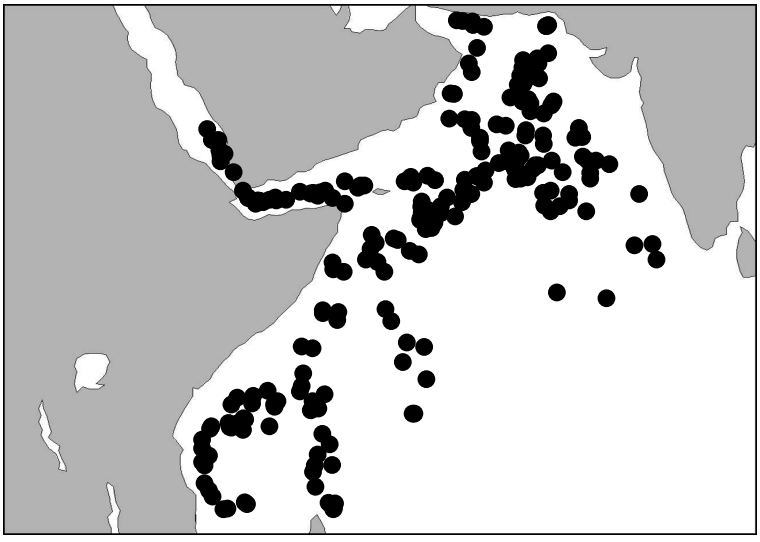
The journey parameter variable defines the home locations, destination locations and paths of vessels. The results presented in Figure 7.6 provide an indication of the effect of varying the size of a pirate zone. Shifting the pirate zone means by a small amount does not make a significant difference to the spatial distribution. This is primarily due to the fact that the transport vessel routes are fixed. Over a set of four simulations where the pirate zone means are varied by 10%, the average log-likelihood is $-3257$. This is compared with the optimal value of $-3139$ presented in Table 7.5.

By varying the home ports, destination locations and via points, the structure of the spatial distribution can be altered. As an example, a simulation is run where the vessel route paths illustrated in Figure 7.1 are altered. The results are presented in Figure 7.17. The altered vessel route paths are illustrated in Figure 7.17a. The resulting simulated set of pirate attack locations is illustrated in Figure 7.17b. As expected, the distribution of pirate attacks conforms to the vessel routes.

The class variable defines how many vessels will belong to a particular class in a simulation. By varying the class variable parameter, the distribution of vessels over the various classes is varied accordingly. As an example, the probability of sampling the transport vessel class is reduced by 10% and the probability of sampling the pirate vessel class is increased by 10%. Other than the class parameter adjustment, the same configuration used to produce Figure 7.14 is used. The temporal distribution for a set of three Monte Carlo simulations is presented in Figure 7.18. The structure of this distribution remains similar to the real-world temporal distribution illustrated in Figure 7.14. The Bhattacharyya coefficient between the simulated and the real-world temporal distributions is



**(a)** Vessel paths.



**(b)** Simulated pirate attack locations.

**Figure 7.17:** The result of adjusting the vessel route paths in reference to Figure 7.1.

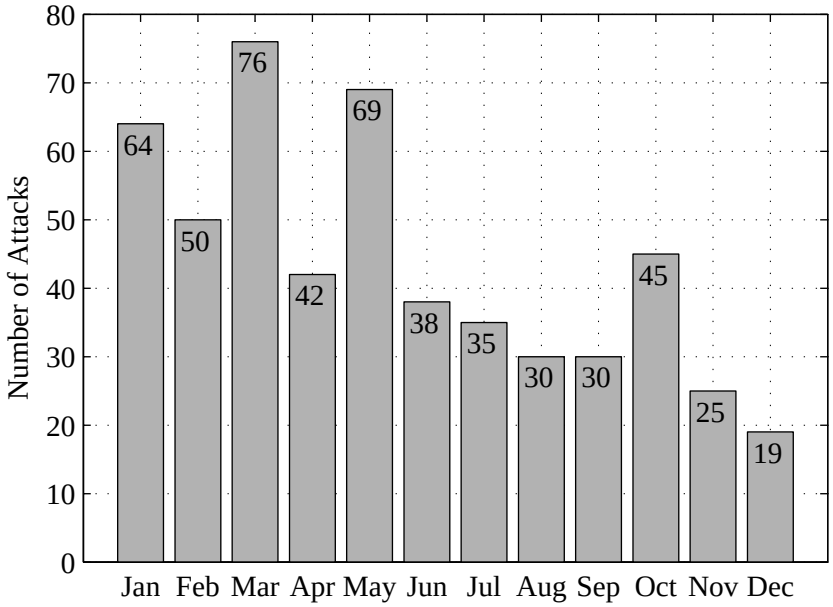



**Figure 7.18:** A histogram describing a simulated pirate attack temporal distribution. The probability of sampling the pirate class is increased by 10% and the probability of selecting the transport class is decreased by 10%. A total increase in the number of pirate attacks is evident. This figure may be compared to Figure 7.15 and Figure 7.14.

computed as 0.998. This value is near unity indicating a high level of structural similarity between the distributions. The total number of attacks over the simulation is however greater than the number of real-world attacks. The total number real-world pirate attacks is 439. The total number of simulated pirate attacks is 523. This is expected as the number of pirate vessels is increased which increases the probability of a pirate attack.

The state transition probabilities of the state parameter determines the amount of time a vessel remains in a particular state. The results presented in Section 7.6 demonstrate how the switching state transition probabilities, together with the external parameters, can be adjusted such that the model produces a particular temporal distribution. The stationary distribution of the state transition matrix provides an indication of how much time a vessel is expected to spend in each state over a simulation. This provides a means to estimate the effect of varying the state transition probabilities. The external parameters directly affect the state transition parameter. For each external parameter combination, a state transition matrix may be defined. By evaluating the stationary distribution of the state transition probabilities, the effects of adjusting external parameters may also be evaluated.

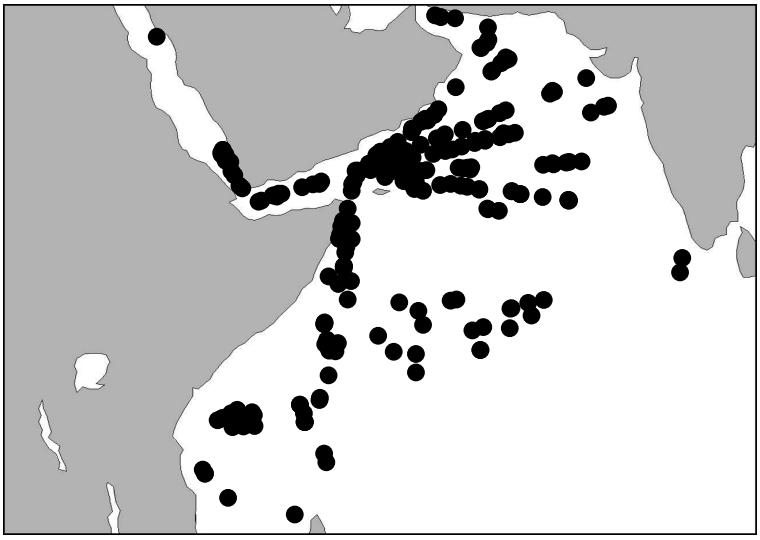


**Figure 7.19:** A plot of simulated pirate attack locations where the vessels have no variance in their motion. The attacks are limited to occur along the path lines described in Figure 7.2.

The variance of a vessel along a path is affected by the state noise processes $\eta_h$, the control vector noise process $\eta_u$ and the random Gaussian variable $\varepsilon$ in the ocean/weather conditions $OW$ parameter. These parameters add 'noise' to the motion of a vessel. This results in variance over the tracked path of the vessel. The paths of vessels are defined by straight lines as illustrated in Figure 7.2. If the variance is not included in the motion of the vessels, pirate attacks will be constrained to the path lines. An example is illustrated in Figure 7.19. If the variance is set too high, the vessel motion becomes too erratic for pirate vessels to capture transport vessels.

## 7.8 CLOSING REMARKS

The generative model results are presented in Chapter 7. In Chapter 8, the classification model is used to classify data generated by the generative model.

# CHAPTER 8

# CLASSIFICATION METHOD RESULTS

AN INFERENCE METHOD for inferring the class of a maritime vessel based on kinematic activity is presented in Chapter 6. Three classes of pirate, transport and fishing vessels are considered. In Chapter 5, a generative model is described. The class $C$ variable in the generative model provides ground truth of a vessel class. The inference method is applied to the data generated by the generative model. The results of the inference method are compared to the ground truth.

Chapter 8 begins with a discussion on the parameter selection and the method initialisation. Results on class inference are presented to demonstrate the inference method. An example illustrating the methodology for parameter selection is presented. Finally a discussion on the effectiveness of the method is given.

## 8.1 PARAMETER SELECTION AND MODEL INITIALISATION

### 8.1.1 Generative Model Configuration

The generative model described in In Chapter 5 is applied to produce data for testing and evaluating the inference method. The generative model is configured to produces a sequence of tracked locations of vessels, sailing conditions and a pirate zone indicator variable. The tracked locations are a sequence of locations of each vessel on the map. The sailing conditions describe *poor*, *adequate* and *favourable* sailing conditions. The pirate zone indicator variable provides an indication of whether a vessel is travelling in a pirate zone or not. The class of each tracked vessel is provided for ground truth.

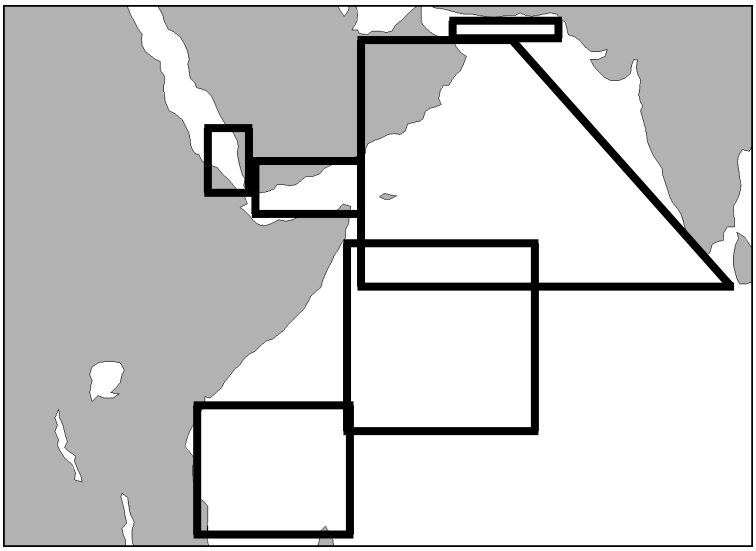


**Figure 8.1:** Pirate zone indicator variable zones. The pirate zone indicator variable indicates if a vessel is travelling within one of the six marked pirate zones.

### 8.1.2 Contextual Elements

Two contextual elements are provided by the generative model simulations. A contextual element is provided that describes the sailing conditions. The sailing conditions at time $t$ are described as $a_t^1 \in \{poor, adequate, favourable\}$. The sailing conditions describe an amalgamation of various contextual elements. These contextual elements include weather conditions, ocean conditions, time of day and season. The sailing conditions parameter is discussed as an external parameter in chapters 5 and 7.

It has been indicated that pirates prefer particular sailing conditions. The sailing conditions influence the switching state variable. Vessel speed is affected by sailing conditions. The vessel speed decreases by 50% in poor sailing conditions and by 20% in adequate sailing conditions.

The second contextual element is the indicator variable that indicates if a vessel is in a high risk pirate zone. Pirate zones are defined according to Figure 8.1. The pirate zones for the indicator variable are determined according to the pirate zones define in Figure 7.4 and according to the 2011 pirate attack data.

Pirate vessels often resemble fishing vessels [6]. In the generative model simulation, fishing behaviour of fishing vessels resembles the drifting behaviour of pirate vessels. The resemblance causes

difficulty in differentiating pirate vessels from fishing vessels. An alternative feature is required to differentiate pirate from fishing vessels. Fishing vessels are known to be targets of maritime pirates [2]. Furthermore, fishing vessels are likely to avoid high traffic regions to avoid collisions. Pirate zones are located in regions of high traffic. Fishing vessels are thus assumed to avoid pirate zones. It follows that a vessel exhibiting fishing or drifting behaviour in a pirate zone is likely to be a pirate vessel. A vessel exhibiting fishing or drifting behaviour outside of a pirate zone is likely to be a fishing vessel.

