Maritime Piracy Situation Modelling with Dynamic Bayesian Networks

Joel Janek Dabrowski[a,*], Johan Pieter de Villiers[b,a]

[a] University of Pretoria, 2 Lynnwood Rd, Pretoria, South Africa
[b] Council for Scientific and Industrial Research, Meiring Naudé Rd, Lynnwood, Pretoria, South Africa

Abstract

A generative model for modelling maritime vessel behaviour is proposed. The model is a novel variant of the dynamic Bayesian network (DBN). The proposed DBN is in the form of a switching linear dynamic system (SLDS) that has been extended into a larger DBN. The application of synthetic data fabrication of maritime vessel behaviour is considered. Behaviour of various vessels in a maritime piracy situation is simulated. A means to integrate information from context based external factors that influence behaviour is provided. Simulated observations of the vessels kinematic states are generated. The generated data may be used for the purpose of developing and evaluating counter-piracy methods and algorithms. A novel methodology for evaluating and optimising behavioural models such as the proposed model is presented. The log-likelihood, cross entropy, Bayes factor and the Bhattacharyya distance measures are applied for evaluation. The results demonstrate that the generative model is able to model both spatial and temporal datasets.

Keywords: Behavior Modeling, Dynamic Bayesian Network, Switching Linear Dynamic System, Contextual Information, Maritime Domain Awareness, Multi-Agent Simulation.

1. Introduction

Real-world data of illegal activities such as maritime piracy and illegal immigration is scarce [1]. Furthermore, the maritime piracy data that does exist is considered incomplete. Some ship owners do not report pirate attacks in order to avoid insurance costs and lengthy investigations [2]. Real world data is often required for developing applications that counter such illegal activities. Applications may include automatic situation assessment and threat assessment methods. In the maritime piracy domain, an ideal threat assessment method should identify a vessel as a threat before a pirate attack occurs. This may be performed by utilising a model that describes the behaviour of pirate vessels in their prowling state. To form the model of pirate prowling behaviour, data of pirate vessels in this state is generally required. No such data has been found. This study proposes a generative model that is able to model behaviour and generate synthetic behavioural data.

A multi-agent generative model of a maritime piracy situation is proposed. The model consists of a novel variant of a dynamic Bayesian network (DBN) that extends the switching linear dynamical system (SLDS). The DBN is hybrid DBN that consists of both discrete and continuous variables. The structure of the DBN is informed by prior knowledge of the problem. Behaviour of various vessels in a maritime piracy situation is modelled by the DBN. The behaviour consists of various activities such as sailing, target acquisition, and attacking. The proposed DBN provides the capability to model a vessel at the level of the motion state vector. The model is provided with information such as the vessel class and various contextual elements. Synthetic data such as track data of a particular vessel being simulated is generated.

The proposed model is applied to generate a synthetic dataset of pirate attack locations. The model is evaluated by comparing the synthetic dataset to a real world dataset. The real world dataset consists of a set of locations of the pirate attacks that occurred in 2011. A novel method of evaluating and optimising the model is proposed. The evaluation is expressed as a likelihood that indicates the capability of the generative model to produce the real world dataset. An optimisation procedure is demonstrated for optimising the model parameters. The results indicate that the generative model has the ability to produce real-world-like data in terms of a statistical distribution. Cross validation is applied to evaluate how well the model generalises the 2011 pirate attack dataset. An evaluation using Bayes factor indicates that the proposed model performs well. The temporal modelling capability of the model is demonstrated. Pirate behaviour is modelled such that a particular temporal distribution of monthly pirate attacks is generated. This distribution is compared to temporal 2011 pirate attack data.

Over each simulation, unique results may be produced by the generative model. The desired statistical structure is maintained over each simulation. The ability to define unique results is ideal for generating data for algorithms such as machine learning algorithms.

The novelty of this work lies in the use of a SLDS in the DBN to generate simulated track data. No applications have been found in literature where the SLDS is extended into a DBN for providing a context based behavioural model. The proposed

[*] Corresponding author.
Email address: joeldabrowski@gmail.com (Joel Janek Dabrowski)

DBN provides a complete framework for synthetic data generation. A novel framework for evaluating behavioural models is presented. The proposed evaluation and optimisation method may easily be adapted to other problems. The purpose of this work is primarily for the testing of maritime pirate behaviour detection algorithms.

2. Background and Related Work

The DBN proposed in this study may be considered as a multi-agent system. Each vessel modelled by the DBN may be considered as an agent. Multi-agent systems have been applied in various fields. These include robotics, computer games, simulation, econometrics, military and social sciences [3, 4]. Multi-agent Based Simulation (MABS) is a relatively new paradigm for modelling and simulating entities in an environment [5]. Agents are generally considered to be autonomous, independent and able to interact with their environment and other agents [6, 5]. The military MABS is intended to enhance training and support decision making [7]. The application considered in this study may be argued to be a form of a military based MABS. A review of military based MABS applications are provided in [8].

The Bayesian network (BN) [9] is a directed graphical model. The BN exists in various forms that include the dynamic Bayesian network and the influence diagram. The DBN [10] is a temporal extension of the Bayesian network (BN) [9]. Applications of the DBN include computer vision based human motion analysis [11], situation awareness [12] and vehicle detection and tracking [13]. The influence diagram (ID) is a BN supplemented with decision variables and utility functions [14]. It could be argued that the higher levels of the DBN model presented in this study form an influence diagram. IDs been applied to solve a vast number of decision problems. Poropudas and Virtanen have used IDs in the analysis of simulation data [15]. Their work has been extended to include the use of DBNs for the application of simulation [16, 17]. Time evolution is studied and what-if-analysis is performed. The simulation approach is applied to problems involving server queuing and simulated air combat. The structure of the DBNs are application specific and are not necessarily relevant to modelling maritime vessels in a pirate situation. The use of expert knowledge to construct the DBN is suggested as a possible extension to their work. In this study, prior knowledge is used to inform the structure of the DBN for modelling maritime vessel behaviour.

The SLDS [18, 19, 20] is a form of a DBN. In literature, various names are associated with the SLDS. These include the switching Kalman filter and the switching state space model [19]. The SLDS has been successfully applied to various problems that include human motion modelling in computer vision [21], econometrics [22] and speech recognition [23]. No attempts to use SLDS as a generative model for data synthesis have been found in literature.

This study is intended to fall within the framework of information fusion. The structure of the DBN is formulated to provide the means fuse information from various sources. A wide variety of maritime surveillance applications within the field of information fusion exist. A simulation test-bed has been developed for coastal surveillance [24]. The test-bed is developed for the study of distributed fusion, dynamic resource and network configuration management, and self synchronising units and agents. The BN has been used for information fusion for maritime security [25, 26] and maritime domain awareness [27, 28]. A DBN has been proposed for multi-sensor information fusion [29]. Data from various sensors such as imaging sensors, acoustic sensors and radar sensors may be fused using the DBN. A DBN has been applied for information fusion in a driver fatigue recognition system [30]. The DBN is a discrete based DBN that provides an indication of the level of fatigue of a driver. A hybrid DBN has been used for gesture recognition in human-computer interfaces [31]. The DBN may be argued to be in the form of an SLDS. In these applications, the DBN is used for recognition and detection. In this study, the DBN is used for simulation and data generation.

Context-based applications incorporate and model contextual information. The model proposed in this study provides a means to incorporate various external elements that influence behaviour. A survey of context modelling has been conducted by Strang and Linnhoff-Popien [32]. A more recent survey on context modelling and reasoning in pervasive computing has been conducted in [33]. Context-based information fusion has been applied to video indexing [34], computer vision [35] and natural language processing [36]. Context-based information fusion has found use in various maritime situation and threat assessment applications [37, 38, 39]. The DBN is not used in any of these applications. A DBN has been used in context based information fusion system for location estimation [40]. This system has been extended to include fuzzy logic for imprecise contextual reasoning [41, 42]. As for information fusion applications, the DBN and BN are generally used for detection and recognition. The use of the DBN for synthetic data has not been found in literature.

Website applications for situation awareness have been made available. The model proposed in this study could be considered for deployment on a website based system. An on line data visualisation and risk assessment tool for maritime piracy is available [43]. The European Commission has developed the Blue Hub for maritime surveillance data gathering [44]. The platform is currently in development for maritime piracy awareness.

Maritime piracy is a problem of international concern. Maritime piracy poses humanitarian, economic and environmental risks [45]. In late 2008 three counter-piracy missions were deployed. These include the EU's Operation 'Atlanta', NATO's Operation 'Ocean Shield' and the US-led Combined Task Force-151 [46]. These operations have deployed war ships to patrol high risk regions and assist maritime piracy victims. Due to the vast patrol regions, patrolling efforts are partially successful. The use of technology is proposed to assist in combating maritime piracy [47]. In September 2011, an advanced study institute (ASI) was held in Salamanca, Spain to discuss the maritime piracy problem. The objective of the discussions was to help deter predict and recognise maritime piracy using information systems [48]. Topics such as information fusion methods, situation assessment methods, surveillance and challenges associated with the collaboration between information systems and humans were discussed. This study is intended to provide an information system that may be used in the counter-piracy endeavour.

This study considers the maritime piracy problem. Various applications have been proposed in literature for combating maritime piracy. Game theory has been used to optimise counter piracy strategies. Game theory has been utilised to suggest transport routes that avoid maritime pirates [49, 50]. A game theoretic approach that seeks to optimise counter piracy patrolling strategies has been implemented [51]. Risk analysis has been used to assist ship owners and captains in managing risk during a pirate attack [52, 53]. Methods for pirate detection have been discussed in literature. An approach to detect pirates through satellite communication monitoring has been proposed [54]. Other approaches intend to detect pirate vessels by classifying small craft in imagery [55], [56].

A state based multi-agent simulation environment has been proposed for simulating maritime entity behaviour [1]. Long-haul shipping, Piracy Behaviour and Patrolling behaviour are simulated in the system. Vessel behaviour simulations are implemented using finite state machines. Long-haul shipping behaviour is based on a model where cargo ships follow a route that minimizes travel time, costs and security. Pirate behaviour includes activities such as discovering, approaching and attacking vessels. Patrol vessels are placed at near optimum locations according to their deterrence potential as well as according to a risk map. An algorithm is used to determine a set of routes for a set of patrol vessels that maximizes deterrence. The behavioural models described in [1] are used to inform the structure of the DBN described in this study.

A method of simulating pirate kinematic behaviour has been proposed [57]. The simulation is based on the model where pirates venture out in skiffs from a home base in search of targets. The skiffs motor out to a predestined location and drift until supplies have been depleted. Once the supplies have been depleted, the pirates return to the base to refresh their supplies. To simulate the drifting of the pirate vessels, meteorological and oceanographic forecasts are utilized. The drifting behaviour described in [57] is integrated into the DBN behavioural model in this study.

3. Dynamic Bayesian Networks and the Linear Dynamic Switching System Model

The Bayesian network provides a means of statistically modelling causal relationships in data. The dynamic Bayesian network (DBN) extends the Bayesian network to allow modelling of sequential data [19, 20, 58, 59, 10].

