

Learning Abnormal Vessel Behaviour from AIS Data with Bayesian Networks at Two Time Scales*

Steven Mascaro, Kevin B. Korb and Ann E. Nicholson

Clayton School of Information Technology, Monash University

August 31, 2010

Abstract

In recent years, electronic vessel tracking has provided abundant data on vessel movements to surveillance authorities. Researchers have begun looking at the use of this data for anomaly detection using a wide variety of data mining techniques. Here we tackle anomaly detection with Bayesian Networks, training them with real world AIS data and producing models at two different time scales — both moment to moment and for the track as a whole. The networks also incorporate additional real world data, including weather, vessel details and details about vessel interactions. We find that the generated networks are quite easy to examine and verify despite incorporating a large number of variables; that combining models at the two different scales improves performance in a variety of cases; and, ultimately, that Bayesian Networks prove a promising approach to anomaly assessment and detection.

1 Introduction

In recent years, a wealth of information on vessel movements has become available to surveillance authorities through the use of the Automated Identification System (AIS). This data has even filtered through to the general public via the Internet (though not without controversy; for example, see Maritime Safety Committee, 1994). Surveillance authorities are interested in using this data to uncover activities of concern, activities that may involve threats to security, illegal trafficking or risks to the safety of other vessels. While in the past, surveillance efforts have suffered from a lack of solid data, today electronic tracking has transformed the problem into one of too much data. In short, automated identification has led to the need for automated analysis.

Typically, the goal of vessel behaviour analysis is to identify anomalies. This requires the development of a model representing normal vessel behaviour; anomalous behaviour is then identified by the degree to which a vessel's motion does *not* conform to that model of normalcy. Researchers use many different machine learning techniques to generate normalcy models from vessel movement data (typically AIS data), and the models are commonly specified in the language of Gaussian Mixture Models (Laxhammar, 2008), Support Vector Machines (Li et al., 2006), neural networks and others.

*The authors would like to thank: the Intelligence, Surveillance and Reconnaissance division within the DSTO for supporting this work and also for providing the AIS data used; Göran Falkman and especially Fredrik Johansson for providing access to their data and generously answering many questions about their approach; and SAAB Microwave Systems for permitting us the use of their simulated data.

Here, we explore the use of Bayesian Networks (BNs) for analysing vessel behaviour and helping in the detection of anomalies. BNs have generally received less attention in maritime anomaly detection research, though there have been a few preliminary investigations along these lines (Johansson and Falkman, 2007; Helldin and Riveiro, 2009). As noted by Johansson and Falkman (2007), BNs have the advantages of being easily understood and allowing for the inclusion of expert knowledge. They are also capable of representing causal relations directly and, given these properties, have the advantage of being more easily verified and validated. To generate our BNs, we employ the CaMML machine learning tool (Korb and Nicholson, 2004), working with AIS data supplied by the Australian Defence Science and Technology Organisation (DSTO). (See Appendices B-E for brief introductions to Bayesian networks and CaMML.)

Since many factors can contribute to the (ab)normality of a vessel’s behaviour, we have expanded the number of attributes beyond the typical set associated with movement. In particular, we incorporate information related to the ship (including type, dimensions and weight), weather (such as temperature, cloud cover and wind speed) and temporal factors (including hour of day and time since dawn or dusk). We have also added rudimentary elements to the model that aim to capture the effect of vessel interactions.

We pursue two different approaches to the model learning problem. In the first, more conventional, approach, we train a model on the track data in its original time series form (called here the *time series approach*). In addition to incorporating the factors described in the previous paragraph into the time series model, we add DBN elements (dynamic Bayesian network; see Appendix C for a discussion) to the network for those variables related to motion. By contrast in the second approach, we create a single summary record of an entire track (called here the *track summary approach*). This summary will include information such as average speed and course, number of stops, major stopping points and percentage of time spent travelling straight, amongst others.

To assess the value of the networks in anomaly detection we take the common approach of providing a measure for how probable a given track is according to the generated models of normalcy. We then use this measure to test how the networks perform with sets of normal and anomalous data. In addition, we also permute the presumed normal tracks to help us see how the network’s probability estimates change. This leads to a very interesting understanding of both the network’s behaviour and the nature of the normal data set. It also suggests the value that an abductive approach — roughly speaking, working backwards to find the best explanation for the data — may have to anomaly detection.

In the following section, we take a look at some of the related research on anomaly detection and vessel behaviour analysis. In Section 3, we describe our method in greater detail, including a description of the two main pre-processing approaches — the time series and track summary approaches — the variables used by the BNs and the methods used for learning. This is followed in Section 4 by an analysis of the resultant models and a discussion of how the models can be used to aid in anomaly detection.

2 Background

The typical approach to both vessel behaviour analysis and anomaly detection involves generating a model of normalcy from a set of features in vessel track data. As noted by Laxhammar (2008), the general aim of this approach is to cluster the data around a set of points in a multi-dimensional feature space, where the features of the track are items such as longitude and latitude, speed and course. Tracks that are found to sit in

or close to one of these clusters may be considered normal tracks, while those that sit at a larger distance from all the clusters may indicate an anomaly.

There are two important choices to be made when following this approach. The first is the choice of model representation, which is much like choosing a suitable language. Models of normal vessel behaviour can be represented in many different forms, with varying degrees of expressiveness, including Support Vector Machines, Gaussian Mixture Models, Kernel Density Estimators, neural networks and Bayesian Networks. The second choice to be made is that of the machine learner or learning technique, which will create the model (whether an SVM, GMM, etc.) based on the training data.