### 8.1.3 Switching States

The switching states for vessels are defined as $s_t \in \{$ *sailing*, *anchored*, *drifting*$\}$. Vessels transition between these states according to the state transition diagrams illustrated in Figure 5.4. For each vessel class, a set of state transition matrices are defined. The set of transition matrices defined for the vessel types provide a means to discriminate between various types of behaviour. Transport vessels will not enter the *drifting* state. The transport vessels alternate between the *anchored* and the *sailing* states. Vessels will naturally prefer to avoid sailing in poor sailing conditions. Fishing and pirate vessels tend to avoid poor conditions more so than transport vessels. This is due to their smaller size and higher vulnerability to sailing conditions.

Pirate vessels transition from *anchored* to *sailing* states. If a pirate vessel is in a pirate zone the vessel may transition from a *sailing* state to a *drifting* state. This transition is also influenced on the sailing conditions. Pirate vessels will tend to drift in preferred sailing conditions.

Fishing vessels behave similar to pirate vessels. Fishing vessels transition from *anchored* to *sailing* to fishing states. Fishing behaviour is modelled by the *drifting* state. Fishing vessels will only fish outside of pirate zones. From the fishing state, fishing vessels transition to the *sailing* state and return to their fishing port.

The transition matrices for pirate vessels in various sailing conditions are provided in Table 8.1. The state transition matrices for transport vessels are provided in Table 8.2. The state transition matrices for fishing vessels are provided in Table 8.3.

**Table 8.1:** Transition matrices for the pirate vessel. The transition matrices are given for the pirate zone indicator variable and for the sailing conditions of *poor*, *adequate* and *favourable*. The transition matrices correspond to the states *sailing* (s), *anchored* (a) and *drifting* (d).

| | Conditions | *poor* | | | *adequate* | | | *favourable* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Pirate zone | States | s | a | d | s | a | d | s | a | d |
| *yes* | s | 0.39 | 0.61 | 0 | 0.78 | 0.18 | 0.04 | 0.77 | 0.17 | 0.06 |
| | a | 0.19 | 0.81 | 0 | 0.3 | 0.7 | 0 | 0.4 | 0.6 | 0 |
| | d | 1 | 0 | 0 | 0.3 | 0 | 0.7 | 0.05 | 0 | 0.95 |
| *no* | s | 0.4 | 0.6 | 0 | 0.8 | 0.2 | 0 | 0.8 | 0.2 | 0 |
| | a | 0.2 | 0.8 | 0 | 0.3 | 0.7 | 0 | 0.4 | 0.6 | 0 |
| | d | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

**Table 8.2:** Transition matrices for the transport vessel. The transition matrices are given for the pirate zone indicator variable and for the sailing conditions of *poor*, *adequate* and *favourable*. The transition matrices correspond to the states *sailing* (s), *anchored* (a) and *drifting* (d).

| | Conditions | *poor* | | | *adequate* | | | *favourable* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Pirate zone | States | s | a | d | s | a | d | s | a | d |
| *yes* | s | 0.4 | 0.6 | 0 | 0.8 | 0.2 | 0 | 0.8 | 0.2 | 0 |
| | a | 0.2 | 0.8 | 0 | 0.3 | 0.7 | 0 | 0.4 | 0.6 | 0 |
| | d | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| *no* | s | 0.4 | 0.6 | 0 | 0.8 | 0.2 | 0 | 0.8 | 0.2 | 0 |
| | a | 0.2 | 0.8 | 0 | 0.3 | 0.7 | 0 | 0.4 | 0.6 | 0 |
| | d | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

**Table 8.3:** Transition matrices for the fishing vessel. The transition matrices are given for the pirate zone indicator variable and for the sailing conditions of *poor*, *adequate* and *favourable*. The transition matrices correspond to the states *sailing* (s), *anchored* (a) and *drifting* (d).

| | Conditions | *poor* | | | *adequate* | | | *favourable* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Pirate zone | States | s | a | d | s | a | d | s | a | d |
| *yes* | s | 0.4 | 0.6 | 0 | 0.8 | 0.2 | 0 | 0.8 | 0.2 | 0 |
| | a | 0.2 | 0.8 | 0 | 0.3 | 0.7 | 0 | 0.4 | 0.6 | 0 |
| | d | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| *no* | s | 0.1 | 0.8 | 0.1 | 0.8 | 0.03 | 0.17 | 0.8 | 0.025 | 0.175 |
| | a | 0.01 | 0.99 | 0 | 0.3 | 0.7 | 0 | 0.4 | 0.6 | 0 |
| | d | 0.9 | 0 | 0.1 | 0.2 | 0 | 0.8 | 0.05 | 0 | 0.95 |

### 8.1.4 State Space Representation

Pirate vessels are reported to travel at speeds between 20 and 25 knots [2]. Transport vessels are advised to travel at speeds of 10, 12, 14, 16 or 18 knots [2]. Vessel speed is a potential feature that may be used for classification. The state space model of the LDS is formulated to exploit this. The state vector as described in (6.1) is represented as follows

$$\begin{bmatrix} x_t \\ y_t \\ \Delta r_t \cos(\theta_t) \\ \Delta r_t \sin(\theta_t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \Delta r_{t-1} \cos(\theta_{t-1}) \\ \Delta r_{t-1} \sin(\theta_{t-1}) \end{bmatrix} + \eta_t^h. \tag{8.1}$$

The $x_t$ and $y_t$ variables describe the position of a vessel at a particular time $t$. The $\Delta r_{t-1}\cos(\theta_{t-1})$ and $\Delta r_{t-1}\sin(\theta_{t-1})$ variables describe the velocity of a vessel in polar coordinates. The angle of the velocity vector at time $t$ is denoted by $\theta_t$. The vessel speed at time $t$ is denoted by $\Delta r_t$. The constant velocity model assumed for the vessel motion. The constant velocity model requires that $\Delta r_t$ is a constant value over time. With constant velocity, $\Delta r_t = \Delta r$. The laws of algebra allow this constant element to be grouped within the constant state transition matrix $\mathbf{A}$. The state transition matrix may

**Table 8.4:** The speed $\Delta r$ of vessels under various sailing conditions.

| Sailing Conditions | Pirate vessel | Transport vessel | Fishing vessel |
|---|---|---|---|
| *favourable* | 20knots | 15knots | 20knots |
| *adequate* | 16knots | 12knots | 16knots |
| *poor* | 10knots | 7.5knots | 10knots |

be written as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta r & 0 \\ 0 & 1 & 0 & \Delta r \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8.2}$$

This representation deviates from the traditional discrete white noise constant velocity model as described in sources such as [150]. In the traditional model, the velocity is inferred and the sampling period is contained in $\mathbf{A}$.

The representation proposed in (8.1) allows for $\mathbf{A}$ to be parameterised by $\Delta r$. The LDS, and hence the state transition matrix $\mathbf{A}$, is conditionally dependent on $s_t$, $c$ and $\bar{a}_t$. A set of state matrices may be defined for each combination of $s_t$, $c$ and $\bar{a}_t$. This requires that $s_t$, $c$ and $\bar{a}_t$ are discrete variables. For each of combination, a particular a priori speed ($\Delta r$) may be embedded in $\mathbf{A}$. This provides a means to classify vessels according on their speed using $\mathbf{A}$.

The values for the speed $\Delta r$ are given in Table 8.4. In the generative model simulation, pirate and fishing vessels are configured to travel at speeds that are 25% faster than transport vessel speeds. In the drifting or fishing state, the transition matrix is the identity matrix. All vessels are stationary in the *anchored* state. In the *anchored* state, the transition matrix is given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The transition and emission covariance matrices are initialised as follows

$$\Sigma_h = \Sigma_v = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}.$$

The covariance matrices affect the sensitivity of the inference algorithm to noise and changes in the system. Higher values in these matrices indicate higher uncertainty of the observations. The classifier will be more tolerant of noise. Higher values however result the classifier to be less distinctive in its classification. Lower values in the covariance indicate higher certainty of the observations. These results provide a classifier that is more distinctive but less tolerant of noise.

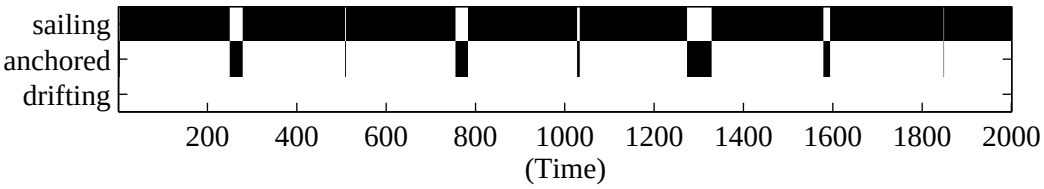
## 8.2 INFERENCE RESULTS

Data produced by the generative model was evaluated using the behaviour classification model. The sequence of tracked locations of pirate, transport and fishing vessels are provided as observed variables. The sailing conditions and pirate zone indicator variables are provided as contextual elements. The behavioural classifier is used to infer the hidden state $h_t$, the switching state $s_t$ and the class $c$ for each simulated vessel.

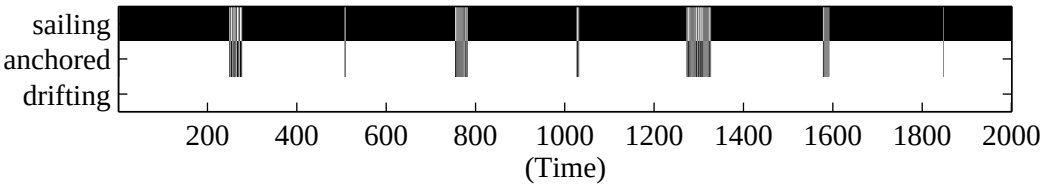
The inferred switching states for a transport vessel are illustrated in Figure 8.2. The presented results demonstrate the transport vessel alternating between the *sailing* and the *anchored* states. A small level of apparent motion is evident in the *anchored* state. This apparent motion is due to the state noise and measurement noise processes. The apparent motion is evident for all vessels in the *anchored* state.

The inferred switching states for a pirate vessel are illustrated in Figure 8.3. The inference method does not infer the *drifting* state completely. This is due to drifting motion that is non-linear and random. Drifting and fishing are modelled by the random process given by (5.16). The proposed model endeavours to model linear motion.

A vessel may change direction at way-points. The change in direction is a non-linear discontinuity in the motion of the vessel. At these discontinuities the LDS model may incorrectly infer the hidden state $h_t$ and the switching state $s_t$. Examples of such events are illustrated in Figure 8.3 at times $t = 121$, $t = 465$ and $t = 1063$. At these times discontinuities occur in linear trend of the motion of the vessel

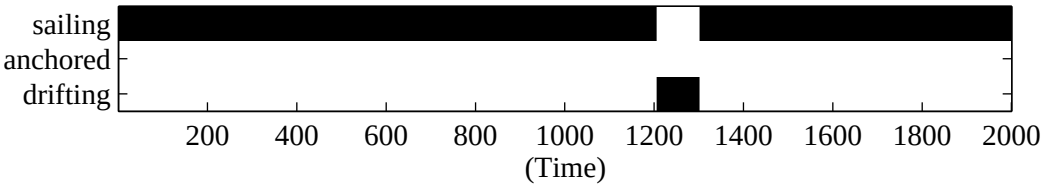



**(a)** True switching states.

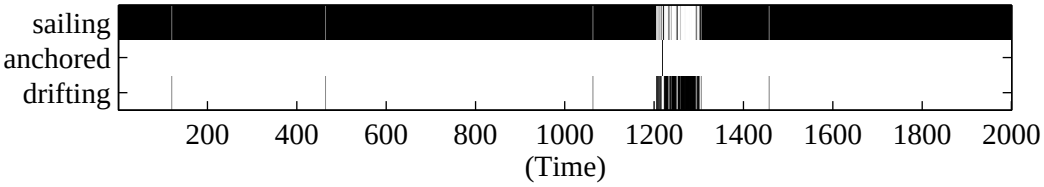



**(b)** Inferred switching states.

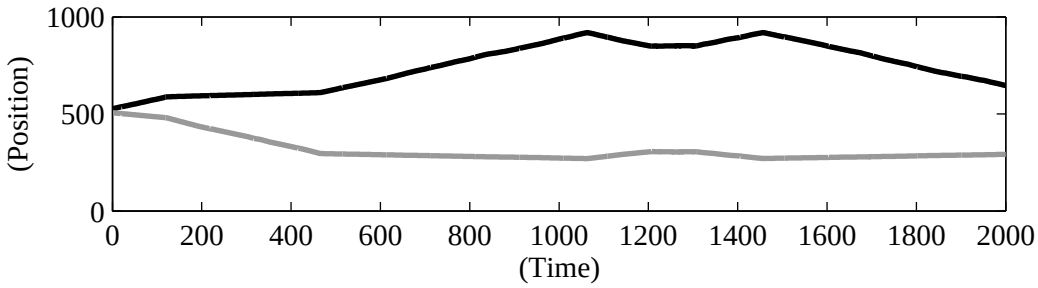
**Figure 8.2:** The inferred switching states for a transport vessel.