The switching linear dynamic system (SLDS) is a mathematical model that may be considered a subclass of DBNs [20]. The switching linear dynamic system provides a means to model a system whose linear parameters change over time. A proposed variation of the classical SLDS model is described by the following state space equations:

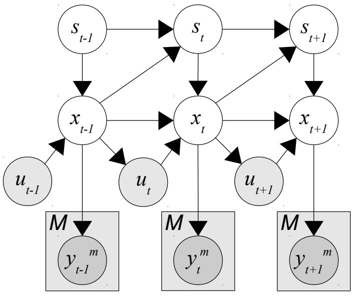


Figure 1: Dynamic Bayesian network (DBN) representation of a switching linear dynamic system (SLDS) for three time slices. The s_t node denotes the switching process state, the x_t node denotes the system state vector, the u_t node denotes the control input and the y^m_t node denotes the m^{th} sensors observed measurement vector at time t.

$$x_t = A(s_t)x_{t-1} + B(s_t)u_t + v_t(s_t), \tag{1}$$

$$y^m_t = C(s_t)^m x_t + w_t(s_t)^m \tag{2}$$

and

$$u_t = f(x_{t-1}). \tag{3}$$

In (1), x_t is the state vector, u_t is the control vector, $A(s_t)$ is the system matrix, $B(s_t)$ is the input matrix and $v_t(s_t)$ is the state noise process. In (2), y^m_t is the observed measurement, $C(s_t)^m$ is the observation matrix and $w_t(s_t)^m$ is the measurement noise. The variables in (2) describe the measurements from the m^{th} sensor selected from a set of M sensors. Equations (1) and (2) describe the typical linear dynamic system equations in state form [60]. In (3), u_t is the control vector and $f(x_{t-1})$ is the control function. The control function $f(x_{t-1})$, transforms the vessel's previous state x_{t-1} to a control vector u_t.

The additional parameter, s_t in the state equations is the switching processes state. It is assumed that s_t follows a first order Markov process [61]. As the switching process state changes, the linear dynamic systems parameters change. This provides the means to model a complex dynamic system through varying states or activities.

The SLDS described by (1), (2) and (3) may be represented as the DBN model illustrated in Figure 1. The DBN model is described by the following joint probability distribution:

$$p(\bar{y}_{0:T}, x_{0:T}, u_{0:T}, s_{0:T}) = \prod_{t=0}^{T}\prod_{m=1}^{M} p(y^m_t|x_t)p(x_t|s_t, x_{t-1}, u_t)p(s_t|s_{t-1}, x_{t-1}). \tag{4}$$

The conditioning between variables corresponds to the SLDS system equations provided in (1), (2) and (3).

4. Maritime Piracy Situation DBN Model

To model behaviour, it is proposed that the switching process state s_t in an SLDS be represented as a DBN. The proposed DBN for modelling vessel behaviour in a maritime piracy situation is illustrated in Figure 2. In this model, the process state,

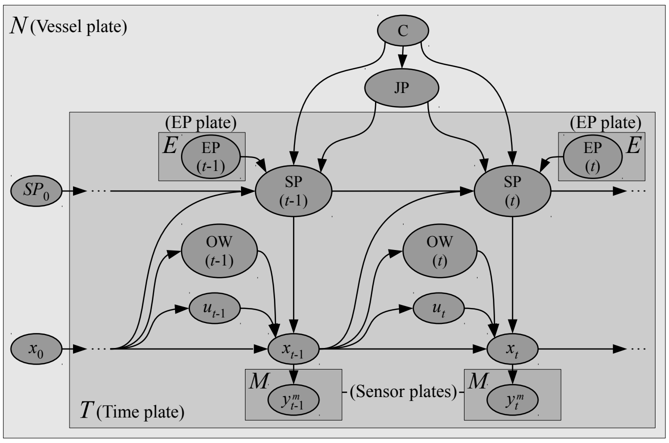


Figure 2: Dynamic Bayesian Network (DBN) model for a vessel in a maritime piracy situation.

s_t is essentially expressed as a DBN that includes the class (C), journey parameters (JP), external parameters (EP) and the state parameter (SP) random variables. The x_t, y_t^m and the u_t variables are the state space equation vectors for the SLDS as described in section 3. The ocean/weather (OW) variable is included to model the influences of ocean and weather conditions on the motion of vessels.

The DBN model in Figure 2 is represented in plate notation. Each of the N vessels in the environment is represented by a vessel plate. The time plate contains the dynamic nodes that transition between states over the T discrete time steps. A set of E external parameters are represented by the EP plates. The set of M sensors are represented by the sensor plates.

4.1. Class Variable (C)

The C variable represents the class of a particular vessel. The C variable describes either a pirate vessel, a transport vessel or a fishing vessel. The pirate vessel label is associated with a pirate mothership and its associated pirate skiffs. The transport vessel label is associated with commercial long-haul vessels. These may include cargo ships, oil tankers, cruise ships and container ships. The fishing vessel label is associated with small private or commercial fishing vessels. In particular, the fishing vessel class considers vessels that remain near the coastline and prefer particular fishing regions.

The probability distribution of transport and fishing vessels may be informed by ship registers and ocean traffic statistics. Lloyd's register of ships [62] is a potential source for such data. The statistics for pirate vessels may be informed by piracy report statistics provided by the International Maritime Bureau (IMB).

4.2. Journey Parameters Variable (JP)

The JP variable is a random variable that selects the home location, the route way-points and the destination location for a particular vessel. These variables are selected from to a list of predefined locations and routes. The predefined locations include world ports, fishing towns, pirate ports, fishing zones and pirate zones. Way-points determine route paths. Route paths consist of straight lines between a set of way-points.

The JP variable is conditionally dependent on the C variable. The dependence places constraints on the possible locations that may be selected given the class. A pirate class may only select a pirate port as a home location and a pirate zone as a destination location. A fishing class may only select a fishing port as a home location and a fishing zone as a destination location. Transport vessels select from a distribution of various world ports.

The probability distribution of world ports may be informed by world port rankings. The American Association of Port Authorities (AAPA) [63] produce world port rankings. Fishing zones may be inferred from fishing vessel traffic data or from legalized fishing zones. Pirate zones may be determined from attack locations provided in IMB published attack reports. The method for determining pirate zones is presented in Section 4.8. Pirate ports may be inferred from locations where pirate ransoms have been conducted.

4.3. External Parameters (EP)

The external parameters variable describes context based external factors that influence the behaviour of vessels. A set of E external parameters may be provided. External factors may include date, time, season, ocean conditions and weather conditions. The EP variables are considered to be observable. The observations of the variables may be obtained from data sources. Data sources may include oceanographic and climatic models or data.

The international comprehensive ocean atmosphere data set (ICOADS) provides surface marine data. The dataset includes information such as air temperature, sea surface temperature, humidity, wind, wave data, cloud cover and air pressure data. Each of these elements may influence behaviour and may be included as external parameters. For example, pirates are known to avoid conditions such as monsoon seasons, high winds, high waves and strong ocean currents [2, 64].

The geographic location of a vessel may have an influence on the vessels behaviour. Transport vessels will generally prefer to travel along known shipping routes. Fishing vessels will prefer fishing zones that are defined by fish habitat and fishing laws and regulations. Pirate vessels will prefer to remain near transport routes where targets may be acquired.

Times and seasons may affect behaviour. Pirates prefer attacking targets during hours of darkness [65]. Fishing vessels may prefer fishing during the hours of dusk and dawn. Dusk and dawn are generally associated with feeding times of many fish species [66]. The small craft used by fishermen and pirates are particularly susceptible to harsh sea conditions. Pirates and fishermen are known to avoid the monsoon seasons due to the harsh sea conditions associated with them [2, 64].

4.4. State Parameters Variable (SP)

The SP variable provides an indication of the nature of a particular vessels kinematic activity or behaviour. The SP variable is defined to contain the *anchor*, *sail-out*, *sail-home*, *fish*, *drift*, *attack* and *abort-attack* states. The SP variable is conditioned on the C variable. This dependence dictates which SP states may be utilised for a particular vessel class. The state parameters and their associativity may be described with state transition diagrams illustrated in Figure 3.

The state transition diagram for the transport vessel is illustrated in Figure 3a. It is assumed that transport vessels travel from a home port to a destination location along the most economical route [67, 1]. The vessel is in an anchor state when located at its home location. The switching state variable transitions to the sail-out state when the vessel is required to sail to its destination. When reaching the destination port, the vessel returns to the anchor state.

The state transition diagram for a fishing vessel is illustrated in Figure 3b. It is assumed that fishermen prefer fishing during particular times and seasons. The fishing vessel will remain in an anchor state at its home location. At dawn or dusk, the vessel will transition to a sail-out state. The vessel sails out to a fishing zone. Once in the fishing zone, the fishing vessel will enter the fish state. After fishing, the fishing vessel will transition into a sail-home state. When the home location is reached, the fishing vessel returns to an anchor state.

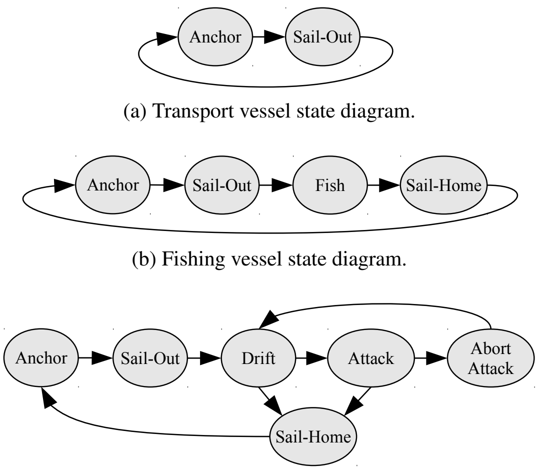
The state transition diagram for a pirate vessel is illustrated in Figure 3c. This model is based on the model proposed in [1]. A pirate vessel will leave its anchor state to sail out to a pirate zone in a mothership. When the pirate zone is reached, the pirate vessel transitions to a drift state where the pirates wait for a target [57]. On detection of a target, the pirate vessel will enter an attack state and attack the target with small high speed boats such as skiffs [65, 45]. If the attack is successful, the pirates will return home with the hijacked vessel to ransom it. The mothership is left abandoned. If the attack is unsuccessful, the pirate vessel enters the abort-attack state. In this state, the skiffs return to the mothership and return to the drift state.

The SP variable is dependent on the EP variable. External factors influence vessel behaviour. Time and ocean conditions are external factors that are considered in this study. Pirate and fishing vessels will avoid harsh sea conditions and will prefer particular times an seasons.

The pirate drift state is a target acquiring state. The pirate class is required have a perception of surrounding vessels. For this, the SP variable must be conditionally dependent on other vessels measurement vector y_t^m. This implies that there is conditional dependence between the N vessel plates.

4.5. Ocean Current/Weather Variable (OW)

The OW variable describes ocean conditions and weather conditions. This variable is included to influence the motion of vessels. The variable provides a means for the ship vessel to drift according to the ocean currents and wind specified at the particular location of the vessel. This variable is considered to be observable. The parameters for this variable may be obtained from oceanographic and climatic models or data such as proposed in [57]. The OW variable is dependent on x_t at the previous time step. This provides a means to determine the localised ocean and weather conditions.



(a) Transport vessel state diagram.

(b) Fishing vessel state diagram.

(c) Pirate vessel state diagram.

Figure 3: State transition diagrams describing the state-parameters node for each class.