2.1 Common model representations

Support Vector Machines (SVMs) partition the multidimensional space and so produce strict boundaries between clusters. In their simplest forms, SVMs suffer from a number of limitations — such as lack of partial assignment, only binary classes can be directly modelled, high computational complexity in some cases and difficulties in summarising and communicating the learnt models — and are generally less used amongst vessel anomaly detection researchers. Li et al. (2006), however, make use of SVMs to perform an interesting analysis of vessel behaviour at a higher level of abstraction than that of the time series. Li et al. extract higher level movement features from the track (such as turn left, or loop) and then cluster these further into what they call “movement motifs”. They show that an SVM trained on the movement motif abstractions can correctly classify a significantly higher percentage of their test data in some cases (specifically, when the level of abstraction was not too high) than an SVM trained on lower level features alone.

One commonly used model is the neural network (Rhodes et al., 2007, 2005), which consists of a network of processing nodes, input/output connections between nodes and weights attached to the connections. For the purposes of anomaly detection, neural networks are typically used to map an input vector of reals to an output in the form of a classification. When used in this way, a neural network partitions the feature space much like an SVM. Unfortunately, neural networks suffer a number of drawbacks, the most significant being that trained models of any moderate degree of complexity are almost completely opaque to human understanding.

Gaussian Mixture Models (GMMs) have proven a popular choice for representing normalcy models of vessel behaviour (Kraiman et al., 2002; Laxhammar, 2008; Laxhammar et al., 2009). As its name implies, a GMM is a combination of usually multi-variate Gaussian distributions. These distributions aim to summarise how the training data cluster and spread around points in the multi-dimensional space. Kernel Density Estimators (KDEs) are a generalisation of GMMs involving non-parametric density functions (with a smoothing parameter, h). In using a sum of (typically Gaussian) distributions for each point, they allow for more flexibility than GMMs in the way clusters are described. Unfortunately, both GMM and KDE models can be difficult for non-technical experts to understand. Laxhammar et al. (2009) trained both GMMs and KDEs on AIS data and evaluated anomaly detection performance by stochastically generating anomalous tracks, and then measuring how many steps it took for each method to flag the track as anomalous. It is of interest to note that they found little extra value in using KDE methods over GMMs, though as the authors suggest, this may have been due to the granularity of their geographical cells.

2.2 Use of Bayesian Networks

Recently, Bayesian Networks have been applied to the problem. Johansson and Falkman (2007) note that BNs have two main advantages over other types of model commonly used: 1) BN models are easily understood by non-specialists and 2) they allow for the straightforward incorporation of expert knowledge. We also would add two important points: 3) BNs are capable of representing causal relations directly in the network and 4) due to these three points together, BNs are much more easily verified and validated than other types of model, as we show in Sections 3 and 4.

A BN consists of a set of nodes (also called variables or attributes), $V = \{v_i\}$, connected together with directed edges $E = \{e_i\}$ in a directed acyclic graph, G . Thus, each v_i (or u for brevity) can have either parent or child nodes or both and the network must have at least one root node (nodes with no parents) and at least one leaf node (nodes with no children). Each variable u can take on a set of values (or states) u_s which are typically discrete for computational purposes, though can also be continuous. The set of states u_s for a variable has associated with it a local probability distribution that is conditional only on the variable's parent nodes, $Y_u = \{y_{u,i}\}$, such that $P(u_s | Y_u = c)$ is defined for each state u_s of u , and each possible combination, c , of the parent nodes' states. The network as a whole represents the joint distribution over its variables, but can take advantage of the conditional independencies in the data to help reduce the complexity of the network and thereby produce a much more compact representation of the joint than a full enumeration of combinations.¹

Johansson and Falkman (2007) describe the use of BNs for anomaly detection and perform a preliminary experiment in which they train a BN on simulated training data representing normal vessel behaviour. Their training data is generated using the GT-SIM simulator² and (once pre-processed) includes seven variables — x, y, speed, speed delta, heading, heading delta and vessel type. They use the constraint-based PC algorithm (Spirtes and Glymour, 1991) to learn the structure of the BN and make use of the usual counting procedure on the training data to parameterise it. To detect anomalies, they first take the observations for a vessel track at time t , and then use the learnt BN to find the joint probability of the observed variables. This is then averaged over a window of k timesteps around t . If, for a window around t in a given track, the average probability falls *below* some threshold, δ , then the system flags the track as anomalous.

It is difficult to assess how well their BN model performs in this experiment as Johansson and Falkman do not indicate what the false (or true) positive rates might be, nor do they examine how their parameters affect anomaly detection. They do note that their approach manages to identify a reasonable proportion of the anomalous tracks, while missing others, but do not give specifics. Nonetheless, their experiment serves as a useful preliminary investigation into the value of BNs for anomaly detection and as a guideline for future work.

Helldin and Riveiro (2009) also look at the use of BNs in anomaly detection, but they focus specifically on how the reasoning capabilities of a BN can assist surveillance system operators. They examine how BNs can help operators by not only flagging potential anomalies, but also providing an explanation for the anomaly identification in the form of Explanation and Causal Explanation Trees (Nielsen et al., 2008).

As can be seen, there is still much work to be done in investigating the value of BNs in the area of anomaly detection, and we take up some of this work here.

¹For a more detailed introduction to BNs, see Appendix B.

²The GTSIM simulator was created by Saab Microwave Systems to generate realistic vessel and car tracks, and to simulate the performance of various radar and other tracking technology. Johansson, Falkman and their colleagues, however, have since moved to using real world AIS data almost exclusively.