**(a)** True switching states.

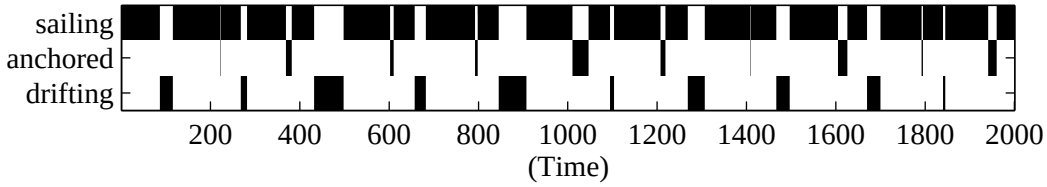



**(b)** Inferred switching states.

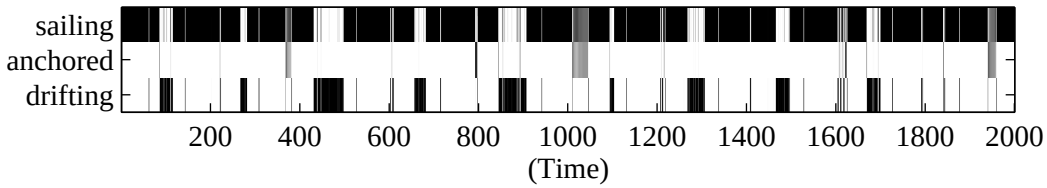



**(c)** True position (black trend: x-position, grey trend: y-position ).

**Figure 8.3:** The inferred switching states for a pirate vessel.




**(a)** True switching states.




**(b)** Inferred switching states.

**Figure 8.4:** The inferred switching states for a fishing vessel.

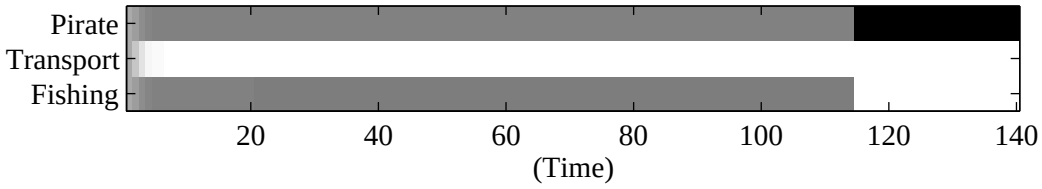



**Figure 8.5:** The inferred class for a pirate vessel for the first 140 time steps. The pirate and fishing vessel classes are equiprobable until the vessel enters a pirate zone at time step 115.

as illustrated in Figure 8.3c. The behavioural classifier model incorrectly infers the switching state $s_t$ as the *drifting* state at these times.

The inferred switching states for a fishing vessel are illustrated in Figure 8.4. This example demonstrates a fishing vessel alternating between the *sailing*, *anchored* and *drifting* (fishing) states.

The inferred class for a pirate vessel for the first 140 time steps are illustrated in Figure 8.5. The model infers equiprobable classes for all classes in the first time step. The model transitions to a more distinctive decision as more data is provided over time. The model transitions to inferring equiprobable classes between pirate and fishing vessels. The vessel sails into a pirate zone at time $t = 115$. The pirate class is inferred for the remaining time step samples after $t = 115$. The pirate class is inferred from this time as fishing vessels avoid the pirate zones.

The inferred class for a transport vessel for the first 40 time steps are illustrated in Figure 8.6. Initially,

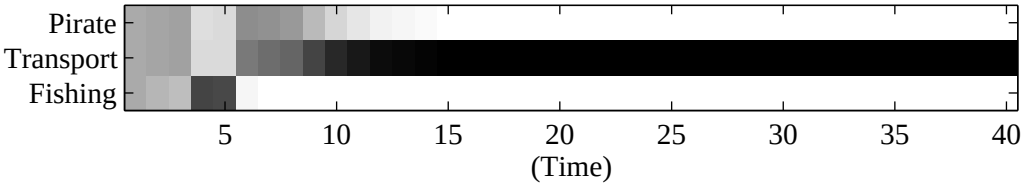



**Figure 8.6:** The inferred class for a transport vessel for the first 40 time steps. As more data is sequentially provided, the classifier increases its belief of the transport class.

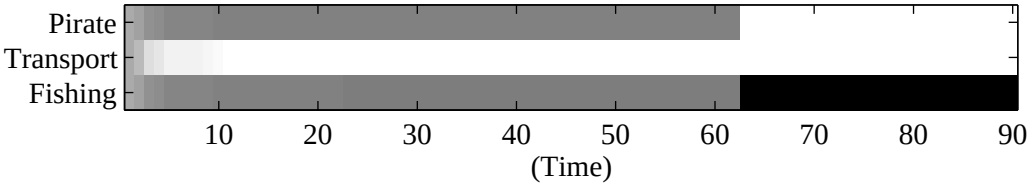



**Figure 8.7:** The inferred class for a fishing vessel for the first 90 time steps. The pirate and fishing vessel classes are initially equiprobable. As the vessel avoids pirate zones, the classifier selects the fishing class.

the model tends towards the inference of the fishing vessel class. The fishing vessel class becomes less probable as the vessel moves into a pirate zone and more data is provided on the vessel speed. The model considers the pirate and transport classes as the vessel sails in the pirate zone. As the vessel sails through the pirate zone, the inferred value in the class variable transitions to the correct class. The transport class is inferred for the remaining duration of the simulation.

The inferred class for a fishing vessel for the first 90 time steps are illustrated in Figure 8.7. The model is unable to distinguish between the fishing and pirate vessel classes. This is due to similar kinematical dynamics of the vessel classes. The model transitions to infer the fishing vessel class as the vessel avoids the pirate zones.

It takes an average of 32 seconds to classify the 2000 time steps of vessel with an Intel i7 core processor in Matlab. This indicates that for many applications, the method may operate on real time data. This however depends on the number of targets monitored at a particular instant and the time difference between time steps.

## 8.3 VARYING SIMULATION PARAMETERS RESULTS

In a further investigation, the model was tested with varying speed features. Various speed ratios between fishing, pirate, and transport vessels were considered. Pirate and fishing vessels travel at the same speed. In each test, the pirate and fishing vessel speed was decreased to a value similar to that of transport vessel speeds. The following three tests were conducted

1. pirate and fishing vessels were configured to travel at speeds 25% faster than transport vessels,
2. pirate and fishing vessels were configured to travel at speeds 10% faster than transport vessels and
3. pirate, fishing and transport vessels were configured to travel at equal speeds.

For each of the three tests, data for set of 1000 vessels was generated. The 1000 vessel simulations consisted of 50 situation simulations comprising of 20 vessels each. Each situation simulation consisted of 2000 time steps. Each time step corresponded to approximately four minutes. The value of 2000 was selected as this provided enough time for all vessels to return to their home location at least once. The behavioural classifier model was utilised to infer the class of each vessel. The classification of a vessel is performed using a thresholding function on the expected value of the class variable over the simulation.

The classification results are described according to recall, precision, F-score and accuracy. Each of these results is computed using the confusion matrix. The confusion matrix $\mathbf{\Lambda} = [\Lambda(j,k)]$, is defined such that an element $\Lambda(j,k)$ contains the number of vessels whose true class label is $j$ and are classified to the inferred class with label $k$ [144]. The rows of the matrix represent the true class. The columns of the matrix represent the inferred class. Pirate vessels are associated with row/column 1. Transport vessels are associated with row/column 2. Fishing vessels are associated with row/column 3. The confusion matrix for the simulation with the speed ratio of 1.25 is given as follows

$$\mathbf{\Lambda} = \begin{bmatrix} 314 & 0 & 0 \\ 5 & 595 & 0 \\ 0 & 0 & 86 \end{bmatrix}. \tag{8.3}$$

**Table 8.5:** Simulated vessel classification recall with varying speed ratios.

| Speed Ratio | $R_{pirate}$ | $R_{transport}$ | $R_{fishing}$ |
|---|---|---|---|
| 1.25 | 100% | 99.2% | 100% |
| 1.10 | 100% | 94.7% | 100% |
| 1.00 | 70.0% | 67.3% | 100% |

The confusion matrix for the results of the simulation with a speed ratio of 1.10 is given as follows

$$\mathbf{\Lambda} = \begin{bmatrix} 297 & 0 & 0 \\ 32 & 575 & 0 \\ 0 & 0 & 96 \end{bmatrix}. \tag{8.4}$$

The confusion matrix for the results of the simulation with a speed ratio of 1.00 is given as follows

$$\mathbf{\Lambda} = \begin{bmatrix} 187 & 80 & 0 \\ 204 & 420 & 0 \\ 0 & 0 & 109 \end{bmatrix}. \tag{8.5}$$

It may be noted that the ratios between the numbers of vessels in each class may not necessarily reflect reality. The results indicate that it is trivial to classify fishing vessels. The classification of transport and pirate vessels is not necessarily as trivial as classifying fishing vessels. The numbers of vessels in each particular class are selected to emphasise the results for the classification of transport and pirate vessels.

The recall (6.32) results are presented in Table 8.5. The recall values for pirate vessels remain relatively high for all the defined speed ratios. A recall value of 100% implies that there were no pirate vessels that were incorrectly classified. In hypothesis testing, the false classification of a pirate vessel would be considered as the false negative hypothesis [145]. The false negative hypothesis is considered to be the most severe hypothesis testing error. The recall values for fishing vessels remains at a value of 100% for all speed ratios. The recall values for the transport class are lower than the values of the other vessel classes. Lower recall values imply that more transport vessels were incorrectly classified. Transport vessels were often incorrectly classified as pirate vessels. This error is known as a false positive error in hypothesis testing. This error is considered to be the least sever hypothesis error.

**Table 8.6:** Simulated vessel classification precision with varying speed ratios.

| Speed Ratio | $P_{pirate}$ | $P_{transport}$ | $P_{fishing}$ |
|---|---|---|---|
| 1.25 | 98.4% | 100% | 100% |
| 1.10 | 90.3% | 100% | 100% |
| 1.00 | 47.8% | 84% | 100% |

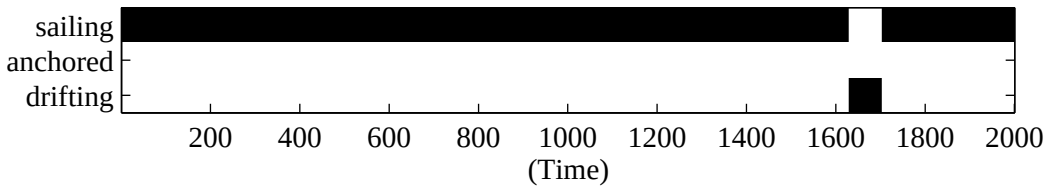
**Table 8.7:** Simulated vessel classification F-score with varying speed ratios.

| Speed Ratio | $F_{pirate}$ | $F_{transport}$ | $F_{fishing}$ |
|---|---|---|---|
| 1.25 | 99.2% | 100% | 100% |
| 1.10 | 94.9% | 100% | 100% |
| 1.00 | 56.8% | 91.3% | 100% |

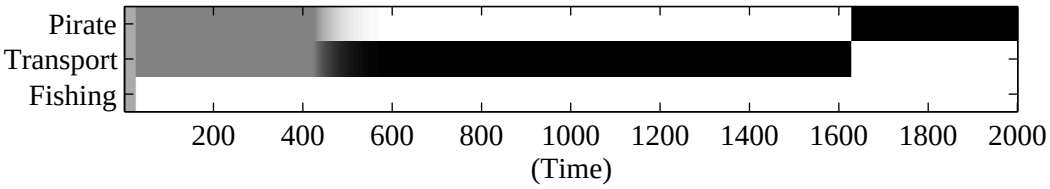
The precision (6.33) is presented in Table 8.6. The precision values are higher than the respective recall values for transport vessels. Higher precision values for the transport class indicate that there were fewer vessels from other classes that were incorrectly classified as transport vessels. The precision values for the pirate vessel class are lower than the corresponding recall values. The lower values are due to the false classification of other vessel classes as pirate vessels. The fishing vessel precision values are 100% for all speed ratios. The F-score (6.34) is presented in Table 8.7 and the overall accuracy (6.35) is presented in Table 8.8. The results presented for the accuracy and F-score describe the classification accuracy. The results indicate that the algorithm accuracy increases as pirate and fishing vessel speed increases with respect to the transport vessel speed. This result is expected. The discriminatory value of the speed feature increases as the pirate and fishing vessel speeds increase with respect to the transport vessel speed. In the case of equal fishing, pirate and transport vessel speeds, the speed feature has no discriminatory value. The switching state is the primary discriminating feature in this case. A vessel in the *sailing* state outside of a pirate zone is ambiguous between classes. A vessel in the *sailing* in a pirate zone is more likely to be of the transport vessel class; especially if the vessel has been in the *sailing* state for a long period of time. A vessel that is in the *drifting* state within a pirate zone is more likely to be of the pirate vessel class. A vessel that is in the *drifting* state outside of a pirate zone is likely to be of the fishing vessel class. The model classification performs is improved when this reasoning is considered. Consider the following example:

**Table 8.8:** Overall classification accuracy with varying speed ratios.

| Speed Ratio | Accuracy |
|---|---|
| 1.25 | 99.5% |
| 1.10 | 96.8% |
| 1.00 | 71.6% |



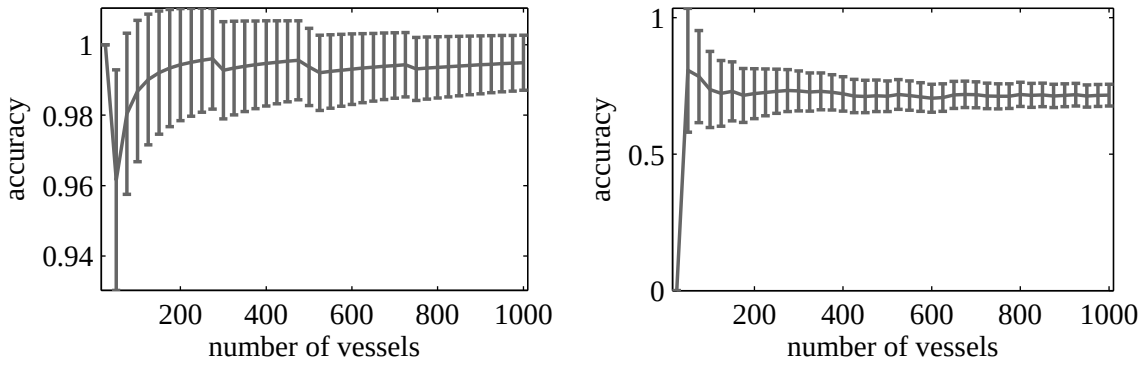

**(a)** True switching states.




**(b)** Inferred class.

**Figure 8.8:** Results for a pirate vessel with its speed equal to that of transport and fishing vessels. The vessel is initially classified as a transport vessel. As soon as the vessel starts to drift, the class is changed to a the correct pirate vessel class.

The switching states and class for a pirate vessel are illustrated in Figure 8.8. All vessels sail at equal speeds in the simulation. For time steps $t = 1$ to $t = 425$ the vessel sails outside of a pirate zone. Initially, the results demonstrate that the classifier considers all vessel classes equiprobable. The fishing vessel is expected to enter a *drifting* state quickly. The probability of the fishing vessel class thus decreases as the vessel continues to sail. The vessel sails in a pirate zone at time $t = 426$. In the pirate zone, the model initially considers the vessel to be either a transport or a pirate vessel. As the vessel continues to sail in the pirate zone, the model transitions to classify the vessel as a transport vessel. At time step $t = 1628$ the vessel enters the *drifting* state. The model changes the class from the transport class to the pirate class at this point. The classification of the vessels behaviour is thus considered to be successful.




**(a)** Accuracy error bar plot for a speed ratio of 1.25.

**(b)** Accuracy error bar plot for a speed ratio of 1.00.

**Figure 8.9:** Error bar plot for the accuracy results for speed ratios of 1.25 and 1.00 between the pirate/fishing vessels and the transport vessels. The trend lines describe the accuracy of per number of vessels considered. The error bars describe the standard deviation of the accuracy per number of vessels.

The time required to classify a vessel according to a specific class depends on the switching state and the vessel speed. The algorithm will classify a vessel in a shorter time period if the speeds between vessel classes differ significantly. A shorter period of time is required to classify a vessel if the transition matrices for the switching states differ significantly between classes. An indication of the effect of time on classification is illustrated in Figure 8.8. The transition probability values for the *sailing* states are similar between vessel classes. Classification of a vessel that is in the *sailing* state requires more time. A vessel that enters the *drifting* state may be classified in a short period of time. Transport vessels do not enter the *drifting* state. Fishing vessels enter the *drifting* state outside of pirate zones. Pirate vessels enter the *drifting* state inside pirate zones.

To consider the statistical significance of the sample size, consider the error bar plots illustrated in Figure 8.9. The trend lines describe the accuracy of per number of vessels considered. The error bars describe the standard deviation of the accuracy per number of vessels. The standard deviation decreases and the trend line settles to a particular value as the number of vessels increase. The law of large numbers indicates that the average accuracy will tend towards the expected accuracy as the number of vessels increase [15]. The settling of the trend lines indicate that the selected number of vessels is sufficient to provide a reasonable approximation of the expected accuracy. This result indicates a form of statistical significance regarding the number of simulated vessels that were selected.

## 8.4 PARAMETER SENSITIVITY

The switching state parameters sensitivity is analysed by varying the switching state transition probabilities for a pirate vessel. The dataset consisting of vessels travelling at equal speeds is used. Only the switching state transition probabilities influence classification in the equal speed dataset. The results are presented in Figure 8.10. In Figure 8.10a, the true switching state transitions are plotted for the vessel. At time $t = 426$, the vessel enters a pirate zone. At time step $t = 1628$ the vessel enters the *drifting* state.

The switching state transition probabilities for the pirate vessel class are set equal to that for the transport vessel class in Figure 8.10b. In this case, the pirate vessel has a zero probability for remaining in a *drifting* state as for the transport vessel class. As the vessel begins to sail, the vessel is classified equally as a pirate, transport or fishing vessel. The fishing vessel is expected to enter a drifting state quickly. As the vessel continues to sail, the fishing vessel class becomes less probable. The vessel is then classified as either a pirate or a transport vessel for the remaining samples. This is expected as the switching state transition probabilities for a pirate and transport vessel are set equal. In Figure 8.10c, the pirate vessel has a non-zero probability for remaining in a *drifting* state. All other switching state transition probabilities remain equal between the pirate and transport vessel classes. As the vessel enters the *drifting* state, the vessel is immediately classified as a pirate vessel.

In Figure 8.10d, the pirate vessel is set to have a 10% higher probability of remaining in the *sailing* state. The state transition probabilities of the pirate and transport vessels are the equal when the vessel is outside of a pirate zone. Thus, the method does not distinguish between the pirate and transport class until the vessel enters a pirate zone. As the vessel sails through the pirate zone, the vessel is classified as a pirate vessel. The pirate vessel class is selected as it has a higher probability of remaining in the *sailing* state than the transport vessel. In Figure 8.10e, the pirate vessel is set to have a 1% higher probability of remaining in the *sailing* state. Comparing this figure to that of Figure 8.10d, the classification seems to take longer to make a decisive decision. This is due to the probability of remaining in the *sailing* state being only 1% higher than the transport vessel rather than 10% higher. The 10% higher probability provides a more discriminative feature. This results a decisive classification in fewer samples.

Figure 8.10f and Figure 8.10g describe the opposite scenario to Figure 8.10d and Figure 8.10e. In these figures, the transport vessel has a 10% and 1% higher probability of remaining in the *sailing*

state respectively. As indicated, the transport class is inferred as the vessel continues to sail through the pirate zone. However, as the vessel enters the *drifting* state, the pirate class is inferred.

## 8.5 MODEL EFFECTIVENESS

The quality of the inference method measures the performance of the model [138]. The performance of the model may be described according to the precision, recall, accuracy and F-score results described in section 8.3. The results presented demonstrate that the behaviour classifier method is successful in classification. The method especially demonstrated high accuracy results with higher speed ratios between vessel classes.

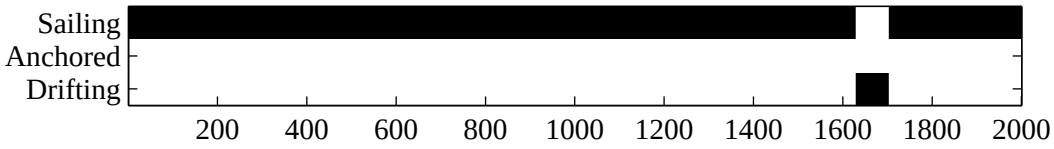
Robustness measures the consistency of the model [138]. The robustness of the model may be described by its consistency in the ability to handle noise. Examples of noise in the model are discussed in section 8.2. These include the state noise process and the measurement noise, the motion discontinuities and the drifting state motion. The method demonstrated the ability to handle noise in the various circumstances associated with these forms of noise.

The robustness of the model may also be described by the types of errors produced by the method. The false negative error is the most severe error type. A false negative error is the error of classifying a pirate vessel as another vessel class. The False negative error has a high risk associated with it. A vessel that is not warned of an eminent attack is in danger. The recall values indicate the number of false negative errors. The recall values for the various vessel classes are provided in table 8.5. The high recall values for pirate vessels indicate few false negative errors. The low percentage of false negative errors indicates a higher level of robustness of the method.

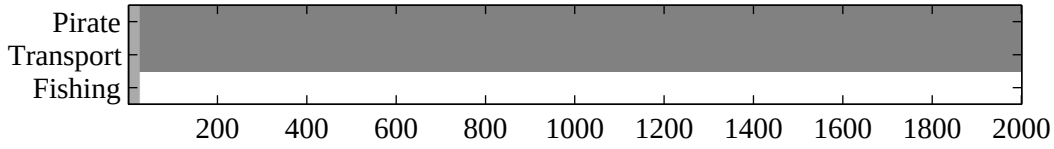
The generative model and the behavioural classification model differ in structure. A form of robustness to model mismatch is indicated by the extent to which the method is able to perform classification.

The method has the tenancy to change the inferred class even if it has indicated a high probability for another class. This is potentially a weakness of the method. This weakness can be managed by varying the transition probabilities in the switching state transition matrices.

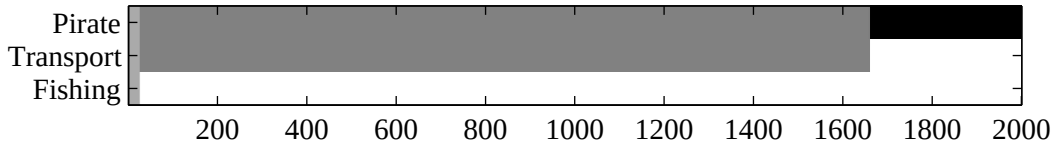
An information data source may be evaluated according to information gain [138, 40]. Information gain provides a measure of the contribution that particular data source offers. Cross entropy is pro-



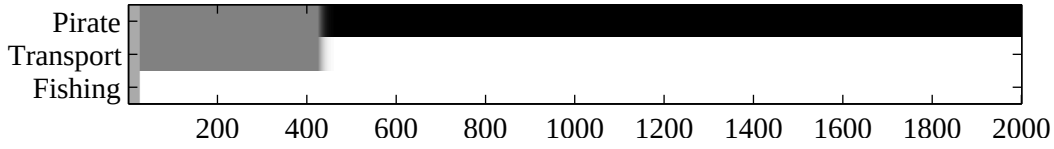

**(a)** True switching states.



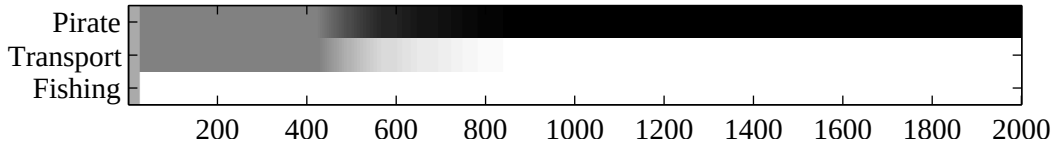

**(b)** Inferred class: Pirate and transport vessel state transition matrix equal.



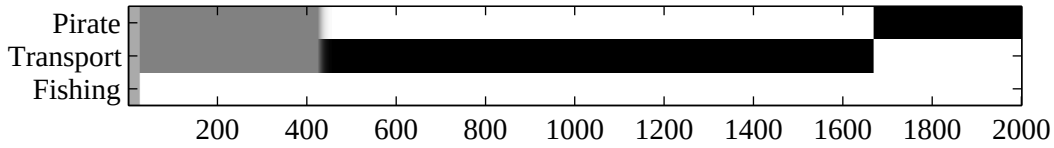

**(c)** Inferred class: Pirate vessel with non-zero probability for entering drifting state



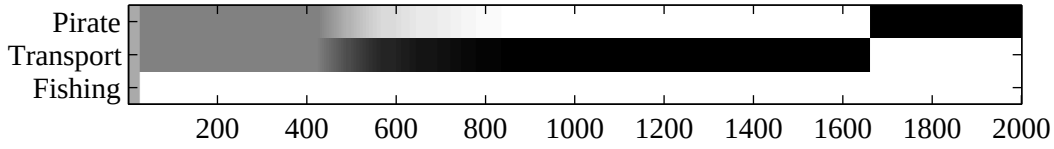

**(d)** Inferred class: Pirate vessel with 10% higher probability of remaining in the sailing state.




**(e)** Inferred class: Pirate vessel with 1% higher probability of remaining in the sailing state.




**(f)** Inferred class: Pirate vessel with 10% lower probability of remaining in the sailing state.




**(g)** Inferred class: Pirate vessel with 1% lower probability of remaining in the sailing state.

**Figure 8.10:** Parameter sensitivity plots.

posed to be utilised as a measure for information gain. Information gain may be utilised to select external factors for the inference model. External factors that contribute to higher information gain are selected. External factors that do not contribute to higher information gain are rejected. This process is essentially a form of feature selection.