4.6. Linear Dynamic System Vectors

The linear dynamic system vectors include the state vector x_t, the control vector u_t and the observed measurement vector y_t^m. These vectors are related by the state space equations (1), (2) and (3). The state vector may be assumed to contain the location, the velocity and the acceleration of a vessel. In this study only the longitudinal and latitudinal components of the location and velocity are considered. The measurement vector may model any or all of the contents of x_t. For example, a radar sensor may provide the position of the vessel.

The control function $f(x_t)$ computes the control vector u_t such that the vessel sails between the way-points specified by the JP variable. The function may control the direction and the speed of a vessel such that it sails through each way-point on its path. At each time step the control function may compute the direction required for the vessel to reach the next way-point. Complex vessel motion may be modelled with the combination of the control function, the way-points and the state noise process.

4.7. Ancestral Sampling in the Proposed Model

To generate data from the generative model, the ancestral sampling method may be used. The ancestral sampling method involves a process of sampling from the root nodes to the leaf nodes. In the generative model, the root nodes include the class and the contextual parameters. The leaf nodes are the measurement vectors. The measurement vector simulates sensors that observe the vessel. At each time step the sampling process is performed to generate simulated samples from the sensors for each vessel. The ancestral sampling algorithm applied to the proposed DBN behavioural model is provided in Algorithm 1.

4.8. Gaussian Mixture Model Fitting

The Gaussian mixture model (GMM) may be fitted to a spatial dataset consisting of pirate attack locations. The GMM provides an estimate of the probability density function that describes the dataset. A GMM may be fitted to a simulated pirate