3 Approach

At a broad level, our approach to creating normalcy model BNs is quite similar to that described by Johansson and Falkman (2007). Vessel track data is first pre-processed before being fed into our machine learner as training data. The machine learner (called CaMML and described below) produces normalcy models that are then used to estimate the probability of each member in a set of test tracks.

At a more detailed level, our approach diverges a little from past work, which can be seen as we proceed below. Figure 1 shows an outline of our approach.

3.1 The Data

The DSTO provided real world AIS data covering the period between May 1st and July 31st, 2009 for a section of the NSW coast framing Sydney harbour (see Figure 2). The data initially contained eight fields: the vessel’s MMSI (Maritime Mobile Service Identity — a nine digit numerical vessel identifier), a timestamp, the latitude and longitude of the vessel, the vessel’s reported speed, course and heading and the navigational status. An example of the data can be seen in Figure 3. After examining the navigation status field, we removed it from the data as in most cases it contained either invalid information, or information already covered more reliably by fields such as speed and course. In the final networks, we also drop the MMSI and Timestamp variables, as (in their original forms) they provide no useful state information for the BNs.³

3.2 Pre-processing

The first step in the pre-processing phase involves cleaning and separating the AIS data into tracks. This is achieved by first assigning each record to a separate track based on the MMSI field. We then clean the data in each track by rounding (and interpolating) each row to the nearest 10 second interval and eliminating duplicate data. However, since the raw data contains many cases in which a single vessel transmits for much of the three month period of the data, further track splitting is required. We decided that for any period in which the vessel is stopped or not transmitting for 6 hours or more indicates a point at which to split the track. In future, we may improve this technique so that, for example, the track is split only when the vessel is stopped near land for long periods.⁴ Using this method yielded 2,473 tracks across 544 unique MMSIs.

After our initial investigations, we settled on producing two different kinds of model based on two different arrangements of the training data: these are given by the time series and track summary approaches, described below.

Time series The first of the pre-processing approaches leaves the data as a time series. Thus, each timestep in a track is associated with a set of variables, such as latitude, longitude, speed and so on, that have corresponding nodes in the BN. This approach, of course, has the advantage that generated models can be used in online analysis situations but may miss patterns at a broader time scale.

³We do extract more general temporal information from the timestamp, such as hour of day and day of week. It should also be noted that the MMSI id itself does exhibit certain patterns related to the type of ship, but we ignore this here.

⁴Indeed we feel that the manner in which tracks are split is crucial to the model’s ability to identify anomalies, given that such stops may themselves indicate an anomaly.