## 8.6 CLOSING REMARKS

The results obtained in this study are presented in Chapter 7 and Chapter 8. The results demonstrate the application of the proposed methods to generate and classify behavioural data. The application pertains to a maritime piracy situation. In chapter 9, a general methodology of applying the proposed methods to various other applications is presented.

# CHAPTER 9

# ADDITIONAL APPLICATIONS

Two methods have been presented and applied to the maritime piracy problem. The first method is a generative model that generates synthetic maritime data in a piracy situation. The second method is an inference and classification method that infers the class of a maritime vessel. The two discussed methods are not intended to be constrained to the maritime piracy application. Various other applications and problems may be addressed with these methods. Simulation and monitoring may be applicable in the following domains:

1. Maritime ports
2. Public airfields
3. Military airspace
4. Public roads and traffic
5. Humans in various environments (such as airports and shopping centres)
6. Financial markets
7. Fraudulent activities
8. Computer networks
9. Illegal immigration
10. Poaching

In this chapter, the generative model and the classification and inference models are generalised. The generalisation is intended to provide a framework for applying the methods to a variety of problems and domains.

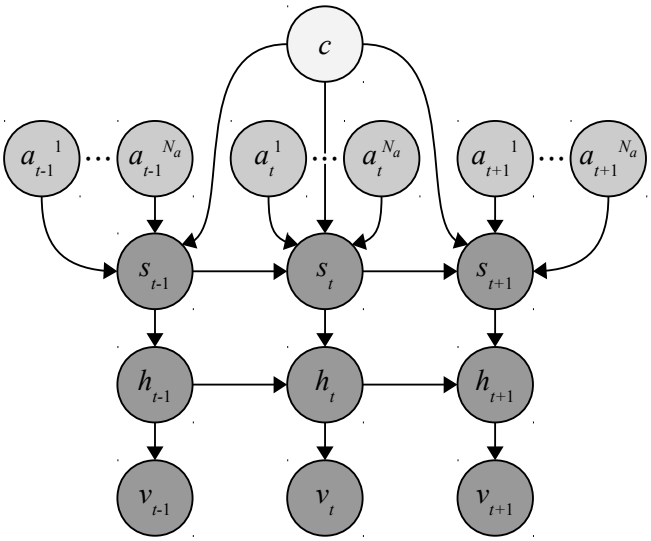



**Figure 9.1:** Generalised generative and classification model.

## 9.1 GENERALISED MODEL

The general model is illustrated in Figure 9.1. This model is identical to the inference model presented in Figure 6.1. The model consists of four layers. The first layer consists of the classification node $c$. The second layer consists of the set of $N$ contextual element nodes $a_t^n$, $n = \{1,...,N\}$. The third layer consists of the switching state $s_t$. The fourth layer is the linear dynamic system. The LDS consists of the visible state variable $v_t$ and the hidden state variable $h_t$.

The model functions as a generative model when samples are generated using the ancestral sampling method. The ancestral sampling method is described in section 2.2.5. Ancestral sampling will sample from the DBN from the class variable down to the visible state variables $v_t$. The sampling is thus performed in a top-down manner. For ancestral sampling the conditional probability distributions between variables are assumed to be known.

The model functions as a classifier when the class variable is inferred. For inference, the conditional probability distributions between variables are assumed to be known. Furthermore, the visible state variables $v_t$ and the contextual element nodes $a_t^n$ may be observed. Given the observed variables (evidence) the class may be inferred. Inference is thus performed in a bottom-up manner.

## 9.2 THE CLASS LAYER

The various entities that are modelled by the system are defined in the class layer. The entities are defined according to the problem that is required to be solved. For example, in a public road domain where traffic is monitored, various classes may be formulated. A problem may exist where various forms of vehicles are to be classified. These may include cars, trucks, motorcycles, bicycles and pedestrians. Alternatively, the problem may be to identify illegal behaviour. In this case, the classes may either be legal or illegal behaviour.

In the generative model case, the prior probability of each class is to be defined. In sampling, a particular class will be selected according to the prior probability. In a simulation, a set entities are defined. The distribution of the entity classes is defined by the prior probability. For example, if a statistic indicates that ten percent of vehicles drive illegally, then in a sample of 1000 simulated vehicles, perhaps 100 of those simulated vehicles will demonstrate illegal behaviour.

In the classification case, the class is inferred. More concretely, the probability that the entity belongs to a specific class is inferred, given the provided evidence. As time progresses, more information will be provided. With more information, the model is able to provide a more accurate inference.

## 9.3 THE CONTEXTUAL ELEMENTS LAYER

Contextual elements provide the means to model behaviour according to external parameters. The contextual elements are defined as elements that influence an entities behaviour. Behaviour may be constrained or promoted under certain contextual element conditions. For example, the environment may constrain certain behaviour to occur in particular locations. Various political situations may promote certain behaviour. In the aviation domain, examples of contextual elements may include time, location, weather, flight path, other aircraft location, other aircraft behaviour and airfield locations.

In both the generative model and the classification cases, the contextual elements are assumed to be observable. The observations may be results of either real-world data or modelled data. The contextual elements do not necessarily have to be observable. It is possible to perform inference on these variables using the GSF algorithm. The contextual elements are required to be represented by discrete distributions.

## 9.4 THE SWITCHING STATE LAYER

The switching state layer defines the behaviour of the various entities that are modelled. A switching state is identified for each behavioural expression. The switching states are linked according to state transition diagrams such as those illustrated in Figure 5.4. The state transition diagrams constrain which behavioural states may lead to others. Probabilities are to be associated with each state transition. A particular defined state may transition to another defined state according to the given predefined probability.

The switching states are influenced by the class variable and the contextual elements. Certain states and state transition probabilities are associated with each class. Some switching states may be excluded from certain classes and others may be shared between classes. As illustrated in Figure 5.4, transport vessels exclude the *drift* state. Pirate, transport and fishing vessels share the *anchor* and *sail-out* states. External factors may influence state transition probabilities. For example, the pirate vessel transitions to the *drift* state when it is in a pirate zone.

As an example, consider the problem of classifying human behaviour in a public environment. The switching states may include walking, loitering, running and weaving. Given such states, suspicious behaviour may be identified. Loitering may be associated with the suspicious behaviour class. An external factor such as the presence of an authority figure may trigger certain types of behaviour.

## 9.5 THE LDS LAYER

The linear dynamic system layer provides the model that describes the parameters that are observed. When performing inference, the observed parameters are tracked. In many applications, the LDS describes the motion model. When performing inference, the location of a target may be tracked. The LDS is however not constrained to modelling motion. In the financial stock market domain, the LDS may be applied to model the stock of a particular company. The stock price may be tracked when performing inference.

In the generative model, the observations of the visible state variable $v_t$ provides the simulated data. In the stock market, the class variable may consist of various companies. A generated sample at a particular time may be a stock price provided by the visible state variable $v_t$.

In the classification model, the visible state variable $v_t$ is given. The data provided for this variable may be obtained from real-world data or simulated data. In the stock market application, real-world data consisting of stock prices for various companies over time may be available. The model may be used to classify the company for investment purposes. The company may be classified as either a low risk, a medium risk or a high risk investment.

## 9.6 MODIFYING THE MODEL STRUCTURE

The model presented in Figure 9.1 is a generalised model. The structure of this model may be adjusted or modified. In this study, the generalised model was modified for the synthetic data generation application. The modifications include three additional nodes. The $JP$ variable, the $OW$ variable and the $u_t$ variables are included in the model. The $JP$ (journey parameter variable) is provided to restrict the paths of vessels. It may be argued that this variable takes the form of an external parameter. The $OW$ (ocean/weather) variable is provided to directly influence the LDS of a vessel. The $u_t$ (control) variable is included to provide control of the maritime vessels. The additional variables do not change the model itself. Both of these variables may be considered to form part of the LDS layer of the model. It should however be noted that, the DBN suffers from the curse of dimensionality. Increasing the number of variables may significantly increase the model complexity.

## 9.7 LEARNING PARAMETERS

In the piracy application, the switching state variable is defined according to expert knowledge. Learning the parameters of the switching state variable from data is not possible as such data is not available. Applications may exist where data for learning the switching state parameters may be available. An example of such an application may be human surveillance. Data may exist of humans engaging in various activities such as walking, running and loitering. Learning the parameters associated with these states may be performed by considering data of humans engaging in each of these activities. In applications relating to financial stock markets and fraud detection, the data may be available. However, the switching states may not be obvious. For example, changes in volatility of a stock index are not obvious.

A particularly interesting emerging approach to dealing with such problems is Bayesian nonparametrics [151]. Bayesian nonparametric methods provide the means for learning and inference in complex dynamical models [152]. The approach is to provide a probabilistic model of the dynamic model itself

[153]. The hierarchical Bayesian nonparametric model allows for infinite-dimensional parameters to be represented hierarchically. This provides the means to represent models that require large degrees of freedom. Furthermore, flexibility is provided for additional parameters or parameter states to be incorporated dynamically [152].

The Dirichlet process (DP) is used for clustering where the number of clusters is unknown [154]. This is commonly formulated as a Dirichlet process mixture model. The hierarchical Dirichlet process (HDP) extends the DP for cases where clusters of data are produced by distinct related generative processes. This is particularly useful in the case where the switching states are not obvious.

In the application by Teh and Jordan [153], the hierarchical Dirichlet process is used to construct a HMM in which the cardinality of the state is infinite [154]. That is, the HDP-HMM allows for state spaces of unknown sizes to be learned from data. The sticky extension has been suggested to provide a more robust learning of dynamics that vary smoothly [155]. The method is applied to learn complex emission distributions in a speaker diarisation application. The task involves the segmentation of an audio recording into speaker-homogeneous sections. The method additionally determines the number of speakers in the recording.

The HDP has been applied for tracking manoeuvring targets [156]. The state estimation of a dynamic system is performed. A set of unknown correlated manoeuvre modes are learned from data. The Markov chain Monte Carlo method is used for learning. Learning involves the adaptation of the HDP to discover an appropriate set of manoeuvre modes.

The HDP is applied to learn the set of switching states in the SLDS [157]. The sticky HDP HMM learning method is extended for the SLDS and the switching vector autoregressive (VAR) process. The method is successfully applied to sequences of dancing honey bees and the IBOVESPA stock index. In future research, this method may be applied to learn the switching states in the model presented in Figure 9.1.

# CHAPTER 10

# SUMMARY AND CONCLUSION

IN CONCLUSION, the results of the research are considered according to the research question and research objectives. The research question considered three facets, namely what describes maritime pirate vessel behaviour, how can this behaviour be modelled and how can this behavioural model be used to identify pirate vessels. The output of this study is a practical demonstration that intends to answer the research question and fulfil the research objectives.

## 10.1 SUMMARY

A literature survey is presented. Background and related work are discussed in the survey as to place this study in the context of literature. Various applications and fields that relate to the application of this study are considered. Methodologies and approaches to solving similar problems are mentioned. This discussion follows into a chapter on the theoretical foundations of this research. A presentation on probability theory and graphical models is presented in the latter chapter. The theory and tools that are used in the remainder of the study are summarised.

A chapter on model informing prior knowledge is provided. In this chapter, maritime piracy is discussed. In the model informing prior knowledge chapter, an extensive discussion that includes pirate appearance, kinematic characteristics, meteorological and oceanographic influences is provided. The discussion provides insight into maritime pirate behaviour and factors that influence pirate behaviour. A set of attributes that describe maritime piracy behaviour are easily extracted from this discussion.