Algorithm 1 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 $JP$.
    for each time step $t$ and each sampled vessel 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 $x_t$.
        Sample each of the $M$ observation vectors $y_t^m$.
    end for
end for
```

attack dataset as well as a real-world pirate attack dataset. In the case of the simulated dataset, the GMM may be used for evaluating the simulation results. In the case of the real-world dataset, the GMM serves as an estimation of pirate zones. The pirate zones density estimation provides a prior distribution of pirate attacks.

Let $\bar{a} = (\bar{a}_1^T, ..., \bar{a}_n^T)^T$ describe the vector of the simulated pirate attack locations. Each $\bar{a}_j$, $j = 1, ..., n$ contains the longitude and latitude of the j^{th} pirate attack location. Similarly, let $\bar{b} = (\bar{b}_1^T, ..., \bar{b}_n^T)^T$ describe the vector of the real-world pirate attack locations.

The g-component GMM describing the dataset $\bar{a}$ is given as follows [68]:

$$q(\bar{a}_j; \psi_S) = \sum_{i=1}^{g} \pi_i \phi_i(\bar{a}_j; \mu_i, \Sigma_i). \tag{5}$$

The variable ψ_S describes the parameters of the GMM for the simulated dataset. These include the mixture weights π_i and the Gaussian parameters μ_i and Σ_i. The function $\phi_i()$ denotes the i^{th} Gaussian mixture component. The variable π_i describes the weight of the i^{th} Gaussian mixture component. Variables μ_i and Σ_i describe the mean and covariance of the i^{th} Gaussian mixture component respectively.

Similarly, The g-component GMM describing the dataset $\bar{b}$ is given as follows:

$$q(\bar{b}_j; \psi_R) = \sum_{i=1}^{g} \pi_i \phi_i(\bar{b}_j; \mu_i, \Sigma_i). \tag{6}$$

The variable ψ_R describes the parameters of the GMM for the real-world dataset. For notational simplicity, the algorithm used for fitting the GMM to a dataset will be described for the simulation dataset $\bar{a}$.

The likelihood for the model parameters ψ_S is formed from the observed data. The likelihood is given by [68]:

$$\mathcal{L}(\psi_S) = \prod_{j=1}^{n} q(\bar{a}_j; \psi_S). \tag{7}$$

The log-likelihood is often a more convenient representation of the likelihood in application. The log-likelihood is given by [68]:

$$\log \mathcal{L}(\psi_S) = \sum_{j=1}^{n} \log q(\bar{a}_j; \psi_S). \tag{8}$$

The GMM is fitted to the simulated dataset using the expectation maximization (EM) algorithm. The observed data vector $\bar{a}$ is considered to be incomplete in the EM algorithm. The complete data vector includes the associated component-label matrix $\bar{l} = (\bar{l}_1, ..., \bar{l}_n)^T$ such that:

$$\bar{a}_c = (\bar{a}^T, \bar{l}^T)^T. \tag{9}$$

Each $\bar{a}_j$ is assumed to have arisen from one of the GMM components. The vector $\bar{l}_j = [l_{1j}, ..., l_{ij}, ..., l_{gj}] \in \bar{l}$ is a g-dimensional vector containing indicator variables. Label $l_{ij} \in \bar{l}_j$ is assigned the value 1 or 0 according to whether a_j arose from the i^{th} mixture component or not ($i = 1, ..., g; j = 1, ...n$). The complete-data log likelihood for ψ_S is given as [68]:

$$\log \mathcal{L}_c(\psi_S) = \sum_{i=1}^{g} \sum_{j=1}^{n} l_{ij} \left(\log \pi_i + \log q(\bar{a}_{cj}; \psi_S) \right). \tag{10}$$

The E-step of the EM algorithm requires the computation of the conditional expectation of $\log \mathcal{L}_c(\psi_S)$ given $\bar{a}$. In the k^{th} iteration of the algorithm, this value is given by the following expectation [68]:

$$\begin{aligned} Q(\psi_S; \psi_S^{(k)}) &= \mathbb{E}_{\psi_S^{(k)}} \left(\log \mathcal{L}_c(\psi_S) | \bar{a} \right) \\ &= \sum_{i=1}^{g} \sum_{j=1}^{n} \tau_i(\bar{a}_j; \psi_S^{(k)}) \left(\log \pi_i + \log q(\bar{a}_{cj}; \psi_S) \right). \end{aligned} \tag{11}$$

The value $\tau_i(\bar{a}_j; \psi_S^{(k)})$ describes the expectation of the random variable Z_{ij} with respect to the observed data $\bar{a}$. This value is given by [68]:

$$\tau_i(\bar{a}_j; \psi_S^{(k)}) = \tau_{ij}^{(k)} = \frac{\pi_i \phi(\bar{a}_j; \mu_i, \Sigma_i)}{\sum_{h=1}^{g} \pi_h \phi(\bar{a}_j; \mu_h, \Sigma_h)}. \tag{12}$$

The M-step of the EM algorithm requires the global maximization of $Q(\psi_S; \psi_S^{(k)})$ with respect to ψ_S. This computation exists in closed form for Gaussian components. The M-step involves the updating of the component means and covariance matrices at the k^{th} iteration. The update for the mean is given as follows [68]:

$$\mu_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} \bar{a}_j}{\sum_{j=1}^{n} \tau_{ij}^{(k)}}. \tag{13}$$

The update for the covariance matrix is given as follows [68]:

$$\Sigma_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} (\bar{a}_j - \mu_i^{(k+1)})(\bar{a}_j - \mu_i^{(k+1)})^T}{\sum_{j=1}^{n} \tau_{ij}^{(k)}}. \tag{14}$$

The update for the mixture weight is given by [68]:

$$\pi_i^{(k+1)} = \sum_{j=1}^{n} \tau_i(\bar{a}_j; \psi_S^{(k)}) / n. \tag{15}$$

The E- and M-steps are alternated repeatedly until convergence. The EM algorithm converges when $\log \mathcal{L}_c(\psi_S^{(k+1)}) - \log \mathcal{L}_c(\psi_S^{(k)}) < \epsilon$, where ϵ is a small arbitrary value [68]. This same procedure applied for fitting a GMM with parameters ψ_R, to the real-world dataset $\bar{b}$.

5. Spatial Domain Evaluation and Results

The proposed model is evaluated by comparing spatial distributions that describe the real-world pirate data and the simulated data. The spatial region of pirate attacks is limited to the region of the Gulf of Aden and the Indian Ocean. A set of 235 reported attacks that occurred in this region are extracted from the 2011 IMB annual piracy report [64]. The set of 235 attack locations forms a real-world dataset to which the proposed model is compared. A set of simulations are run using the proposed model. Pirate attack locations are recorded during the simulation. The simulation is run until at least 235 pirate attacks have occurred. The set of recorded pirate attack locations form the simulated pirate attack dataset. The simulated dataset is compared with the real-world dataset. The model effectiveness is evaluated according to information gain, quality and robustness.

5.1. *Model Configuration for Simulation*

A set of ports, points-of-interest and way-points are fixed on the map of the Gulf of Aden. The particular assignment of simulated routes of transport vessels and the pirate attack zones is critical. The assignments are delineated according to known shipping lanes and the 2011 pirate attack data. Three pirate ports are initialized; Bosaso, Harardhere and Mogadishu. The pirate port locations are selected based on locations of ransom payments and reported hijacked vessel anchorage locations [69].

The Poisson distribution is appropriate for modelling the number events that occur in time or space [70]. The set of N vessels appearing on the map are modelled by the Poisson distribution. Each vessel is assigned a class by sampling from the C distribution. The classes include pirate, fishing and transport vessels.

The JP variable is sampled from the distribution of ports, pirate zones and fishing zones. Ports, pirate zones and vessel paths for transport and pirate vessels are illustrated in Figure 4. Each vessel is assigned a home port, a destination location and a path between the home port and destination location. The path includes a list of via points. The estimation of the pirate zone parameters is described in Section 5.2.

The EP variable plate contains a single variable that describes the sailing conditions. This variable combines information such as season, time-of-day and ocean conditions. The sailing conditions are described as poor, adequate or favourable conditions. Poor conditions are conditions for pirate and fishing vessels relate to daytime, monsoon seasons, poor ocean conditions or poor weather conditions. Favourable conditions for pirate vessels relate to night time, non-monsoon seasons, favourable ocean and weather conditions. Adequate conditions relate to adequate weather and ocean conditions.

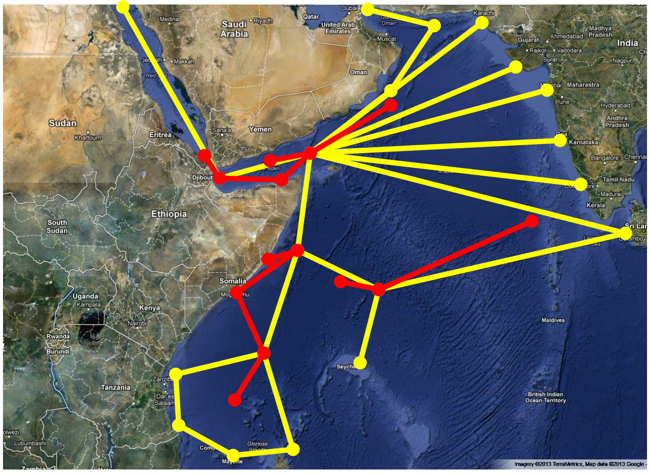


Figure 4: Map of the Gulf of Aden with vessel routes. Yellow routes are routes for transport and pirate vessels. Red routes are routes that extend from pirate ports and pirate zone centres.

The model is configured such that vessels follow a constant velocity model. The x_t, y_t^m and u_t variables are implemented as described by (1), (2) and (3). The y_t^m variable is configured to contain the longitudinal and latitudinal coordinates of the vessel. The OW variable is modelled as a random process.

The control function $f(x_t)$ calculates the control vector u_t such that the vessel sails along a path designated by the JP variable.

5.2. *Pirate Zone Parameters*

The pirate zones are represented by a GMM that has been fitted to the 2011 pirate attack dataset. Each GMM component describes a pirate zone. The mean of a component provides an estimate of the centre of a pirate zone. The covariance matrix provides an estimate of the shape and size of the pirate zone. The probability that a pirate will select a particular pirate zone is estimated by the corresponding GMM component mixture weight. The GMM parameters are determined using the EM-algorithm described in Section 4.8.

The number of mixture components (pirate zones) for a GMM is required to be specified. By computing the likelihood of the data given the number of components, an optimal number of mixture components may be determined. The likelihood of the GMM of the dataset $\bar{b}$, parameterised by ψ_R, given the number of mixture components is given by:

$$\log \mathcal{L}(\psi_r|g) = \sum_{j=1}^{n} \log q(\bar{b}_j; \psi_r|g). \tag{16}$$

This likelihood may be determined over a range of number of mixture components, g.

A plot of the likelihood for the set $g = \{1 \ldots 20\}$ is presented in Figure 5. The results demonstrate that the likelihood increases with increasing g. For the case where $g = 1$, the

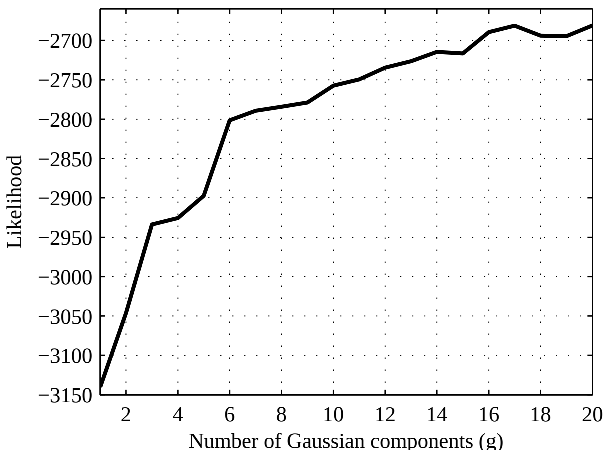


Figure 5: The self-likelihood (16) is plotted over a range of the number of GMM components (g). For each value of g, a GMM is fitted to the 2011 pirate attack dataset and the self-likelihood is computed. The optimal number of components is the value that corresponds to the “knee” of the curve.

likelihood is lowest due to under-fitting. A single GMM component is not able to sufficiently describe the distribution structure. The structure of the distribution requires multiple mixture components for a more accurate representation. Problems with over-fitting may become evident if too many mixture components are used to represent the distribution. The recommended number of components is the value that lies at the “knee” of the curve plotted in Figure 5 [71]. The “knee” of the curve is located at the value of $g = 6$ mixture components.

A plot of the GMM pirate zones is illustrated in Figure 6. In the process of sampling the JP variable, a pirate zone is sampled for each pirate vessel. The probability of pirate zone being selected is given by the GMM weight parameter. The destination of a pirate vessel is located within the selected pirate zone. This destination is a random location. The random location is generated from a Gaussian distribution with parameters given by the pirate zone mixture component parameters.