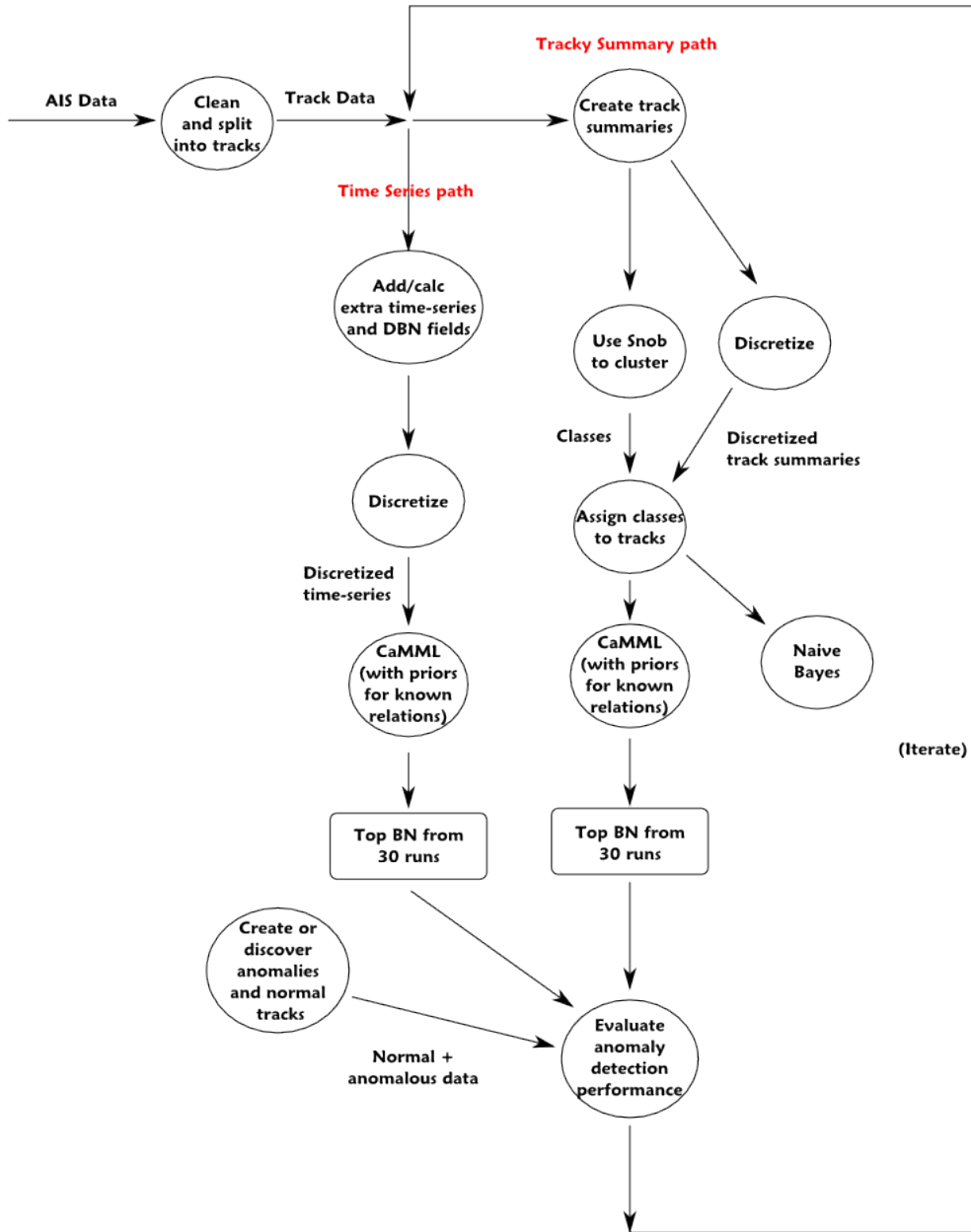


Figure 1: Workflow for the experiments

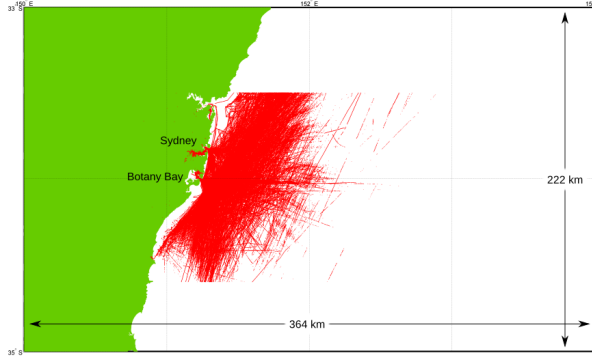


Figure 2: The area covered by the AIS track data

MMSI	Timestamp	Latitude	Longitude	Speed	Course	Heading	NavStatus
xxxxxxxx	200905xxxxxxxx	-33.xxxxxxx	151.xxxxxxx	18.7	49.9	46	0
xxxxxxxx	200905xxxxxxxx	-34.xxxxxxx	151.xxxxxxx	2.1	218	80	0
xxxxxxxx	200905xxxxxxxx	-33.xxxxxxx	151.xxxxxxx	0	0	511	0
xxxxxxxx	200905xxxxxxxx	-34.xxxxxxx	151.xxxxxxx	17.5	183	179	0
xxxxxxxx	200905xxxxxxxx	-33.xxxxxxx	151.xxxxxxx	1.2	28	64	9

Figure 3: An example of five consecutive rows from the original AIS data, with information removed to preserve anonymity. Each row has been produced by a different ship

Track summaries The second approach is to summarise each track as a whole — for example, identifying the number of times the vessel stops, the main stopping locations in the track and so forth.⁵ While track summaries cannot be used as easily in online situations, they can capture track patterns that occur at the time scale of the track as a whole. As an example, if a vessel heads out to sea in a straight line, turns around at a constant rate, then returns directly home, each timestep in the track may appear perfectly normal to any time series-based normalcy model. However, the behaviour embodied by the track as a whole may be quite anomalous and worthy of attention.

3.2.1 Variables

Most of the anomaly detection models that we have encountered thus far tend to be limited to kinematic variables, such as location, speed and course, coupled with the type of the vessel. We feel that there are advantages in expanding the number and type of variables used to model vessel behaviour, particularly in the context of BNs in which the resultant model, while more complex, will still be quite easy to understand for non-specialists.

Thus, for the present experiments, we have added variables related to the ship itself, the weather, natural temporal factors, kinematic DBN nodes and elementary information on vessel interactions for both the time series and track summary models. Information about each ship has been sourced from three locations: the public websites marinetraffic.com and digital-seas.com and also from the DSTO. Coverage is generally excellent; for example, only 13 of the 544 vessels lack ship type information. On the few occasions in which data is missing, we include a missing or invalid state. Weather information for the period was retrieved from the Australian Bureau of Meteorology website and is based on observation stations around Sydney harbour (Bureau of Meteo-

⁵Of course, these stops are the “brief” ones of less than 6 hours, otherwise a track split would have occurred.

rology, 2010). Temporal variables, such as time since dawn, have been calculated based on sunrise and sunset tables for the period. We have also added DBN nodes for all the original variables

The variables for each type of model are described in Appendix A.

3.3 Classification and discretisation

In addition to the vessel type for the track summaries, we also classify the tracks using a tool developed at Monash University called Snob (Wallace and Freeman, 1992). Snob is an unsupervised classification and factor analysis tool, and is comparable to Auto-Class (Cheeseman et al., 1988). It uses Minimum Message Length (MML; described shortly and in Appendix D) to identify the most suitable set of Gaussian models for classifying a set of continuous and multinomial data (though it can also work with Poisson and von Mises models). Snob can assign data to classes either partially (i.e. by identifying a set of classes to which the data might belong, along with a probability of membership) or to the single most probable class. We use Snob here to perform unsupervised classification on the track summaries, which yields a class for each track. The track class is represented by the ‘Class’ variable in the track summary network, which can be seen in Figure 5 under the head ‘Vessel Type’.⁶

Not only do we use Snob to classify the track summaries, we also use it to perform discretisation. Discretisation of the variables in the data set is needed for technical reasons — the version of CaMML that we use in this project works only with discrete data along with the ultimately discrete representations used by the Netica programming interface.⁷ To perform discretisation, we take the set of values for a single variable, and classify this one dimensional data. Each discovered class becomes a state of the variable, and each state is represented by the mid-point value of the class. Using Snob in this way allows us to recover any hidden regularities that underlie the variable data and is similar to the attribute clustering approach taken by Li et al. (2006). This can often lead to nodes with uneven distributions. For example, the ‘Speed’ node in Figure 4 contains lower probability states wedged in amongst higher probability states. One might expect to see a more even distribution, however Snob has identified 12 underlying Gaussians corresponding to these 12 states — some of which happen to occur much more frequently than their neighbours.