A multi-agent generative model for simulating a maritime piracy situation is proposed. The generative model models the behaviour of various maritime vessel types. A model of maritime pirate vessels is

included in the generative model. The model is in the form of a DBN that extends the SLDS. The DBN may be considered as a Markov model that encodes the behaviour of maritime vessels. A means to include behaviour influencing external factors is included in the model. Data from the external factors are directly input into the model. Through ancestral sampling, a set of vessels are simulated in an environment. The output of the sampling process is a set of data that may be considered as data sampled from simulated sensors. The proposed DBN is as a versatile model that unifies probabilistic, graphical and physical attributes to model behaviour.

Reported pirate attack locations from 2011 are used to construct and evaluate the generative model. The likelihood of the real-world pirate attack data given the simulated data is used for optimisation and evaluation. The model effectiveness is measured according to quality, robustness and information gain. The likelihood and Bhattacharyya distance are utilised for evaluating the quality of the model. A description of the consistency of the model is used to provide an indication of the robustness of the model. The information gain is measured using an approximation of the cross entropy. The cross validation method and Bayes factor are used to measure how well the model generalises the 2011 pirate attack dataset. The results demonstrate that the model is highly effective in modelling 2011 pirate attack data in the spatial domain.

The temporal modelling ability of the model is evaluated. The accumulated pirate attacks over each month are compared to the 2011 IMB pirate attack dataset. The Bhattacharyya is applied to evaluate the results. The results demonstrate a high level of effectiveness in modelling temporal characteristics.

The data produced by the generative model may be utilised in various fields and applications. For example, the generative model may be used in a multi-sensor simulation application. A plate of sensor variables is contained in the generative model. A set of particular sensors may be modelled by the set of sensor plates. The data generated by the simulated sensors could be used for evaluating and testing multi-sensor information fusion methods. The generative model is not restricted to the maritime piracy application. The model may easily be adapted for other applications such as air or land based applications.

In this study, the data produced by the generative model is used in testing and evaluating a behavioural classification method. A novel behavioural model for classifying dynamic systems is proposed. The model may be considered as a generalisation of the generative model. The model is a DBN that

consists of an SLDS, contextual element nodes and a class node. Data is provided to the observation or measurement nodes of the SLDS and the external element nodes. The GSF algorithm is applied for inferring the unobservable nodes in the DBN. The switching node of the SLDS is inferred according to a set of kinematic behaviours. A vessel class is inferred according to a set of types. The inferred vessel class describes the probability of vessel class given the sensor data and the contextual element data.

The behavioural classification method is applied to classify fishing vessels, transport vessels and pirate vessels in a maritime piracy situation. The ground truth and data are provided by the generative model. The behavioural classification method is evaluated according to the precision, recall, F-score and accuracy measurements. The method demonstrated a high level of success in classification.

The proposed model may be applied to various problems in a wide range of applications. Examples of applications include criminal behaviour detection in public areas, terrorist behaviour detection in aircraft surveillance and animal behaviour classification. Fields of application may include econometrics, computer vision signal processing and robotics.

## 10.2 CONCLUSIVE REMARKS

The first research objective requires a set of attributes that describe maritime piracy behaviour. Chapter 4 provides an in depth description of the various attributes of maritime piracy behaviour. These attributes form prior information for constructing the generative and classification models presented in the remaining study. The second research objective requires a generative model that is able to generate vessel behaviour data in a pirate situation. The generative model fulfils the second research objective. The behavioural classification DBN and its associated inference method fulfil the classification method required by the third research objective.

In the first research hypothesis, it is postulated that a DBN may be formulated to represent kinematic behaviour of maritime vessels. It is postulated in the second research hypothesis, that a vessel class may be inferred using a DBN behavioural model. The results presented in this study satisfy these hypotheses. A DBN behavioural model is constructed and applied to both generate and classify maritime vessel behaviour.

The generative model is demonstrated to have the ability to produce unique and varying results while

preserving the structural integrity of a spatial or temporal distribution. This property is valuable when the data is utilised in testing, evaluating and training machine learning, statistical or other models and methods. The behavioural classification method is an effective and efficient method. The method provides a closed solution that converts low level temporal data into behavioural attributes and performs classification based on these attributes. The method is applicable to many domains and problems and could potentially operate in various real-time environments.

## 10.3 FUTURE RESEARCH

The generative model may to be refined according to real world data and statistics. Parameters such as wind speed, wave height, wind direction, air temperature and cloud cover may be included in the EP variable.

Alternative algorithms may be investigated and applied for the parametrisation of the generative model. A brute force and gradient descent algorithm are suggested in the study. More efficient algorithms may exists, such as those described in Section 9.7.

The behavioural classifier method may be extended to include various additional external parameters; such as those recommended for the EP variable. The smoothing operation rather than the filtering operation may be applied for the inference. It is expected that the smoothing operation will significantly improve the results described in Section 8.3. The filtering operation considers data in a sequential manner. The entire dataset is considered by the smoothing operation at any particular point in time.

Scenarios such as those illustrated in Figure 8.8 are expected to benefit from the smoothing operation. At time $t = 1$, the smoothing operation will consider that the vessel entered the *drifting* state at time $t = 1628$. The pirate class may thus be inferred at time $t = 1$.

The Gaussian sum smoothing (GSS) algorithm requires several approximations to be made for the SLDS application [18]. The approximations result in the algorithm being more prone to failure than the GSF algorithm. Additional approximations will be required when applying the algorithm to the behavioural classification model. It is expected that this will result in further instability of the GSS algorithm. A more suitable smoothing algorithm may to be determined in future research.

The generative model was evaluated on real-world piracy data as far as possible. The classification

model was not evaluated on real-world data as this data is not available. In future work, both models may be applied to other case studies in which real-world data exists. Examples include the financial application and video surveillance of people.

# APPENDIX A

# LEMMAS

**Lemma 1** Let $y$ be related to $x$ through $y = Mx + \eta$, where $x \perp\!\!\!\perp \eta$, $\eta \sim \mathcal{N}(\mu, \Sigma)$, and $x \sim \mathcal{N}(\mu_x, \Sigma_x)$. Then the marginal $p(y) = \int_x p(y|x)p(x)$ is a Gaussian given by [18]

$$p(y) = \mathcal{N}(y|M\mu_x + \mu, M\Sigma_x M^T + \Sigma).$$

**Lemma 2** Consider a Gaussian distribution parameterised by $\mu$ and $\Sigma$, defined jointly over two vectors $x$ and $y$ of potentially differing dimensions,

$$z = \begin{pmatrix} x \\ y \end{pmatrix},$$

with corresponding mean and partitioned covariance

$$\mu = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix},$$

where $\Sigma_{yx} = \Sigma_{xy}^T$. The marginal distribution is given by [18]

$$p(x) = \mathcal{N}(x|\mu_x, \Sigma_{xx})$$

and conditional

$$p(x|y) = \mathcal{N}(x|\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}).$$

# REFERENCES

[1] R. Middleton, "Piracy in somalia: Threatening global trade, feeding local wars," Chatham House, October 2008, briefing Paper. (7 citations on pages 1, 30, 39, 40, and 55.)

[2] ICC-IMB, "ICC-IMB piracy and armed robbery against ships report - annual report 2011," ICC International Maritime Bureau, Annual Report, January 2012. (17 citations on pages 1, 2, 29, 31, 32, 41, 42, 53, 59, 83, 101, 109, and 111.)

[3] A. Bowden and S. Basnet, "The economic cost of somali piracy, 2011," Oceans Beyond Piracy, One Earth Future Foundation, Tech. Rep., 2012. (1 citation on page 1.)

[4] K. Hurlburt and C. Mody, "The human cost of somali piracy, 2011," International Maritime Bureau and Oceans Beyond Piracy, Tech. Rep., 2012. (2 citations on page 1.)

[5] C. Bueger, J. Stockbruegger, and S. Werthes, "Pirates, fishermen and peacebuilding: Options for counter-piracy strategy in somalia," *Contemporary Security Policy*, vol. 32, no. 2, pp. 356–381, 2011. (2 citations on pages 1 and 31.)

[6] M. Heger, J. Oberg, M. Dumiak, S. Moore, and P. Patel-Predd, "Technology vs. pirates," *Spectrum, IEEE*, vol. 46, no. 2, pp. 9–10, 2009. (3 citations on pages 1, 41, and 108.)

[7] M. Jakob, O. Vaněk, and M. Pěchouček, "Using agents to improve international maritime transport security," *Intelligent Systems, IEEE*, vol. 26, no. 1, pp. 90–96, 2011. (4 citations on pages 2, 36, 54, and 55.)

[8] S. Kazemi, S. Abghari, N. Lavesson, H. Johnson, and P. Ryman, "Open data for anomaly detection in maritime surveillance," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5719 – 5729, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/

S0957417413002765 (1 citation on page 2.)

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. (1 citation on page 3.)

[10] W. Bolstad, *Understanding Computational Bayesian Statistics*, ser. Wiley Series in Computational Statistics. Wiley, 2011. (2 citations on pages 8 and 9.)

[11] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Pacific Grove, CA 93950 USA: Duxbury, 2002. (7 citations on pages 9, 10, 13, 14, and 16.)

[12] P. Z. Peebles, *Probability, Random Variables and Random Signal Principles*, 4th ed. New York, USA: McGraw-Hill, 2001. (13 citations on pages 9, 10, 12, 13, 14, and 16.)

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. 233 Spring Street, New York, NY 10013, USA: Springer Science, 2006. (8 citations on pages 10, 12, 14, 18, 24, and 44.)

[14] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, ser. Adaptive Computation and Machine Learning. MIT Press, 2009. (11 citations on pages 16, 19, 20, 21, 23, 24, 26, and 31.)

[15] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, 3rd ed. Great Clarendon Street, Oxford OX2 6DP: Oxford University Press, 2001. (4 citations on pages 19 and 121.)

[16] J. F. Verner and T. D. Nielsen, *Bayesian Networks and Decision Graphs*, ser. Information Science and Statistics. Springer, 2007. (3 citations on pages 21, 22, and 31.)

[17] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition." *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. (1 citation on page 24.)

[18] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012. (27 citations on pages 24, 25, 26, 27, 31, 32, 33, 63, 64, 69, 70, 71, 74, 75, 96, 135, and 137.)

[19] B. Mesot, "Inference in switching linear dynamical systems applied to noise robust speech recognition of isolated digits," Ph.D. dissertation, STI, Lausanne, 2008. (4 citations on pages

25, 32, and 33.)

[20] S. Russell and P. Norvig, *Artificial Intelligence A Modern Approach*, 3rd ed. Upper Saddle River, New Jearsey 07458: Pearson, 2010. (7 citations on pages 26, 51, 52, and 63.)

[21] IMO, "Piracy and armed robbery against ships: Guidance to shipowners and ship operators, shipmasters and crews on preventing and suppressing acts of piracy and armed robbery against ships," International Maritime Organization, Tech. Rep., June 2009. (2 citations on pages 29 and 37.)

[22] B. White and K. Wydajewski, "Commercial ship self defense against piracy and maritime terrorism," in *OCEANS '02 MTS/IEEE*, vol. 2, 2002, pp. 1164–1171 vol.2. (7 citations on pages 30, 37, 39, 41, 53, and 55.)

[23] K. Menkhaus, "Dangerous waters," *Survival: Global Politics and Strategy*, vol. 51, no. 1, pp. 21–25, 2009. (5 citations on pages 30, 39, 40, and 41.)

[24] P. Chalk, "Maritime piracy: Reasons, dangers and solutions," DTIC Document, February 2009. (1 citation on page 30.)

[25] ——, "Contemporary maritime piracy in southeast asia," *Studies in Conflict & Terrorism*, vol. 21, no. 1, pp. 87–112, 1998. (3 citations on pages 30 and 40.)

[26] H. Jesus, "Protection of foreign ships against piracy and terrorism at sea: legal aspects," *The International Journal of Marine and Coastal Law*, vol. 18, no. 3, pp. 363–400, 2003. (1 citation on page 30.)

[27] P. Birnie, "Piracy: Past, present and future," *Marine Policy*, vol. 11, no. 3, pp. 163 – 183, 1987. (1 citation on page 30.)

[28] G. Teitler, "Piracy in southeast asia: A historical comparison," *Maritime Studies*, vol. 1, no. 1, pp. 67–83, 2002. (1 citation on page 30.)

[29] B. Hughes and S. Jones, "Convoys to combat somali piracy," *Small Wars & Insurgencies*, vol. 23, no. 1, pp. 74–92, March 2012. (1 citation on page 30.)

[30] J. Kraska and B. Wilson, "Fighting pirates: The pen and the sword," *World Policy Journal*,

vol. 25, no. 4, pp. 41–52, 2008. (2 citations on pages 30 and 40.)

[31] J. Abbot and N. Renwick, "Pirates? maritime piracy and societal security in southeast asia," *Pacifica Review: Peace, Security & Global Change*, vol. 11, no. 1, pp. 7–24, 1999. (1 citation on page 31.)