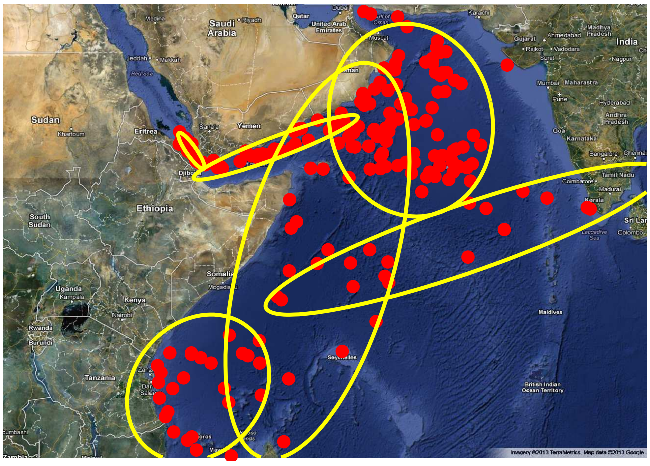


Figure 6: Pirate zones according to a GMM fitted to the 2011 pirate attack dataset. Each pirate zone is represented by a Gaussian mixture component. The yellow ellipses describe the standard deviation of each Gaussian mixture component. The 2011 pirate attack locations are plotted as red markers.

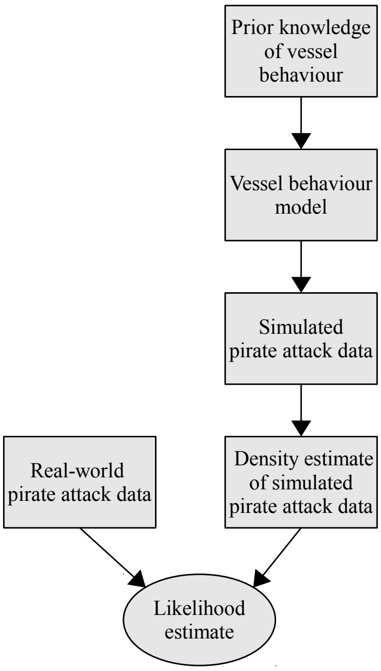

Prior knowledge of vessel behaviour
Vessel behaviour model
Simulated pirate attack data
Real-world pirate attack data
Density estimate of simulated pirate attack data
Likelihood estimate

Figure 7: Block diagram of the optimisation and evaluation process of the proposed model.

5.3. *Evaluation and Optimisation Methodology*

The process of evaluation of the proposed model is described by the block diagram in Figure 7. Prior knowledge is used to form the proposed model of pirate, transport and fishing vessel behaviour. The model is utilised to generate simulated pirate attack data. A GMM estimate of the probability density function of the simulated pirate attack data is computed. The likelihood of the real-world dataset given the simulated dataset is computed. The likelihood provides an indication of the ability of the proposed model to generate the real-world pirate data. This likelihood is applied for the evaluation of the proposed model.

The method described in Figure 7 may be used for the optimisation of the model and its parameters. The likelihood estimate provides a relative measure of the model accuracy. Optimisation involves a parameter space search. The parameter value that results in the most superior model is considered to be the optimal parameter value. The likelihood estimate provides a means to compare the results from using different parameter values. In this study, the parameters are determinable and thus optimisation is not necessary. A demonstration is however provided to describe the optimisation procedure. Furthermore, the data generated in the demonstration is used for evaluating the model.

5.4. *Model Likelihood*

As illustrated in Figure 7, a likelihood estimate is required to be computed for the purpose of evaluating of the proposed model. The likelihood function is defined according to a set of observations originating from a distribution with parameters ψ_S [68]. A likelihood function is to be formed that describes the likelihood of the real-world pirate dataset with respect to the simulated pirate dataset. This likelihood function may be described according to the observations from the real-world dataset and the parameters of the simulated dataset. Note that this likelihood is not to be confused with the likelihood associated with the EM algorithm described in Section 4.8. The parameters of the simulated dataset are the GMM model parameters ψ_S, described in (5). The vector $\bar{b} = (b_1^T, ..., b_n^T)^T$ describes the pirate attack locations of the 2011 dataset. With reference to (8), the log-likelihood function of the real-world dataset with respect to the simulated dataset is given as follows:

$$\log \mathcal{L}(\psi_S) = \sum_{j=1}^{n} \log q(\bar{b}_j; \psi_S). \qquad (17)$$

The function $q(\bar{b}_j; \psi_S)$ is the GMM of the simulated dataset evaluated at the locations given by $\bar{b}_j$, $j = 1...n$.

The results of (17) provide a means for model optimisation and evaluation.

5.5. *Model Optimisation Demonstration*

The optimal model is the model that produces the most likely results. The likelihood is described in section 5.4. For model optimisation, the model parameters may be varied. Parameters associated with the transport routes and pirate zones may be considered. Parameters include the transport vessel paths, the number of pirate zones, pirate zone locations, pirate zone sizes and pirate zone probabilities. For demonstration purposes, the pirate zone size shall be considered for optimisation.

A set of six pirate zones are selected as illustrated in Figure 8. The locations and relative sizes of the pirate zones are selected according to the 2011 attack data and the GMM presented in Figure 6. The probability of a pirate zone is determined by the number of attacks in the pirate zone and the size of pirate zone. The probability of a pirate zone is the probability that the pirate zone will be selected by a pirate. The pirate zones are represented by bivariate Gaussian distributions. The mean value of the Gaussian distribution defines the location of the pirate zone. The covariance of the Gaussian distribution determines the size of the pirate zone.

The covariance of the Gaussian distributions are varied for the purpose of model optimisation. Each Gaussian distribution has a preselected covariance. The covariance of all the Gaussian distributions are scaled by a single scaling factor, σ. The scaling factor is considered over the set of values $\sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. A simulation is performed for each value of σ. The simulated pirate attacks for $\sigma = 1$ is illustrated in Figure 9. This result illustrates condensed clusters of pirate attacks. For comparison, the pirate attack locations of 2011 are illustrated in Figure 10. The results in Figure 9 seem to demonstrate little correlation with the real-world data. The simulated pirate attacks for $\sigma = 6$ is illustrated in Figure 11. The results seem to demonstrate a higher correlation with the 2011 pirate attack data. The simulated pirate attacks for $\sigma = 9$ is illustrated in Figure 12. The results demonstrate a more uniform distribution of pirate attacks.

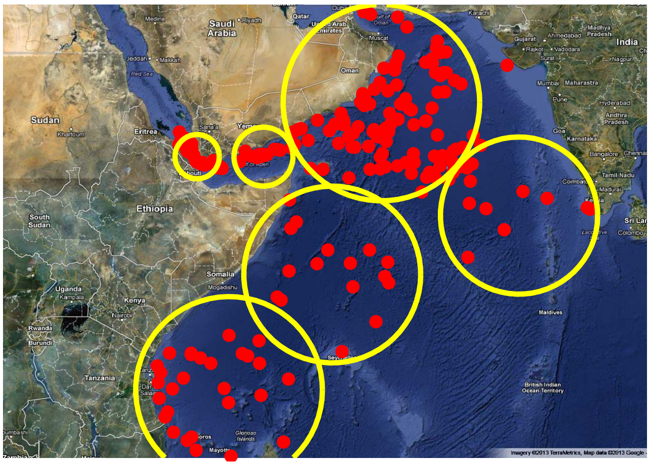


Figure 8: Selected pirate zones for the optimisation demonstration. Each pirate zone is represented by a Gaussian distribution. The yellow rings describe the standard deviation of the Gaussian distributions. The 2011 pirate attack locations are plotted as red markers.

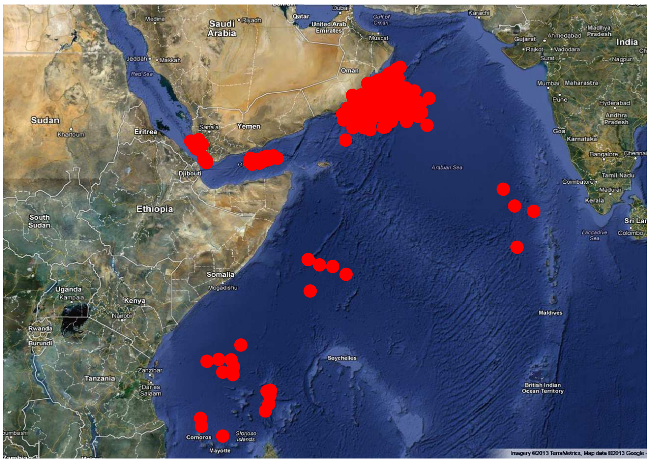


Figure 9: Simulated pirate attacks for $\sigma = 1$

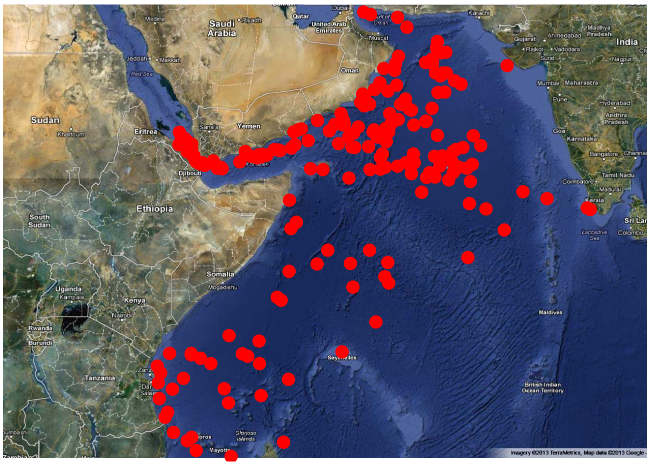


Figure 10: Pirate attack locations of 2011 [64]

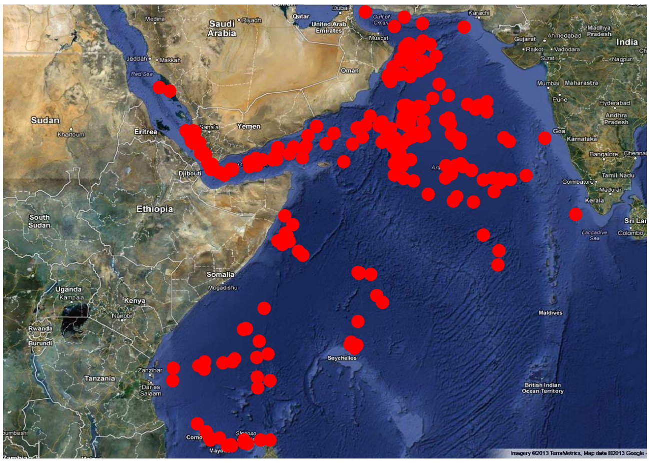


Figure 11: Simulated pirate attacks for $\sigma = 6$. This value produces the optimum results.

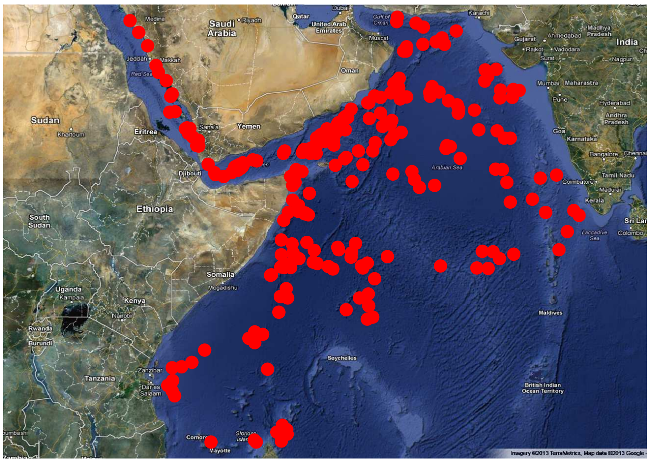


Figure 12: Simulated pirate attacks for $\sigma = 9$

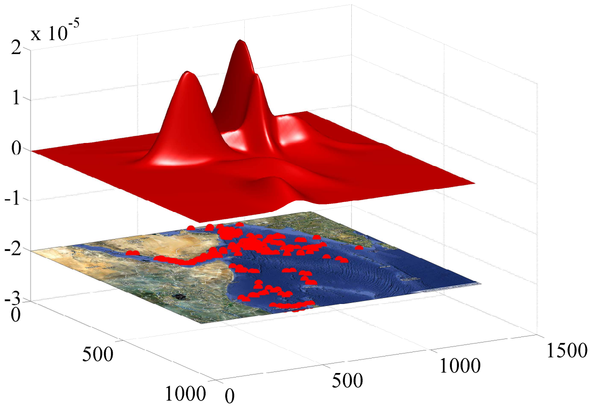


Figure 13: GMM for the simulated pirate attack data for $\sigma = 6$. The surface illustrates the GMM probability density function above the map of the region considered.

Table 1: 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.

σ	$\log \mathcal{L}(\psi_S)$
1	-3983
2	-4172
3	-3781
4	-3232
5	-3173
6	**-2987**
7	-3161
8	-3138
9	-3112

A set of GMMs are fitted to each of the nine simulation results. The EM algorithm described in section 4.8 is applied for fitting the GMMs. The EM-algorithm requires initial parameters for the GMM. The number of Gaussians in the GMM was set as $g = 6$. This corresponds to the number of preselected pirate zones. The initial mean values were set as the preselected pirate zone Gaussian distribution means. The covariance matrices were initialized as diagonal matrices. The diagonal elements were set as the variance of the real-world pirate attack data. The initial weights were set uniformly. The resulting GMM probability density function for $\sigma = 6$ is illustrated in Figure 13.

The model is optimised by considering the likelihood described by (17). The likelihood results for the set of σ values is presented in Table 1. The optimum value is demonstrated to be $\sigma = 6$. The model with $\sigma = 6$ is considered to be the optimum model.

5.6. Optimisation Results Validation

The simulations described in section 5.5 were repeated. Four simulations were run over the set $\sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. A considerable amount of time is required to perform a set of simulations. This limited the number of simulations performed. The error-bar plot of the simulation results is illustrated in Figure 14. The line plot and associated error bars represent the mean values and standard deviations respectively of the log-likelihood results over four simulations for each of the σ values considered. The trend provides a confirmation that the maximum likelihood occurs for $\sigma = 6$.

It may be noted that the likelihood does not seem to decrease as σ increases beyond $\sigma = 6$. A cause of this is the structure of the transport vessel routes. A large covariance of a pirate zone will result in larger area considered by the pirate. Pirate attacks will however not occur in regions where no transport vessels sail. The distribution of pirate attacks is thus constrained to the regions in which transport vessels sail. The results illustrated in Figure 12 demonstrate this. Pirate zones with high covariance are simulated. A structure in the simulated pirate attack location distribution is maintained. With reference to Figure 4, the transport routes define the maintained distribution structure. In-

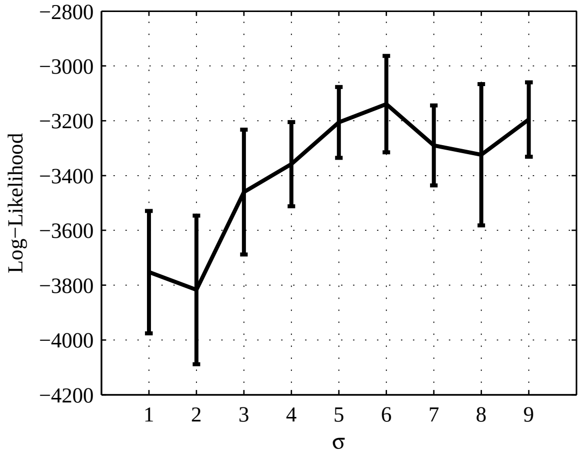


Figure 14: An error-bar plot for a set of four simulations over the set $\sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. The mean value for each σ value is plotted as a line. The error bars describe the standard deviation of the results for each σ value.

creasing the pirate zone variance beyond the constraint will not cause a change in the variance of the simulated attacks. The likelihood results will not vary significantly over large values of σ.

5.7. Cross Validation Based Verification

The ability for a model to generalise a dataset may be evaluated using the method of cross validation [72, 20]. In the cross validation method, a dataset is split into a training set and a validation set. The training set is used for training the model. The validation set is used for evaluating the model. Separating the datasets provides a means to evaluate the performance of the model on unseen data [73]. A number of cross validation folds may be performed. In each cross validation fold a new training set and validation set are formed. The model is trained and evaluated over each fold.

In Algorithm 2, a cross validation based method for evaluating the proposed model is described. The methodology of this algorithm is illustrated in Figure 15. The 2011 pirate attack dataset is split into a training dataset (D_T) and a validation dataset (D_V). The GMM is fitted to the training dataset. This GMM may be considered a model of the real-world dataset. A simulation is performed using the parameters of the training dataset GMM as pirate zone parameters. The simulation produces a simulation dataset (D_S). A GMM is fitted to the simulation dataset. The training set GMM is compared to the simulation dataset GMM using Bayes factor. Bayes factor is given as:

$$K_i = \frac{q(D_V^{(i)};\psi_T)}{q(D_V^{(i)};\psi_S)}. \tag{18}$$

Algorithm 2 Cross validation based verification method for evaluating the proposed model.