Sometimes, Snob is unable to find more than one or two classes — in those cases, we fall back to using an equal frequency discretisation to yield three states for the variable.⁸

3.4 The CaMML machine learner

In this work, we make use of the CaMML machine learning tool (Korb and Nicholson, 2004). CaMML (which stands for Causal discovery via MML) was developed at Monash University and aims to recover causal BNs from training data using a search and score approach. CaMML uses simulated annealing followed by a Markov chain Monte Carlo (MCMC) for search and MML (Minimum Message Length; Wallace and Boulton, 1968) for scoring how well candidate networks describe the training data while penalising networks that overfit. MML is an information theoretic measure, related to maximum

⁶After classifying the data with Snob, we trained a Naive Bayes network with the Snob class as the classification variable to see how well such a network could perform. This produced a network capable of classifying the data with a very high level of accuracy (of around 85%), suggesting that even a simple network such as this may be of some use in assessing anomalies.

⁷A continuous version of CaMML exists, but was unsuited to this project for practical reasons.

⁸A better, though slower, approach to discretisation would be to maximise the effect that a finer grained discretisation of one variable has on the remaining variables, while trading off on the cost of the degree of discretisation.

1st Tier	ShipType, ShipSize, Rainfall, MaxTemp, EstWindSpeed, EstOk-tas
-----------------	--

2nd Tier	Lat, Lon, Speed, Course, Heading, Acceleration, DayOfWeek, HourOfDay, CourseChangeRate, HeadingChangeRate, NumCloseInteractions, NumLocalInteractions, ClosestType, ClosestSpeed, ClosestCourse, ClosestDistance, SinceDawn, SinceDusk
-----------------	--

3rd Tier	Lat-t2, Lon-t2, Course-t2, Heading-t2, Speed-t2, Acceleration-t2
-----------------	--

Table 1: Causal tiers for the variables in the time series model, given as hard priors to CaMML

likelihood and similar (though prior) to MDL (Minimum Description Length; Rissanen, 1978).⁹

A particularly useful feature in CaMML is the ability to specify expert priors for the structure of the learnt network. These can be in the form of hard priors (e.g. an arc *must* be either present or absent) or soft priors that specify the probability of certain arcs (or more generally relationships) that can hold between the variables of the learnt network. In this work, we use some simple hard priors in the time series model to guarantee that the right DBN relationships hold between the kinematic variables in one time step and the next — thus, for example, we indicate that a directed arc exists that connects **Lat** to **Lat-t2** with probability 1. We also break the variables into tiers so that variables that cannot cause other variables also cannot be parents to those variables in the learnt network. Thus, for instance, the speed of a ship cannot affect its size.¹⁰ The tiers are shown in Table 1.

4 Generated models & discussion

After pre-processing the source data, dividing the workflow into both time series and track summary modelling paths, and adding in the extra variables and calculations described above, we perform training. We randomly set aside 80% (or 1,978 tracks) for this purpose and 20% for testing. We then feed the training data, both time series and track summaries, into CaMML 10 times, each run having a different random seed. Figures 4 and 5 show examples of the networks that CaMML produces in each case.

The first thing one can observe in the track summary model is that information about the weather (cloud cover, temperature and so forth) has no effect on the remainder of the network (see the isolated subnetwork on the left in Figure 5). While we expected the influence of weather to be weak in the track summary data — since much of the ship’s behaviour at the track-level is unlikely to be influenced by anything but extreme weather — we were surprised that it exerted no influence in any of the networks learnt. We believe that the period of time studied (three consecutive months, from late autumn to mid-winter) was too little and the distance of the weather station from each vessel was too great and that further data may still turn up a connection. By contrast, training for the time series model *has* incorporated weather into the network, although the strength of the influence, at least to kinematic variables, is relatively weak.

Beyond this, it is also clear that very few arcs in the generated networks represent

⁹For a more detailed description of both MML and CaMML, see Appendices D and E.

¹⁰If we ignore relativity.

intuitive directed causal relations. That is largely because there are no such relations to be found — the obvious exceptions being the DBN arcs, which we setup in advance as hard priors, and potentially the weather variables, which were found to have only a weak influence given the data available. Many of the remaining variables are simultaneous properties of the vessel and its motion, which will indeed be correlated but only via hidden common ancestors. Thus, while a ship’s speed, size and course will be correlated, it would be a stretch to argue that any one is a direct (or ancestral) cause of any other — that a ship’s size, for example, could cause its speed or course.¹¹ More likely, these variables correlate due to common causes, such as the aims of the vessel’s owner, geography, resource constraints or earlier states of the system (i.e. DBN arcs).

Interacting with the models turns up many points of interest; there are too many to describe here, but it is worth describing a selection of them. In the time series model, for example, we see that entering ‘Tug’ or ‘Pilot Vessel’ into the ‘ShipType’ variable significantly increases the chance of a vessel being located nearby — i.e. that the ‘ClosestType’ variable will have a state other than ‘None’ and that the ‘NumCloseInteractions’ and other related variables will have non-zero values. By contrast, cargo ships travel mostly solo while tankers travel almost exclusively alone. Ship sizes (i.e. the ‘ShipSize’ variable) are also quite well correlated with position (the ‘Lat’ and ‘Lon’ variables) via the ‘ShipType’ variable, with larger vessels, such as ‘Tankers’ and ‘Cargo’ tending to appear in a restricted set of locations. Thus setting ‘ShipSize’ to 0.827 causes the latitude and longitude variables ‘Lat’ and ‘Lon’ to shift the weight of probability on to -33.948 and 151.458, respectively. Latitudes also tend to correlate with certain longitudes, as expected when vessels appear in common locations and travel common paths.