[32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, ser. Representation and Reasoning Series. Morgan Kaufman, 1988. (1 citation on page 31.)

[33] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Computational Intelligence*, vol. 5, no. 2, pp. 142Ű–150, February 1989. (2 citations on page 31.)

[34] K. P. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, July 2002. (7 citations on pages 31, 32, 33, 70, and 71.)

[35] Y. Luo, T.-D. Wu, and J.-N. Hwang, "Object-based analysis and interpretation of human motion in sports video sequences by dynamic bayesian networks," *Computer Vision and Image Understanding*, vol. 92, no. 2-3, pp. 196 – 216, 2003, special Issue on Video Retrieval and Summarization. (1 citation on page 32.)

[36] S. O. nón, J. L. M. na, and A. J. Gonzalez, "A dynamic-bayesian network framework for modeling and evaluating learning from observation," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5212 – 5226, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417414001262 (1 citation on page 32.)

[37] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009. (1 citation on page 32.)

[38] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian fault detection and diagnosis in dynamic systems," in *AAAI/IAAI*, 2000, pp. 531–537. (1 citation on page 32.)

[39] P. Wiggers, B. Mertens, and L. Rothkrantz, "Dynamic bayesian networks for situational awareness in the presence of noisy data," in *Proceedings of the 12th International Conference on Computer Systems and Technologies*, ser. CompSysTech '11. New York, NY, USA: ACM,

2011, pp. 411–416. (1 citation on page 32.)

[40] S. Das, *High-Level Data Fusion*, ser. Artech House electronic warfare library. Artech House, Incorporated, 2008. (3 citations on pages 32, 34, and 123.)

[41] J. Poropudas and K. Virtanen, "Influence diagrams in analysis of discrete event simulation data," in *Winter Simulation Conference*, ser. WSC '09. Winter Simulation Conference, 2009, pp. 696–708. (1 citation on page 32.)

[42] ——, "Simulation metamodeling in continuous time using dynamic bayesian networks," in *Proceedings of the Winter Simulation Conference*, ser. WSC '10. Winter Simulation Conference, 2010, pp. 935–946. (1 citation on page 32.)

[43] ——, "Simulation metamodeling with dynamic bayesian networks," *European Journal of Operational Research*, vol. 214, no. 3, pp. 644 – 655, 2011. (1 citation on page 32.)

[44] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 1st ed., ser. Intelligent robotics and autonomous agents series. United States of America: MIT Press, 2006. (2 citations on pages 32 and 45.)

[45] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques, and Software*, ser. The Artech House radar library. Artech House, Incorporated, 1993. (2 citations on pages 32 and 33.)

[46] C.-J. Kim, "Dynamic linear models with markov-switching," *Journal of Econometrics*, vol. 60, no. 1-2, pp. 1 – 22, 1994. (1 citation on page 32.)

[47] V. Pavlovic, J. Rehg, T.-J. Cham, and K. Murphy, "A dynamic bayesian network approach to figure tracking using learned dynamic models," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, 1999, pp. 94–101 vol.1. (2 citations on pages 32 and 33.)

[48] B. Mesot and D. Barber, "Switching linear dynamical systems for noise robust speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 6, pp. 1850–1858, 2007. (1 citation on page 32.)

[49] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960. (1 citation on page 33.)

[50] D. Alspach and H. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *Automatic Control, IEEE Transactions on*, vol. 17, no. 4, pp. 439–448, 1972. (1 citation on page 33.)

[51] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 33–42. (1 citation on page 33.)

[52] T. P. Minka, "A family of algorithms for approximate bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001. (1 citation on page 33.)

[53] V. Pavlovic and J. Rehg, "Impact of dynamic model learning on classification of human motion," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 1, 2000, pp. 788–795 vol.1. (1 citation on page 33.)

[54] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural Computation*, vol. 12, no. 4, pp. 831–864, 2000. (3 citations on page 33.)

[55] D. Barber, "Expectation correction for smoothed inference in switching linear dynamical systems," *The Journal of Machine Learning Research*, vol. 7, pp. 2515–2540, 2006. (3 citations on pages 33, 71, and 76.)

[56] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 176–183. [Online]. Available: http://0-dl.acm.org.innopac.up.ac.za/citation.cfm?id=2073946.2073968 (1 citation on page 33.)

[57] R. Van Der Merwe, "Sigma-point kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, University of Stellenbosch, 2004. (1 citation on page 33.)

[58] U. N. Lerner, "Hybrid bayesian networks for reasoning about complex systems," Ph.D. disser-

tation, Stanford University, October 2002. (1 citation on page 33.)

[59] M. Wooldridge, *An Introduction to Multiagent Systems*. John Wiley and Sons, Ltd, 2008. (1 citation on page 34.)

[60] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press, 2009. (1 citation on page 34.)

[61] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," in *Proceedings of the 37th conference on Winter simulation*, ser. WSC '05. Winter Simulation Conference, 2005, pp. 2–15. (2 citations on page 34.)

[62] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, ser. Intelligent Robotics and Autonomous Agents Series. The MIT Press, 1999. (1 citation on page 34.)

[63] T. Cioppa, T. Lucas, and S. Sanchez, "Military applications of agent-based simulations," in *Simulation Conference, 2004. Proceedings of the 2004 Winter*, vol. 1, 2004, pp. –180. (1 citation on page 34.)

[64] S. M. Sanchez and T. W. Lucas, "Exploring the world of agent-based simulations: simple models, complex analyses," in *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, ser. WSC '02. Winter Simulation Conference, 2002, pp. 116–126. (1 citation on page 34.)

[65] M. E. Liggins, D. L. Hall, and J. Llinas, *Handbook of Multisensor Data Fusion Theory and Practice*, 2nd ed. 6000 Broken Sound Parkway NW, Suite 300: CRC Press, 2009. (1 citation on page 34.)

[66] H. Wehn, R. Yates, P. Valin, A. Guitouni, E. Bosse, A. Dlugan, and H. Zwick, "A distributed information fusion testbed for coastal surveillance," in *Information Fusion, 2007 10th International Conference on*, 2007, pp. 1–7. (1 citation on page 34.)

[67] P. C. G. Costa, K. B. Laskey, K.-C. Chang, W. Sun, C. Y. Park, , and S. Matsumoto, "High-level information fusion with bayesian semantics," in *UAI 9th Bayesian Modeling Applications Workshop*, Catalina Island, CA, August 2012, pp. –, held at the Conference of Uncertainty in

Artificial Intelligence (BMAW UAI 2012). (1 citation on page 34.)

[68] R. Carvalho, R. Haberlin, P. Costa, K. Laskey, and K. Chang, "Modeling a probabilistic ontology for maritime domain awareness," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, 2011, pp. 1–8. (1 citation on page 34.)

[69] M. Hanninen, O. A. V. Banda, and P. Kujala, "Bayesian network model of maritime safety management," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7837 – 7846, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417414003704 (1 citation on page 34.)

[70] M. Hanninen and P. Kujala, "Bayesian network modeling of port state control inspection findings and ship accident involvement," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1632 – 1646, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417413006817 (1 citation on page 34.)

[71] M. Kruger, L. Ziegler, and K. Heller, "A generic bayesian network for identification and assessment of objects in maritime surveillance," in *Information Fusion (FUSION), 2012 15th International Conference on*, 2012, pp. 2309–2316. (1 citation on page 34.)

[72] F. Fooladvandi, C. Brax, P. Gustavsson, and M. Fredin, "Signature-based activity detection based on bayesian networks acquired from expert knowledge," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, 2009, pp. 436–443. (1 citation on page 34.)

[73] Y. Zhang and Q. Ji, "Active and dynamic information fusion for multisensor systems with dynamic bayesian networks," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 2, pp. 467–472, 2006. (1 citation on page 34.)

[74] G. Yang, Y. Lin, and P. Bhattacharya, "A driver fatigue recognition model based on information fusion and dynamic bayesian network," *Information Sciences*, vol. 180, no. 10, pp. 1942 – 1954, 2010, special Issue on Intelligent Distributed Information Systems. (1 citation on page 34.)

[75] V. I. Pavlovic, "Dynamic bayesian networks for information fusion with applications to human-computer interfaces," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1999. (1 citation on page 34.)

[76] E. Blasch, E. Bosse, and D. Lambert, *High-level Information Fusion Management and Systems Design*, ser. Artech House intelligence and information operations series. Artech House, 2012. (4 citations on pages 35, 62, 78, and 80.)

[77] D. A. Lambert, "A blueprint for higher-level fusion systems," *Information Fusion*, vol. 10, no. 1, pp. 6 – 24, 2009. (1 citation on page 35.)

[78] D. Lambert, "Stdf model based maritime situation assessments," in *Information Fusion, 2007 10th International Conference on*, 2007, pp. 1–8. (1 citation on page 35.)

[79] T. Strang and C. Linnhoff-Popien, "A context modeling survey," in *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004, pp. –. (1 citation on page 35.)

[80] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161 – 180, 2010, context Modelling, Reasoning and Management. (1 citation on page 35.)

[81] L. S. Kennedy and S.-F. Chang, "A reranking approach for context-based concept fusion in video indexing and retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, ser. CIVR '07. New York, NY, USA: ACM, 2007, pp. 333–340. (1 citation on page 35.)

[82] A. Steinberg and G. Rogova, "Situation and context in data fusion and natural language understanding," in *Information Fusion, 2008 11th International Conference on*, 2008, pp. 1–8. (1 citation on page 35.)

[83] J. Gómez-Romero, M. A. Serrano, M. A. Patricio, J. García, and J. M. Molina, "Context-based scene recognition from visual data in smart homes: an information fusion approach," *Personal and Ubiquitous Computing*, vol. 16, no. 7, pp. 835–857, 2012. (1 citation on page 35.)

[84] J. Garcia, J. Gomez-Romero, M. Patricio, J. Molina, and G. Rogova, "On the representation and exploitation of context knowledge in a harbor surveillance scenario," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, 2011, pp. 1–8. (1

citation on page 35.)

[85] R. Hegde, J. Kurniawan, and B. Rao, "On the design and prototype implementation of a multimodal situation aware system," *Multimedia, IEEE Transactions on*, vol. 11, no. 4, pp. 645–657, 2009. (1 citation on page 35.)

[86] J. George, J. Crassidis, and T. Singh, "Threat assessment using context-based tracking in a maritime environment," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, 2009, pp. 187–194. (1 citation on page 35.)

[87] O. Sekkas, S. Hadjiefthymiades, and E. Zervas, "Enhancing location estimation through data fusion," in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, 2006, pp. 1–5. (1 citation on page 35.)

[88] O. Sekkas, C. B. Anagnostopoulos, and S. Hadjiefthymiades, "Context fusion through imprecise reasoning," in *Pervasive Services, IEEE International Conference on*, 2007, pp. 88–91. (1 citation on page 35.)

[89] C. Anagnostopoulos, O. Sekkas, and S. Hadjiefthymiades, "Context fusion: Dealing with sensor reliability," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, 2007, pp. 1–6. (1 citation on page 35.)

[90] B. Tetreault, "Use of the automatic identification system (ais) for maritime domain awareness (mda)," in *OCEANS, 2005. Proceedings of MTS/IEEE*, Sept 2005, pp. 1590–1594 Vol. 2. (2 citations on pages 35 and 42.)

[91] N. Bon, G. Hajduch, A. Khenchaf, R. Garello, and J.-M. Quellec, "Recent developments in detection, imaging and classification for airborne maritime surveillance," *Signal Processing, IET*, vol. 2, no. 3, pp. 192–203, September 2008. (1 citation on page 35.)

[92] H. Greidanus, "Satellite imaging for maritime surveillance of the european seas," in *Remote Sensing of the European Seas*, V. Barale and M. Gade, Eds. Springer Netherlands, 2008, pp. 343–358. [Online]. Available: http://dx.doi.org/10.1007/978-1-4020-6772-3_26 (1 citation on page 35.)

[93] ——, "Assessing the operationality of ship detection from space," in *EURISY Symposium New*

*space services for maritime users: The impact of satellite technology on maritime legislation*, 2005. (3 citations on pages 35 and 36.)

[94] T. N. Arnesen and R. B. Olsen, "Literature review on vessel detection," Forsvarets forskningsinstitutt, Tech. Rep., 2004, fFI/RAPPORT-2004/02619. (1 citation on page 35.)

[95] D. Gibbins, D. Gray, and D. Dempsey, "Classifying ships using low resolution maritime radar," in *Signal Processing and Its Applications, 1999. ISSPA '99. Proceedings of the Fifth International Symposium on*, vol. 1, 1999, pp. 325–328 vol.1. (1 citation on page 35.)