```
Require: 2011 attack data consisting of the pirate attack locations.
for fold = 1 to 10 do
    Sample (without replacement) 10% of the 2011 attack dataset to form the validation set $D_V$.
    Set the remaining 90% remaining data as the training set $D_T$.
    Generate a simulation dataset $D_S$ using pirate zone parameters determined from $D_T$.
    Fit a GMM with parameters $\psi_T$ to $D_T$.
    Fit a GMM with parameters $\psi_S$ to $D_S$.
    for each validation sample $i$ do
        Compute the likelihood the validation set sample given the GMM model of the training set $q(D_V^{(i)};\psi_T)$.
        Compute the likelihood the validation set sample given the GMM model of the simulation set $q(D_V^{(i)};\psi_S)$.
        Compute Bayes Factor for each of the two computed sample likelihoods using (18).
    end for
    Compute the median of the Bayes factor using (19).
end for
Plot the Bayes factor median for each fold.
```

The numerator $q(D_V^{(i)};\psi_T)$, is the likelihood of the i^{th} validation dataset sample, given the parameters, ψ_T of the training dataset GMM. The denominator $q(D_V^{(i)};\psi_S)$, is the likelihood of the i^{th} validation dataset sample, given the parameters, ψ_S of the simulation dataset GMM. A likelihood is computed by evaluating the respective GMM at the point given by $D_V^{(i)}$. Bayes factor in (18) 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 given by:

$$K = \text{median}\,(K_i) = \text{median}\left(\frac{q(D_V^{(i)};\psi_T)}{q(D_V^{(i)};\psi_S)}\right), \tag{19}$$

where $i = \{1, \ldots, 10\}$. The median of Bayes factor provides a measure of how many more times likely the validation dataset fits the real world data model than that of the simulation data model. The closer a Bayes factor value is to unity, the more similar the models are. The more similar the models are, the more superior the proposed model is.

The log-likelihood of the validation dataset, given the GMMs is illustrated in Figure 16. The black curve describes the log-likelihood of the validation dataset given the training dataset GMM. The grey curve describes the log-likelihood of the validation dataset given the simulation dataset GMM. The deviation between the two curves is low and the trends of the curves are similar. The similarity between the curves provides an indication that the model performs well.

The median Bayes factor for the ten folds is plotted in Figure 17. The values lie within a range $\{1 < K < 1.7\}$. The average median Bayes factor value over the ten folds is 1.3. The Bayes factor values are all greater than unity as the likelihood of the real-world dataset is naturally greater than the likelihood of the simulated dataset. The Bayes factor values do not deviate far from unity, indicating that the model performs well.

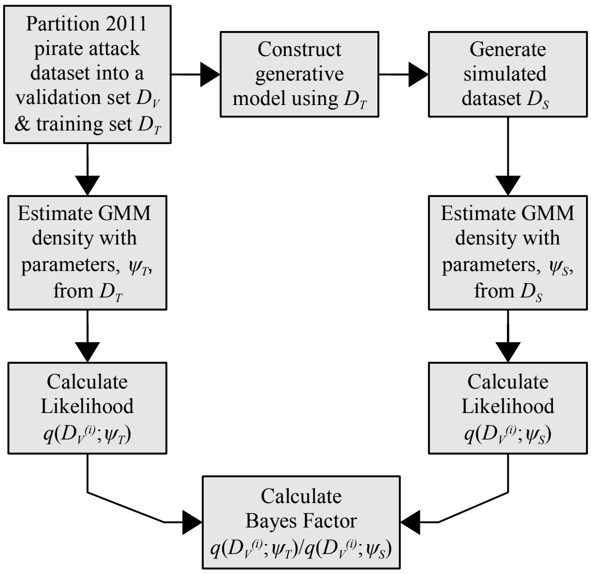

Partition 2011 pirate attack dataset into a validation set D_V & training set D_T
Construct generative model using D_T
Generate simulated dataset D_S
Estimate GMM density with parameters, ψ_T, from D_T
Estimate GMM density with parameters, ψ_S, from D_S
Calculate Likelihood $q(D_V^{(i)};\psi_T)$
Calculate Likelihood $q(D_V^{(i)};\psi_S)$
Calculate Bayes Factor $q(D_V^{(i)};\psi_T)/q(D_V^{(i)};\psi_S)$

Figure 15: Cross validation based verification method for evaluating the proposed model.

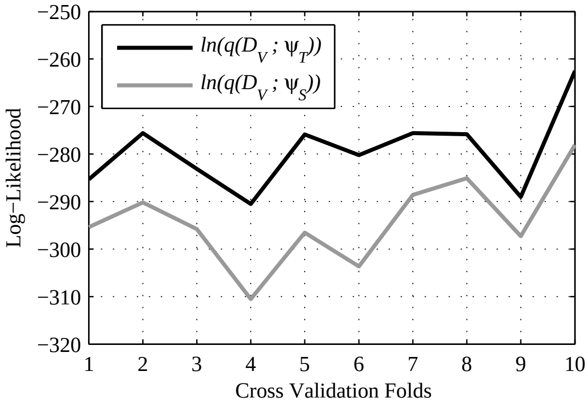


Figure 16: The log-likelihood of the validation dataset given the real-world model $q(D_V;\psi_T)$ and the simulation model $q(D_V;\psi_S)$.

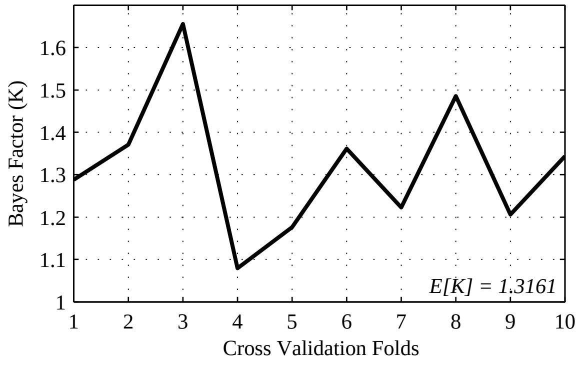


Figure 17: Median value of the Bayes factor given by (19) is plotted for each validation set over ten folds.

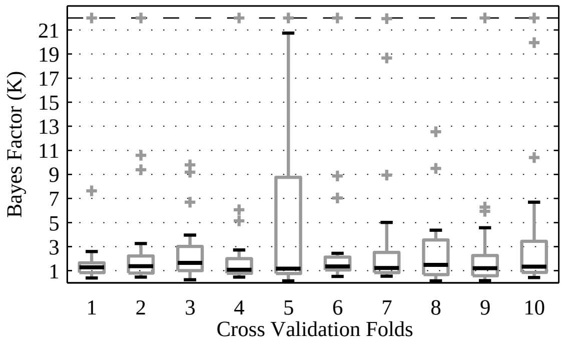


Figure 18: Box plots of Bayes factor for 23 validation samples over 10 cross validation folds.

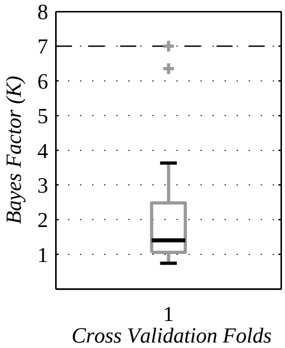


Figure 19: Box plot of Bayes factor for a test set with 1176 samples. This figure may be compared to Figure 18 where the test set for each cross validation fold consists of 212 samples.

The Bayes factor results are presented as box plots in Figure 18. The median Bayes factor is indicated by the central mark in each box. The box upper and lower edges indicate the upper and lower quartiles respectively. Over the ten folds, the Bayes factor quartiles remain within a range of 0.57 and 8.6. The extreme Bayes factor values are represented by the whiskers. The extreme values remain below a value of 21. The majority of extreme values remain below a value of 5. The outlier Bayes factor values are indicated by grey '+' markers. The outliers are constrained to a value of 22 in the plot. The outliers are validation samples that the proposed model is not able to generalise well.

In each cross validation fold, the simulation dataset consists of 212 samples as in the training set. The GMM estimate based on the simulation samples may be improved by increasing the number of simulation samples. To demonstrate this, a simulation dataset consisting of 1176 samples is generated. The box plot for this single cross validation fold is presented in Figure 19. The median Bayes factor value is 1.4. The range of quartiles is reduced to the range of 1.05 and 2.47. This indicates a significant decrease in the variance of the results. The upper quartile indicates that the validation sample given the real-world data GMM is only 2.47 times more likely than the validation sample given the simulated data GMM. This indicates a high level of similarity between the simulation and training dataset models.

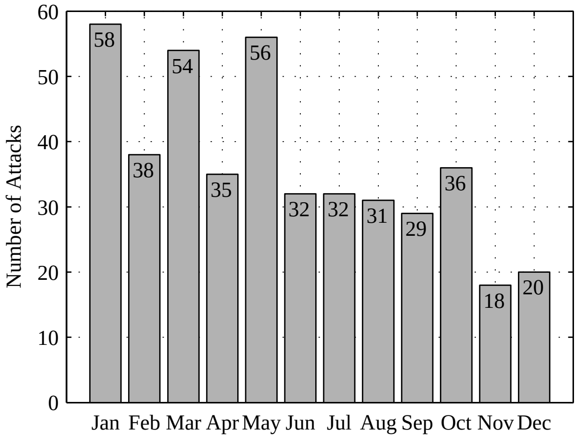


Figure 20: A bar graph describing p_{RW}, the number of attacks that occurred in each month of 2011 [64]. The generative model temporal results are compared to this figure.

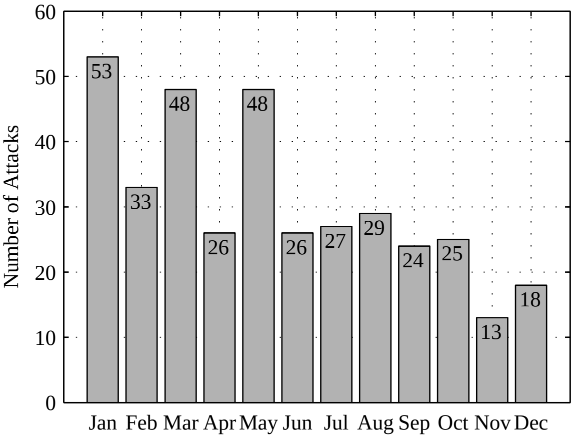


Figure 21: A histogram describing p_{MC}, the number of simulated attacks that occurred in each month, averaged over 10 simulations. This figure may be compared to Figure 20.

6. Temporal Domain Evaluation and Results

The proposed model is evaluated by comparing temporal distributions that describe the real-world pirate data and the simulated data. As indicated in Section 4.3, pirate behaviour is dependent on time. This is evident given histogram presented in Figure 20. The histogram describes the monthly number of attacks over the year, 2011. The real-world temporal distribution illustrated by the histogram is given by

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

To evaluate the proposed model, the generated results may be compared to the distribution, p_{RW}.

To generate results that are comparable to p_{RW}, the temporal behaviour of vessels must be considered. Pirate vessel temporal behaviour is of particular interest. To affect the number of monthly pirate attacks, the amount of time spent at sea of pirate vessels may be adjusted. The pirate state transition probabilities may be considered for this purpose. In particular, the *anchor* to *sail-out* or the *drift* to *sail-home* transition probabilities may be considered. For example, by increasing the *drift* to *sail-home* transition probability, the vessel will spend less time at sea. Fewer pirate attacks will occur if the pirate spends less time searching for targets. Inversely, by decreasing the *drift* to *sail-home* transition probability, the vessel will spend more time at sea, resulting in more pirate attacks. A value may be associated with the *drift* to *sail-home* transition probability for each month in a year. For the model to generate data that is comparable to p_{RW}, the transition probabilities may be set as values that are proportional to p_{RW}. In this study, the *drift* to *sail-home* transition probability for month $i = \{1, \ldots, 12\}$ is given by:

$$p(\textit{sail-home}|\textit{drift}, i) = \frac{p_{RW}(i)}{\max(p_{RW}) + \min(p_{RW})}. \qquad (21)$$

The denominator in (21) does not affect the temporal distribution. It affects the total number of attacks that occur throughout the year.

A set of 10 simulations are performed. Each simulation epoch represents a year. In each simulation, the monthly number of attacks is accumulated. A set of 10 temporal distributions are formed, $p_S^{(j)}$, $j = \{1, \ldots, 10\}$. The Monte Carlo temporal distribution, p_{MC} is defined as the average monthly attacks over the set of 10 simulations. That is, each of the 12 elements in p_{MC} contains the average number of pirate attacks over the 10 simulations for each month in a year. Each element in p_{MC} is given by:

$$p_{MC}(i) = \frac{1}{10} \sum_{j=1}^{10} p_S^{(j)}(i), \qquad (22)$$

where $i = \{1, \ldots, 12\}$. The histogram of p_{MC} is illustrated in Figure 21. This histogram may be compared to the histogram illustrated in Figure 20.

The Bhattacharyya coefficient between two discrete distributions p_1 and p_2 over some variable i, is defined as [74, 75]:

$$\rho(p_1, p_2) = \sum_i \sqrt{p_1(i) p_2(i)}. \qquad (23)$$

The Bhattacharyya coefficient may be explained as the cosine of the angle between the unit vectors formed with p_1 and p_2. This may be used as a measure for comparing the generated and real-world temporal distributions. The Bhattacharyya coefficient for each of the 10 simulations $\rho(p_S^{(j)}, p_{RW})$, $j = \{1 \ldots 10\}$, is plotted in Figure 22. The values of the Bhattacharyya coefficient are in the range $\{0.993 < \rho(p_S^{(j)}, p_{RW}) < 0.997\}$. The range of values is near unity indicating a high level of similarity between the generated and real-world distributions. The similarity may be improved using the Monte Carlo approach. The Bhattacharyya coefficient between p_{MC} and p_{RW} is calculated as

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

As indicated in Section 7.2, the Bhattacharyya coefficient may be explained as the cosine of an angle between the two distributions. The 'Bhattacharyya angle' between p_{MC} and p_{RW}, for $\rho(p_{MC}, p_{RW}) = 0.9991$, is 2.431°.

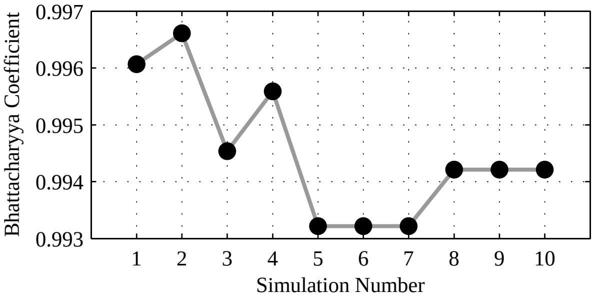


Figure 22: The Bhattacharyya coefficient $\rho(p_{RW}(i), p_{RW})$, $i = 1 \ldots 10$, comparing the 2011 temporal distribution and the 10 simulated temporal distributions.

7. Model Effectiveness

The effectiveness of an information fusion system may be described according to robustness, quality and information gain [76]. Robustness measures the consistency of the model. Quality measures the performance of the model. Information gain measures the ability of the model to provide improvement.

7.1. Model Robustness

The robustness of the model may be described by ability of the model to generalise the data. A model that is not able to generalise data is not considered to be robust. The ability for the model to generalise data is demonstrated by the cross validation results presented in Section 5.7. The exceptional Bayes factor results indicate a high level of robustness of the model.

The simulated pirate attack locations are not required to be identical to real-world attack locations. A robust model will maintain the form of the spatial distribution while providing a level of uncertainty on the attack locations. The proposed model produces varying results while maintaining the required structural form of the distribution.