In the track summary network, the Class variable (produced by Snob) has a significant influence on other nodes in the network, with CaMML identifying it as a direct cause of 12 other nodes. This is as expected, especially given how well the Naive Bayes model was able to utilise the information in the Snob classes. The model also shows that cargo ships and tankers spend most of their time travelling straight (i.e. setting ‘Cargo’ or ‘Tanker’ in the variable ‘shipType’ causes the probability distribution associated with ‘straightPc’ to shift to higher percentages of time spent travelling straight), while tugs are much less likely to travel straight. Tugs also tend to stop in different locations to cargo ships (the distributions of ‘mainStopLat’ and ‘mainStopLon’ are very different when ‘Tug’ rather than ‘Cargo’ is entered into the ‘shipType’ variable) and tugs tend to be stopped for a larger proportion of the time as compared to cargo ships (i.e. the probability distribution for ‘stopPc’ is weighted towards higher values for tugs).

Summaries of the 10 generated models for both approaches are shown in the arc frequency matrices of Figures 6 and 7. All the possible nodes in the network run down the left-hand side, and again across the top. The matrices show how often in the 10 models a directed arc is present between each node from the left and each node from the top. The frequency is shown as a proportion and is reflected in the shading of the cell — 1 (or bright green) indicates every network contains the arc, 0.5 (or pale green) indicates that half (in this case 5) of the networks contain it and 0 (white) indicates no networks contain it. There are two items to note about these arc frequency matrices: they are relatively sparse (indeed most cells are blank for the track summary model) and they show that the models do not contain much variation — arcs are often found in at least 70% of the networks or found in only 20%. There are nonetheless a few cases in which CaMML is less certain about the arcs, particularly in the time series model — the variables Lat and Lon in the time series model, for example, have an arc running between them as often in one direction as the other.

¹¹Note that, while ship size will be temporally prior to speed and course, that does not make it a direct or ancestral cause. In the absence of evidence to the contrary, however, that is often the best working assumption — and what CaMML has indeed suggested in the time series network.

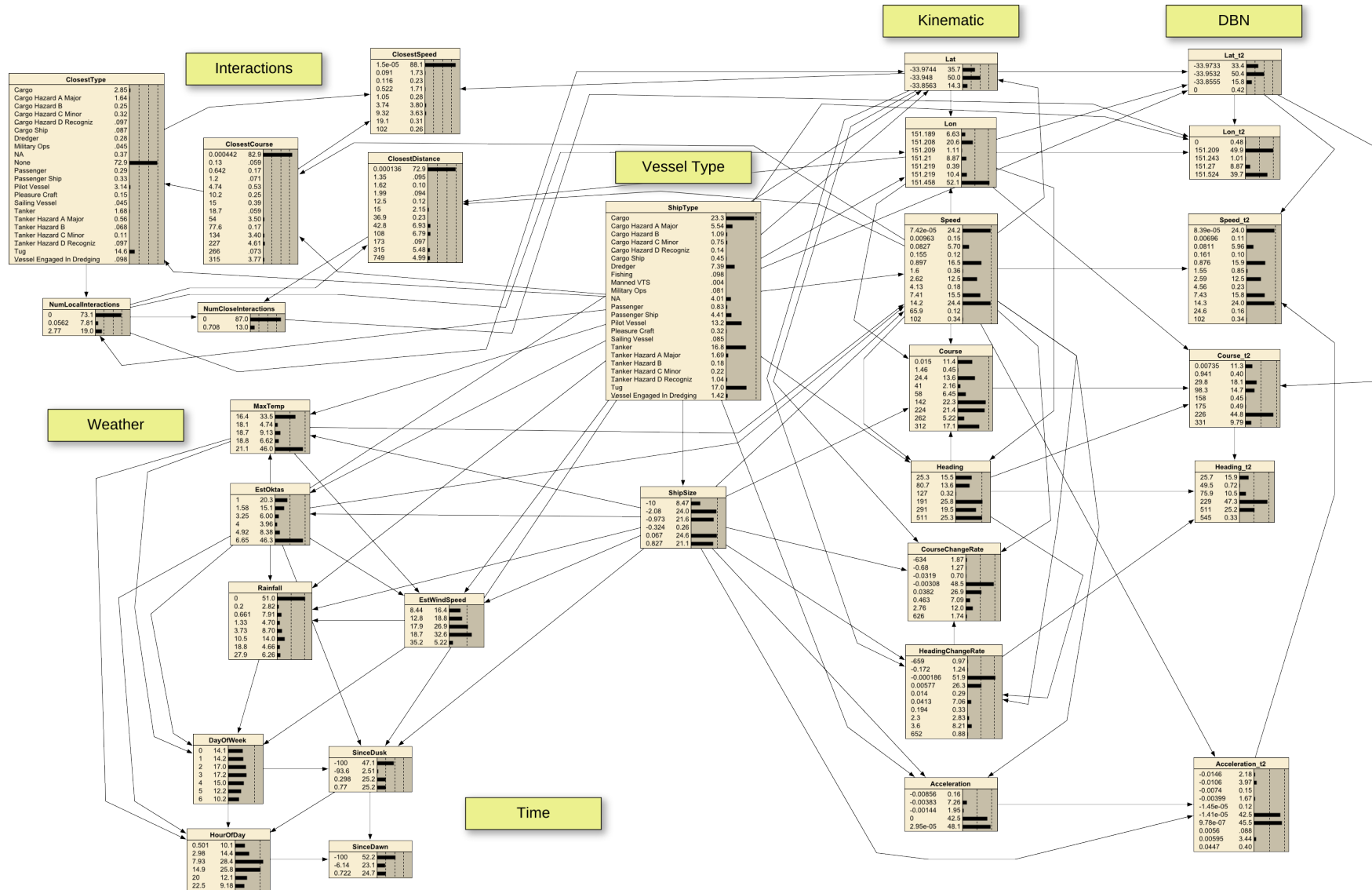


Figure 4: An example BN produced by CaMML for the time series data

Figure 5: An example BN produced by CaMML for the track summary data

The last 5 columns and the last 5 rows of Figure 7 also clearly shows the isolation of the weather variables in the track summary model, with CaMML being certain about the connections between all the weather variables (bottom right corner) and certain about the lack of connection to other variables.