[96] M. Teutsch and W. Kruger, "Classification of small boats in infrared images for maritime surveillance," in *Waterside Security Conference (WSS), 2010 International*, 2010, pp. 1–7. (2 citations on pages 35 and 37.)

[97] J. Alves, J. Herman, and N. C. Rowe, "Robust recognition of ship types from an infrared silhouette," in *Command and Control Research and Technology Symposium, San Diego, CA, USA*. Monterey, California. Naval Postgraduate School, June 2004. (1 citation on page 35.)

[98] H. Li and X. Wang, "Automatic recognition of ship types from infrared images using support vector machines," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 6, Dec 2008, pp. 483–486. (1 citation on page 35.)

[99] *Visual surveillance in maritime port facilities*, vol. 6978, 2008. [Online]. Available: http://dx.doi.org/10.1117/12.777645 (1 citation on page 35.)

[100] *Autonomous ship classification using synthetic and real color images*, vol. 8661, 2013. [Online]. Available: http://dx.doi.org/10.1117/12.2005749 (1 citation on page 35.)

[101] C. Corbane, F. Marre, and M. Petit, "Using spot-5 hrg data in panchromatic mode for operational detection of small ships in tropical area," *Sensors*, vol. 8, no. 5, pp. 2959–2973, 2008. (2 citations on page 35.)

[102] O. Vaněk, M. Jakob, O. Hrstka, B. Bošanský, and M. Pěchouček, "Agentc: Fighting maritime piracy using data analysis, simulation and optimization," June 2013, URL: http://agentc-project.appspot.com/. (1 citation on page 36.)

[103] "Blue hub - integrating maritime surveillance data," June 2013, URL: https://bluehub.jrc.ec.europa.eu/. (1 citation on page 36.)

[104] O. Vaněk, B. Bošanský, M. Jakob, and M. Pěchouček, "Transiting areas patrolled by a mobile adversary," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, 2010, pp. 9–16. (1 citation on page 36.)

[105] O. Vaněk, M. Jakob, V. Lisý, B. Bošanský, and M. Pěchouček, "Iterative game-theoretic route selection for hostile area transit and patrolling," in *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, ser. AAMAS '11. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 1273–1274. (1 citation on page 36.)

[106] C. D. Marsh, "Counter piracy: A repeated game with asymmetric information," Master's thesis, Naval Postgraduate School, September 2009. (1 citation on page 36.)

[107] J. C. Sevillano, D. Rios Insua, and J. Rios, "Adversarial risk analysis: The somali pirates case," *Decision Analysis*, vol. 9, no. 2, pp. 86–95, 2012. (1 citation on page 36.)

[108] H. Liwang, J. W. Ringsberg, and M. Norsell, "Quantitative risk analysis: Ship security analysis for effective risk control options," *Safety Science*, vol. 58, no. 0, pp. 98 – 112, 2013. (1 citation on page 36.)

[109] L. Esher, S. Hall, E. Regnier, P. Sanchez, J. Hansen, and D. Singham, "Simulating pirate behavior to exploit environmental information," in *Simulation Conference (WSC), Proceedings of the 2010 Winter*, 2010, pp. 1330–1335. (4 citations on pages 36, 40, 41, and 55.)

[110] G. Baldini, D. Shaw, and F. Dimc, "A communication monitoring system to support maritime security," in *ELMAR, 2010 PROCEEDINGS*, 2010, pp. 243–246. (5 citations on pages 37, 40, 42, and 43.)

[111] Z. L. Szpak and J. R. Tapamo, "Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6669 – 6680, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417410013060 (1 citation on page 38.)

[112] J. G. Sanderson, M. K. Teal, and T. Ellis, "Characterisation of a complex maritime scene using fourier space analysis to identify small craft," in *Image Processing and Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465)*, vol. 2, 1999, pp. 803–807 vol.2. (1 citation on page 38.)

[113] D. Nincic, "Maritime piracy in africa: The humanitarian dimension," *African Security Review*, vol. 18, no. 3, pp. 1–16, 2009. (2 citations on pages 39 and 41.)

[114] D. Guilfoyle, "II. piracy off somalia: UN security council resolution 1816 and IMO regional counter-piracy efforts," *International and Comparative Law Quarterly*, vol. 57, pp. 690–699, 7 2008. (3 citations on pages 39 and 40.)

[115] T. Tsilis, "Counter-piracy escort operations in the gulf of aden," Master's thesis, Naval Postgraduate School, 2011. (1 citation on page 40.)

[116] S. Percy and A. Shortland, "The business of piracy in somalia," *Journal of Strategic Studies*, vol. 0, no. 0, pp. 1–38, 0. (3 citations on pages 40, 41, and 53.)

[117] M. Q. Mejia, P. Cariou, and F.-C. Wolff, "Is maritime piracy random?" *Applied Economics Letters*, vol. 16, no. 9, pp. 891–895, 2009. (1 citation on page 41.)

[118] J. Roy, "Automated reasoning for maritime anomaly detection," in *Proceedings of the NATO Workshop on Data Fusion and Anomaly Detection for Maritime Situational Awareness (MSA 2009), NATO Undersea Research Centre (NURC), La Spezia, Italy*, 2009, pp. 15–17. (2 citations on pages 42 and 43.)

[119] C. Carthel, S. Coraluppi, and P. Grignan, "Multisensor tracking and fusion for maritime surveillance," in *Information Fusion, 2007 10th International Conference on*, 2007, pp. 1–6. (1 citation on page 42.)

[120] M. Balci and R. Pegg, "Towards global maritime domain awareness - recent developments and challenges," in *Information Fusion, 2006 9th International Conference on*, 2006, pp. 1–5. (1 citation on page 42.)

[121] B. Rhodes, N. Bomberger, M. Seibert, and A. Waxman, "Maritime situation monitoring and awareness using learning mechanisms," in *Military Communications Conference, 2005. MIL-*

*COM 2005. IEEE*, 2005, pp. 646–652 Vol. 1. (1 citation on page 42.)

[122] K. Bryan and C. Carthel, "An end-to-end maritime surveillance simulator prototype for the assessment of system effectiveness," in *Waterside Security Conference (WSS), 2010 International*, 2010, pp. 1–5. (1 citation on page 43.)

[123] D. Garagic, B. Rhodes, N. Bomberger, and M. Zandipour, "Adaptive mixture-based neural network approach for higher-level fusion and automated behavior monitoring," in *Communications, 2009. ICC '09. IEEE International Conference on*, 2009, pp. 1–6. (1 citation on page 43.)

[124] R. Lane, D. Nevell, S. Hayward, and T. Beaney, "Maritime anomaly detection and threat assessment," in *Information Fusion (FUSION), 2010 13th Conference on*, 2010, pp. 1–8. (3 citations on pages 43 and 54.)

[125] K. Murphy, "Switching kalman filters," Dept. of Computer Science, University of California, Berkeley, Tech. Rep., 1998. (1 citation on page 45.)

[126] L. R. G. Limited, "Lloyd's register of ships online," September 2013, URL: http://www.lr.org/about_us/shipping_information/Lloyds_Register_of_Ships_online.aspx. (2 citations on pages 47 and 58.)

[127] "The american association of port authorities (aapa) website," June 2013, URL: http://www.aapa-ports.org/. (1 citation on page 48.)

[128] Newsonline, "Somalia & pirates - a mash-up of information from the google earth community on piracy around somalia," November 2008, URL: https://maps.google.com/maps/ms?ie=UTF8&t=h&oe=UTF8&msa=0&msid=101065261274254213289.00045bf8668933f828869&dg=feature. (2 citations on pages 48 and 58.)

[129] G. McLachlan and D. Peel, *Finite Mixture Models*. Canada: John Wiley and Sons, inc, 2000. (13 citations on pages 48, 49, 50, and 61.)

[130] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, 1968. (2 citations on page 51.)

[131] P. E. Hart, N. J. Nilsson, and B. Raphael, "Correction to 'a formal basis for the heuristic determination of minimum cost paths'," *SIGART Bull.*, no. 37, pp. 28–29, Dec. 1972. (2 citations on page 51.)

[132] I. Millington and J. Funge, *Artificial Intelligence for Games*. Morgan Kaufmann/Elsevier, 2009. (2 citations on pages 51 and 52.)

[133] P. Kaluza, A. Kölzsch, M. T. Gastner, and B. Blasius, "The complex network of global cargo ship movements," *Journal of The Royal Society Interface*, vol. 7, no. 48, pp. 1093–1103, 2010, see Figure 1a, page 1094. (2 citations on page 53.)

[134] M. Jakob, O. Vaněk, O. Hrstka, and M. Pěchouček, "Agents vs. pirates: multi-agent simulation and optimization to fight maritime piracy," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS '12. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 37–44. (2 citations on pages 53 and 55.)

[135] J. Bardach, "Vision and the feeding of fishes," in *Fish Behavior and Its Use in the Capture and Culture of Fishes: Proceedings of the Conference on the Physiological and Behavioral Manipulation of Food Fish as Production and Management Tools*, ser. ICLARM conference proceedings 5. International Center for Living Aquatic Resources Management, November 1980, pp. 32–56. (1 citation on page 55.)

[136] O. Breivik and A. A. Allen, "An operational search and rescue model for the norwegian sea and the north sea," *Journal of Marine Systems*, vol. 69, no. 1-2, pp. 99 – 113, 2008. (2 citations on page 56.)

[137] T. T. Zin, P. Tin, T. Toriu, and H. Hama, "A markov random walk model for loitering people detection," in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, 2010, pp. 680–683. (1 citation on page 56.)

[138] E. Blasch, P. Valin, and E. Bosse, "Measures of effectiveness for high-level fusion," in *Information Fusion (FUSION), 2010 13th Conference on*, 2010, pp. 1–8. (5 citations on pages 62, 80, and 123.)

[139] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean

shift," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, 2000, pp. 142–149 vol.2. (3 citations on page 62.)

[140] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, no. 1, pp. 99–109, 1943. (2 citations on page 62.)

[141] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, ser. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, 2009. (6 citations on pages 63 and 95.)

[142] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000. (1 citation on page 63.)

[143] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed., ser. Chapman & Hall/CRC Texts in Statistical Science. Chapman and Hall/CRC, 2003. (1 citation on page 64.)

[144] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Elsevier, 2009. (5 citations on pages 78 and 117.)

[145] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, ser. Adaptive computation and machine learning series. MIT Press, 2012. (4 citations on pages 78 and 118.)

[146] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White, "Revisiting the jdl data fusion model ii," in *In P. Svensson and J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004*, July 2004, pp. 1218–1230. (1 citation on page 80.)

[147] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, Nov 2004, pp. 576–584. (1 citation on page 84.)

[148] V. Monbet, P. Ailliot, and M. Prevosto, "Survey of stochastic models for wind and sea state time series," *Probabilistic Engineering Mechanics*, vol. 22, no. 2, pp. 113 – 126, 2007. (1 citation on page 85.)

[149] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, 2004. (2 citations on page 87.)

[150] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, ser. Artech House Radar Library. Artech House, 1999. (3 citations on pages 87 and 112.)

[151] M. Jordan, E. Sudderth, M. Wainwright, and A. Willsky, "Major advances and emerging developments of graphical models [from the guest editors]," *Signal Processing Magazine, IEEE*, vol. 27, no. 6, pp. 17–138, Nov 2010. (1 citation on page 130.)

[152] E. B. Fox, "Bayesian nonparametric learning of complex dynamical phenomena," Ph.D. dissertation, Massachusetts Institute of Technology, September 2009. (2 citations on pages 130 and 131.)

[153] Y. W. Teh and M. I. Jordan, "Hierarchical bayesian nonparametric models with applications," in *Bayesian Nonparametrics*. Cambridge University Press, 2010, pp. 158–207. (2 citations on page 131.)

[154] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "Bayesian nonparametric methods for learning markov switching processes," *Signal Processing Magazine, IEEE*, vol. 27, no. 6, pp. 43–54, Nov 2010. (2 citations on page 131.)

[155] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "An hdp-hmm for systems with state persistence," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 312–319. [Online]. Available: http://0-doi.acm.org.innopac.up.ac.za/10.1145/1390156.1390196 (1 citation on page 131.)

[156] E. Fox, E. Sudderth, and A. Willsky, "Hierarchical dirichlet processes for tracking maneuvering targets," in *Information Fusion, 2007 10th International Conference on*, July 2007, pp. 1–8. (1 citation on page 131.)

[157] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "Bayesian nonparametric inference of switching dynamic linear models," *Signal Processing, IEEE Transactions on*, vol. 59, no. 4, pp. 1569–1585, April 2011. (1 citation on page 131.)