The results illustrated in Figure 14 also provide an indication of the robustness of the model. The standard deviation of the model for each σ is described by the error bars. The log-likelihood seems to vary more over the σ parameter values than over different simulation instances with the same σ value. This is particularly true for $\sigma \leq 5$. This implies that the model is more strongly affected by the parameter selection than model uncertainty. Furthermore, the standard deviation values remain similar between the various model parameters.

7.2. Model Quality

The quality of the model may be described by the similarity between the simulated data and the real-world data. The temporal results presented in Section 6, demonstrate a high level of similarity between the simulated and real-world data. The high level of similarity indicates a high level of quality of the model.

The likelihood results discussed in section 5.5 provide an indication of the quality of the model. The maximum likelihood value indicates the model with highest quality. The Bhattacharyya distance may be considered as a simpler and more intuitive measure than the likelihood. In this case, the Bhattacharyya distance is however a less rigorous measure.

Table 2: Pirate attack location spatial distribution comparisons between the simulated results and the 2011 attacks. The Bhattacharyya distance and the Bhattacharyya coefficient 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 x 4600km.

Bin Size (pixels)	$\rho(p_1, p_2)$	$D_B(p_1, p_2)$
25kmx25km	0.0358	0.9819
100kmx100km	0.2702	0.8543
250kmx250km	0.6684	0.5758
400kmx400km	0.7988	0.4485
500kmx500km	0.8632	0.3699

The Bhattacharyya coefficient between two discrete distributions p_1 and p_2 is given by (23). The Bhattacharyya distance between discrete distributions p_1 and p_2 may be calculated as [74, 75]:

$$D_B(p_1, p_2) = \sqrt{1 - \rho(p_1, p_2)}. \tag{24}$$

The discrete distributions of the pirate attack locations are determined using two dimensional histograms. The maps are divided into square cells to form the histogram bins. For this evaluation, let the distribution p_1 represent the histogram determined from the simulated data. Let the distribution p_2 represent histogram determined from the 2011 pirate attack data. Results of distances between the spatial distributions for various histogram bin sizes are provided in Table 2. For small histogram bin sizes, the distributions appear unrelated. For large histogram bin sizes, the distributions are more similar. The results describe the scale at which the model performance becomes acceptable. The model performance becomes acceptable around the 250kmx250km region.

7.3. Information Gain (Cross Entropy)

The information gain may be considered as the information that is required to be gained for the simulated distribution to match the real-world data distribution. Entropy is a measure that is used to describe the level of information. To compare different probabilistic models, cross entropy may be used [77]. In language processing, a sequence of words or parts of speech may be compared with a particular model using cross entropy [77]. In this study, cross entropy is used to compare the real world dataset with the proposed model. Suppose some natural probability distribution $p(y)$ generated the real world dataset within the space $y \in \mathcal{Y}$. The GMM distribution, $q(y; \psi_S)$ models the simulated dataset. The cross entropy between $p(y)$ and $q(y; \psi_S)$, is given as [77]:

$$H(p, q) = -\int p(y) \log(q(y; \psi_S))dy \tag{25}$$

Given the set of real world data $\bar{b}$, the cross entropy is given by [77]:

$$H(p, q) = \lim_{n\to\infty} -\frac{1}{n} \sum_{j}^{n} p(\bar{b}_j) \log(q(\bar{b}_j; \psi_S)) \tag{26}$$

Table 3: Cross entropy (27) in nats for the set of simulations over various values of σ.

σ	$H(p,q)$
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 Shannon-McMillan-Breiman theorem dictates that, for a stationary ergodic process, the cross entropy may be written as follows [77]:

$$H(p,q) = \lim_{n\to\infty} -\frac{1}{n}\sum_{j}^{n} \log(q(\bar{b}_j;\psi_S)) \tag{27}$$

This equation may be interpreted as describing the information of the GMM distribution evaluated at the sample points of the real world dataset. If the real-world dataset does not correlate well with the GMM distribution, then the information will be high. The GMM requires a higher amount of information to 'encode' the real-world dataset.

The cross entropy values for the various simulation parameters are presented in table 3. The simulation parameters are described in section 5.5. The cross entropy values are displayed in nats. The minimum cross entropy value corresponds to the covariance for $\sigma = 6$. This result agrees with the maximum likelihood value displayed in table 1. The minimum cross entropy value indicates that the corresponding simulation parameters provide a model that requires the least amount of information to be gained to represent the real-world dataset.

It may be noted that the cross entropy given in (27) is similar in form to the log likelihood given in (17). The cross entropy in (27) may be considered as the negated normalised likelihood given in (17). The maximum likelihood method described for optimisation is equivalent to the minimisation of the cross entropy. This provides a form of validation of the optimisation procedure.

8. Future Research and Applications

The implementation of the model is to be refined using real world data and statistics. The EP variable may be expanded to include parameters such as wave height, wind speed, wind direction, cloud cover and air temperature. Additional parameters may be varied in the optimisation of the model as discussed in section 5.5.

The proposed model is developed for the purpose of simulation and data generation. The data generated by this model shall be utilized for the purpose of developing and testing maritime pirate detection algorithms. Research is being conducted on using a DBN for classification of vessels in the maritime environment using the generated data. The DBN classifier is a generalised variation of the proposed model where the class variable is inferred.

The data generated by the proposed model may be utilised in various other applications. For example, the proposed model may be used for multi-sensor simulation. The proposed model contains a plate of sensor variables. The sensor variables could be used in modelling a set of particular sensors. The data generated by the sensors could be used for testing and evaluating multi-sensor information fusion methods. The model is not limited to the maritime piracy application. The variables in the proposed model can be adapted to be applied to other applications such as land or air based applications.

The authors intend to integrate the proposed model into the ICODE-MDA open source tool for maritime domain awareness [78].

9. Summary and Conclusion

A multi-agent generative model is proposed for the purpose of simulating a maritime piracy situation. The model comprises of a SLDS represented in the form of a DBN. The DBN describes a Markovian state based model that determines the behaviour and motion of the modelled vessel. The states of the model are determined by a set of higher level variables whose probability distributions may be inferred from data. The proposed DBN thus provides a versatile model that unifies physical, graphical and probabilistic attributes to model behaviour.

The proposed model is modelled and evaluated with respect to the attack locations of 2011. Optimisation and evaluation is conducted based on likelihood computations of the real-world pirate attack data with respect to the simulated data. Furthermore, method of cross validation is performed using Bayes factor. The temporal modelling capability of the proposed model is demonstrated. The temporal results are evaluated using the Bhattacharyya coefficient.

The model effectiveness is measured according to quality, robustness and information gain. Cross entropy is utilised to describe the information gain. The likelihood and Bhattacharyya distance is used to describe the quality. The robustness of the model is measured by the consistency of the model and the cross validation results. The proposed model is able to produce unique and varying results while maintaining the structural integrity of the general spatial and temporal distributions. The distributions that were produced correlate well with the real-world data. The evaluation and optimisation methodology may be applied to other behavioural modelling applications.

A possible deficiency of the model is that the DBN is subject to the curse of dimensionality. The conditional distributions for each link in the DBN are required to be defined. If many contextual elements and many vessel classes are considered, the DBN may become cumbersome to configure. This is addressed in this study by combining contextual elements to reduce the number of variables.

The data generated by the model may be utilised for various applications. The intended use of the data is for training, testing and evaluating threat assessment methods. Machine learning methods may require training data. Statistical methods may require prior information. In general, most algorithms and methods will require testing and evaluation. The model is able to produce unique and varying results while maintaining the structure of a spatial distribution. This quality is desirable for producing realistic results and for generating suitable data for training, testing and evaluation.

Acknowledgement

This work was supported by the Advanced Sensors and Electronics Defence (ASED) Centre of KACST through the Council for Scientific and Industrial Research (CSIR).

The maps used in the illustrations presented in this work are obtained from Google Maps [79] under the fair use principle.

References

[1] M. Jakob, O. Vaněk, M. Pěchouček, Using agents to improve international maritime transport security, Intelligent Systems, IEEE 26 (2011) 90–96.

[2] S. Percy, A. Shortland, The business of piracy in somalia, Journal of Strategic Studies 0 (0) 1–38.

[3] M. Wooldridge, An Introduction to Multiagent Systems, John Wiley and Sons, Ltd, 2008.

[4] Y. Shoham, K. Leyton-Brown, Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations, Cambridge University Press, 2009.

[5] C. M. Macal, M. J. North, Tutorial on agent-based modeling and simulation, in: Proceedings of the 37th conference on Winter simulation, WSC '05, Winter Simulation Conference, 2005, pp. 2–15.

[6] G. Weiss, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, Intelligent Robotics and Autonomous Agents Series, The MIT Press, 1999.

[7] T. Cioppa, T. Lucas, S. Sanchez, Military applications of agent-based simulations, in: Simulation Conference, 2004. Proceedings of the 2004 Winter, volume 1, 2004, pp. –180. doi:10.1109/WSC.2004.1371314.