4.1 Assessing network performance

The traditional method for assessing anomaly detection performance is to create a test set containing normal and anomalous tracks and seeing how many of the anomalous tracks can be identified and how many are misidentified. We pursue this approach below, but we can also gain substantial insight into how the models behave in the face of anomalous data by altering our AIS derived normal tracks (or what we presume to be normal tracks) and examining the change in the models’ conclusions. This will involve looking directly at the probabilities assigned to tracks by the model, rather than examining only the binary classification results of normal or anomalous.

To assess the probability of a track using the track summary network, we first calculate a summary for the track and then enter the summary as evidence into the network.¹² Since the track summary network has been discretised using Snob classification and the attributes in the new summary are unlikely to exactly match states in the network, we enter the evidence by finding the nearest matching state. Once this is done, we find the probability of the entered evidence which directly gives us our estimate of the probability of the track. Since these probabilities can be very low (around the order of 10^{-10}) we follow the information-theoretic approach of taking the negative log (base 2) to produce what we call here an “anomaly score”. Put simply, the higher the anomaly score, the less probable the track.

In the case of the time series network, we take a similar approach, but instead feed each timestep of the track into the network to yield a probability estimate for that timestep. We then take the average probability estimate for all timesteps as the estimated probability for the track. Again, we take the negative log (base 2) to give an anomaly score.

If we do this for all the tracks in our data set and plot the distribution of the results (obtained using a Gaussian KDE), we get the graphs shown in Figures 8a and 8b. The graphs show that there is a fair amount of diversity amongst the tracks’ anomaly scores — they do not simply clump around the lowest possible anomaly score. It should also be noted that the scores produced by the time series model are not directly comparable to those of the track summary model; amongst other reasons, it is likely that the track summary scores will be higher given the substantially larger number of nodes in that network (making each set of evidence more specific and thus less individually likely). There is a surprisingly small correlation between the two sets of scores (0.159; though statistically significant with $p < 0.001$). Earlier iterations with much cruder discretisation and more variables in common did show a stronger correlation — however, as each model grew more detailed, the correlation grew weaker. The two models look at different aspects of each track (for example, the time series has no notion of stopping points or track duration, while the track summary has no understanding of DBN elements) and as we will see later, they reinforce each other when performing anomaly detection.

In the event these techniques are used to assist surveillance system operators, the graphs suggest an interesting possibility: that live anomaly scores for vessels could be plotted relative to such a graph on the equivalent of an anomaly score radar. Thus, as particular vessels’ scores become higher relative to that of all others, the operator will

¹²Since the models do not exhibit much structural variation, and little parameter variation, and given the slow speed of calculating track probabilities, we settled on using just one of the models for each case in these experiments. Ideally, we would take an average of the 10 learnt models.