[8] S. M. Sanchez, 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, WSC '02, Winter Simulation Conference, 2002, pp. 116–126.

[9] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Representation and Reasoning Series, Morgan Kaufman, 1988.

[10] T. Dean, K. Kanazawa, A model for reasoning about persistence and causation, Computational Intelligence 5 (1989) 142–150.

[11] Y. Luo, T.-D. Wu, J.-N. Hwang, Object-based analysis and interpretation of human motion in sports video sequences by dynamic bayesian networks, Computer Vision and Image Understanding 92 (2003) 196 – 216. Special Issue on Video Retrieval and Summarization.

[12] P. Wiggers, B. Mertens, 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, CompSysTech '11, ACM, New York, NY, USA, 2011, pp. 411–416. doi:10.1145/2023607.2023676.

[13] A. Petrovskaya, S. Thrun, Model based vehicle detection and tracking for autonomous urban driving, Autonomous Robots 26 (2009) 123–139.

[14] S. Das, High-Level Data Fusion, Artech House electronic warfare library, Artech House, Incorporated, 2008.

[15] J. Poropudas, K. Virtanen, Influence diagrams in analysis of discrete event simulation data, in: Winter Simulation Conference, WSC '09, Winter Simulation Conference, 2009, pp. 696–708.

[16] J. Poropudas, K. Virtanen, Simulation metamodeling in continuous time using dynamic bayesian networks, in: Proceedings of the Winter Simulation Conference, WSC '10, Winter Simulation Conference, 2010, pp. 935–946.

[17] J. Poropudas, K. Virtanen, Simulation metamodeling with dynamic bayesian networks, European Journal of Operational Research 214 (2011) 644 – 655.

[18] Y. Bar-Shalom, X. Li, Estimation and Tracking: Principles, Techniques, and Software, The Artech House radar library, Artech House, Incorporated, 1993.

[19] K. P. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning, Ph.D. thesis, University of California, Berkeley, 2002.

[20] D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.

[21] V. Pavlovic, J. Rehg, T.-J. Cham, 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, volume 1, 1999, pp. 94–101 vol.1. doi:10.1109/ICCV.1999.791203.

[22] C.-J. Kim, Dynamic linear models with markov-switching, Journal of Econometrics 60 (1994) 1 – 22.

[23] B. Mesot, D. Barber, Switching linear dynamical systems for noise robust speech recognition, Audio, Speech, and Language Processing, IEEE Transactions on 15 (2007) 1850–1858.

[24] H. Wehn, R. Yates, P. Valin, A. Guitouni, E. Bosse, A. Dlugan, H. Zwick, A distributed information fusion testbed for coastal surveillance, in: Information Fusion, 2007 10th International Conference on, 2007, pp. 1–7. doi:10.1109/ICIF.2007.4408089.

[25] M. Kruger, L. Ziegler, 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.

[26] F. Fooladvandi, C. Brax, P. Gustavsson, 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.

[27] P. C. G. Costa, K. B. Laskey, K.-C. Chang, W. Sun, C. Y. Park, , S. Matsumoto, High-level information fusion with bayesian semantics, in: UAI 9th Bayesian Modeling Applications Workshop, Catalina Island, CA, 2012, pp. –. Held at the Conference of Uncertainty in Artificial Intelligence (BMAW UAI 2012).

[28] R. Carvalho, R. Haberlin, P. Costa, K. Laskey, 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.

[29] Y. Zhang, Q. Ji, Active and dynamic information fusion for multisensor systems with dynamic bayesian networks, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 36 (2006) 467–472.

[30] G. Yang, Y. Lin, P. Bhattacharya, A driver fatigue recognition model based on information fusion and dynamic bayesian network, Information Sciences 180 (2010) 1942 – 1954. Special Issue on Intelligent Distributed Information Systems.

[31] V. I. Pavlovic, Dynamic Bayesian Networks for information fusion with applications to human-computer interfaces, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1999.

[32] T. Strang, 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. –.

[33] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive and Mobile Computing 6 (2010) 161 – 180. Context Modelling, Reasoning and Management.

[34] L. S. Kennedy, 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, CIVR '07, ACM, New York, NY, USA, 2007, pp. 333–340. doi:10.1145/1282280.1282331.

[35] J. Gómez-Romero, M. A. Serrano, M. A. Patricio, J. García, J. M. Molina, Context-based scene recognition from visual data in smart homes: an information fusion approach, Personal and Ubiquitous Computing 16

(2012) 835–857.

[36] A. Steinberg, G. Rogova, Situation and context in data fusion and natural language understanding, in: Information Fusion, 2008 11th International Conference on, 2008, pp. 1–8.

[37] J. Garcia, J. Gomez-Romero, M. Patricio, J. Molina, 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.

[38] R. Hegde, J. Kurniawan, B. Rao, On the design and prototype implementation of a multimodal situation aware system, Multimedia, IEEE Transactions on 11 (2009) 645–657.

[39] J. George, J. Crassidis, 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.

[40] O. Sekkas, S. Hadjiefthymiades, 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. doi:10.1109/PIMRC.2006.254053.

[41] O. Sekkas, C. B. Anagnostopoulos, S. Hadjiefthymiades, Context fusion through imprecise reasoning, in: Pervasive Services, IEEE International Conference on, 2007, pp. 88–91. doi:10.1109/PERSER.2007.4283896.

[42] C. Anagnostopoulos, O. Sekkas, 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. doi:10.1109/MOBHOC.2007.4428752.

[43] O. Vaněk, M. Jakob, O. Hrstka, B. Bošanský, M. Pěchouček, Agentc: Fighting maritime piracy using data analysis, simulation and optimization, 2013. URL: http://agentc-project.appspot.com/.

[44] European Commission: Joint Research Centre., Blue hub - integrating maritime surveillance data, 2013. URL: https://bluehub.jrc.ec.europa.eu/.

[45] R. Middleton, Piracy in somalia: Threatening global trade, feeding local wars, Chatham House, 2008. Briefing Paper.

[46] C. Bueger, J. Stockbruegger, S. Werthes, Pirates, fishermen and peacebuilding: Options for counter-piracy strategy in somalia, Contemporary Security Policy 32 (2011) 356–381.

[47] M. Heger, J. Oberg, M. Dumiak, S. Moore, P. Patel-Predd, Technology vs. pirates, Spectrum, IEEE 46 (2009) 9–10.

[48] E. Bossé, E. Shahbazian, G. Rogova, Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems, NATO Science for Peace and Security Sub-Series E: Human and Societal Dynamics, IOS Press, Incorporated, 2013.

[49] O. Vaněk, B. Bošanský, M. Jakob, M. Pěchouček, Transiting areas patrolled by a mobile adversary, in: Computational Intelligence and Games (CIG), 2010 IEEE Symposium on, 2010, pp. 9–16. doi:10.1109/ITW.2010.5593377.

[50] O. Vaněk, M. Jakob, V. Lisý, B. Bošanský, 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, AAMAS '11, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2011, pp. 1273–1274.

[51] C. D. Marsh, Counter Piracy: A Repeated Game with Asymmetric Information, Master's thesis, Naval Postgraduate School, 2009.

[52] J. C. Sevillano, D. Rios Insua, J. Rios, Adversarial risk analysis: The somali pirates case, Decision Analysis 9 (2012) 86–95.

[53] H. Liwang, J. W. Ringsberg, M. Norsell, Quantitative risk analysis: Ship security analysis for effective risk control options, Safety Science 58 (2013) 98 – 112.

[54] G. Baldini, D. Shaw, F. Dimc, A communication monitoring system to support maritime security, in: ELMAR, 2010 PROCEEDINGS, 2010, pp. 243–246.

[55] M. Teutsch, W. Kruger, Classification of small boats in infrared images for maritime surveillance, in: Waterside Security Conference (WSS), 2010 International, 2010, pp. 1–7. doi:10.1109/WSSC.2010.5730289.

[56] J. G. Sanderson, M. K. Teal, 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), volume 2, 1999, pp. 803–807 vol.2. doi:10.1049/cp:19990435.

[57] L. Esher, S. Hall, E. Regnier, P. Sanchez, J. Hansen, D. Singham, Simulating pirate behavior to exploit environmental information, in: Simulation Conference (WSC), Proceedings of the 2010 Winter, 2010, pp. 1330–1335. doi:10.1109/WSC.2010.5679060.

[58] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, Adaptive Computation and Machine Learning, MIT Press, 2009.

[59] J. F. Verner, T. D. Nielsen, Bayesian Networks and Decision Graphs, Information Science and Statistics, Springer, 2007.

[60] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, Intelligent robotics and autonomous agents series, first ed., MIT Press, United States of America, 2006.

[61] K. Murphy, Switching kalman filters, Technical Report, Dept. of Computer Science, University of California, Berkeley, 1998.

[62] L. R. G. Limited, Lloyd's register of ships online, 2013. URL: http://www.lr.org/about_us/shipping_information/Lloyds_Register_of_Ships_online.aspx.

[63] AAPA, The american association of port authorities (aapa) website, 2013. URL: http://www.aapa-ports.org/.

[64] ICC-IMB, ICC-IMB Piracy and Armed Robbery Against Ships Report - Annual Report 2011, Annual Report, ICC International Maritime Bureau, 2012.

[65] B. White, K. Wydajewski, Commercial ship self defense against piracy and maritime terrorism, in: OCEANS '02 MTS/IEEE, volume 2, 2002, pp. 1164–1171 vol.2. doi:10.1109/OCEANS.2002.1192131.

[66] 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, ICLARM conference proceedings 5, International Center for Living Aquatic Resources Management, 1980, pp. 32–56.

[67] R. Lane, D. Nevell, S. Hayward, T. Beaney, Maritime anomaly detection and threat assessment, in: Information Fusion (FUSION), 2010 13th Conference on, 2010, pp. 1–8.

[68] G. McLachlan, D. Peel, Finite Mixture Models, John Wiley and Sons, inc, Canada, 2000.

[69] Expedition, Somalia pirate activity areas, 2013. URL: http://productforums.google.com/forum/m/#!msg/gec-current-events/x0_XPmUe8HA/x5deCf5gtQ0J.

[70] W. Martinez, A. Martinez, Computational Statistics Handbook with MATLAB, Chapman & Hall/CRC Computer Science & Data Analysis, Taylor & Francis, 2001.

[71] S. Salvador, 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, 2004, pp. 576–584. doi:10.1109/ICTAI.2004.50.

[72] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, second ed., Wiley-Interscience, 2000.

[73] S. Russell, P. Norvig, Artificial Intelligence A Modern Approach, third ed., Pearson, Upper Saddle River, New Jearsey 07458, 2010.

[74] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, volume 2, 2000, pp. 142–149 vol.2. doi:10.1109/CVPR.2000.854761.

[75] A. Bhattacharyya, On a measure of divergence between two statistical populations defined by their probability distributions, Bulletin of the Calcutta Mathematical Society 35 (1943) 99–109.

[76] E. Blasch, P. Valin, E. Bosse, Measures of effectiveness for high-level fusion, in: Information Fusion (FUSION), 2010 13th Conference on, 2010, pp. 1–8.

[77] D. Jurafsky, J. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall series in artificial intelligence, Pearson Prentice Hall, 2009.

[78] ICODE-MDA, icode-mda website, 2013. URL: https://code.google.com/p/icode-mda/.

[79] Google Inc., Google maps, 2013. URL: maps.google.com.