		EstOktas	Rainfall	MaxTemp	EstWindSpeed	ShipType	ShipSize	Course	HourOfDay	DayOfWeek	ClosestType	Lat	Heading	Lon	CourseChangeRate	Lat_t2	Speed	Acceleration	HeadingChangeRate	NumCloseInteractions	Lon_t2	Acceleration_t2	NumLocalInteractions	ClosestCourse	Course_t2	SinceDusk	SinceDawn	ClosestDistance	Heading_t2	Speed_t2	ClosestSpeed
EstOktas		0.30	0.90	0.90	0.40	0.30	0.30	0.90	1.00	0.10	0.20	0.20	0.10	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.20	0.10	0.10	0.00	0.00	0.00
Rainfall	0.70		0.90	0.90	0.40	0.30	0.00	0.10	1.00	0.00	0.00	0.50	0.00	0.10	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	
MaxTemp	0.10	0.00		1.00	0.40	0.30	0.10	0.40	1.00	0.30	0.20	0.10	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.10	0.00	0.00	0.30	0.00	0.00	0.10	0.20	0.20	0.00	0.00	0.00
EstWindSpeed	0.10	0.20	0.00		0.40	0.30	0.40	0.20	1.00	0.10	0.20	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.30	0.00	0.00	0.10	0.30	0.10	0.00	0.00	0.00
ShipType	0.60	0.60	0.60	0.50		1.00	0.80	0.20	0.00	0.30	1.00	0.80	0.90	0.40	0.60	0.50	1.00	0.60	0.30	0.90	0.40	0.40	0.20	0.00	0.20	0.10	0.20	0.00	0.00	0.10	
ShipSize	0.50	0.60	0.60	0.40	0.00		1.00	0.20	0.00	0.70	0.40	0.60	0.20	0.70	0.40	0.70	0.70	0.60	0.30	0.10	0.50	0.50	0.00	0.00	0.30	0.20	0.20	0.00	0.30	0.10	
Course	0.00	0.00	0.00	0.00	0.00	0.00		0.20	0.00	0.00	0.00	0.40	0.20	0.30	0.00	0.60	0.10	0.00	0.10	0.00	0.00	0.00	0.30	1.00	0.00	0.10	0.00	0.00	0.00	0.00	
HourOfDay	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70	0.50	0.00	0.00	0.00	0.00	
DayOfWeek	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.60		0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.20	0.20	0.00	0.00	0.00	0.00	
ClosestType	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.10	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.40	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.10	0.00	0.00	0.30	
Lat	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.30		0.20	0.50	0.60	1.00	0.30	0.10	0.50	0.00	0.20	0.00	0.40	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.20	
Heading	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.20	0.00	0.00	0.30		0.60	0.40	0.30	0.70	0.20	0.90	0.10	0.00	0.20	0.00	0.80	0.10	0.20	0.60	1.00	0.00	0.00		
Lon	0.00	0.00	0.00	0.00	0.00	0.00	0.40	0.00	0.30	0.50	0.20		0.20	0.90	0.40	0.10	0.10	0.30	1.00	0.20	0.40	0.20	0.80	0.00	0.10	0.70	0.00	0.10	0.40		
CourseChangeRate	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.20	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Lat_t2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.00	0.60	0.00	0.00	0.30	0.50	0.00		
Speed	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.50	0.20	0.30	0.50	0.10		0.70	0.30	0.00	0.10	0.60	0.00	0.10	0.00	0.00	0.10	0.00	1.00	0.50		
Acceleration	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.10	0.00	0.00	0.30		0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	
HeadingChangeRate	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.60	0.00	0.10	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	
NumCloseInteractions	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.10	0.00	0.10	0.10	0.00	0.00	0.20	0.00	0.00		0.50	0.00	0.30	0.10	0.00	0.00	0.50	0.00	0.00	0.60	0.00	
Lon_t2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.00	
Acceleration_t2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00	
NumLocalInteractions	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.30	0.00	0.20	0.00	0.00	0.10	0.00	0.00	0.50	0.40	0.00		0.20	0.00	0.00	0.30	0.70	0.00	0.00	0.20	
ClosestCourse	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.10		0.00	0.00	0.00	0.00	0.00	0.00	0.40	
Course_t2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.00	
SinceDusk	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.70	0.00	0.00	0.00	0.00	0.00	
SinceDawn	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.30		0.00	0.00	0.00	0.00	0.00	
ClosestDistance	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.40	0.10	0.00	0.30	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.20	0.20	0.00	0.00	0.00		0.00	0.00	0.10	0.00	
Heading_t2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.20	0.00	0.00	0.00		0.00	0.00	0.00	
Speed_t2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.30	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	
ClosestSpeed	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

Figure 6: A summary arc matrix of the 10 BNs generated for the time series data. Each cell represents an arc (row to column) and displays the proportion of networks containing that arc

[illegible]

Figure 7: A summary arc matrix of the 10 BNs generated for the time series data. Each cell represents an arc (row to column) and displays the proportion of networks containing that arc

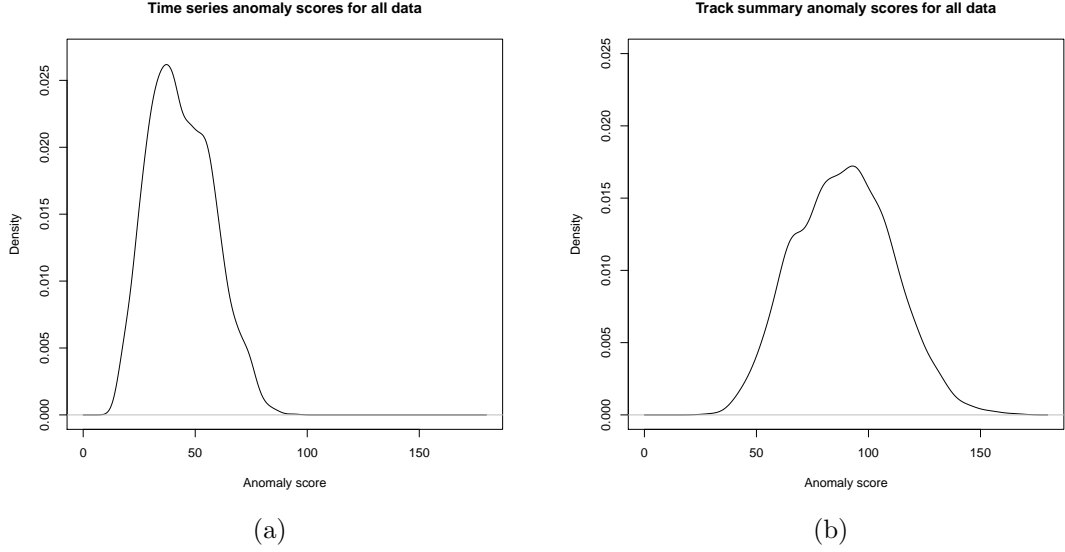


Figure 8: The KDE distributions of anomaly scores for all tracks in the data set according to the (a) time series and (b) track summary networks

know to pay it more attention.

4.1.1 Creating anomalous data

There are many ways to create anomalous data; here, we take three approaches. In the first case, we take each track from our training set and switch the ship information with that of another randomly selected ship of a different type. In the second case, we splice together different tracks — in some cases, these are tracks generated by ships of the same type, and in others, they are generated by ships of different types. In the third case, we manually create tracks by drawing them on to a map. This last approach provides a great deal of flexibility in what kind of anomalous tracks we can create. By contrast, the first two approaches allow us to compare *directly* to the anomaly scores of the original tracks.

4.1.2 The false ship effect

Figures 9a and 9b show how changing the ship to a random alternative (while leaving the remaining track data as is) affects the anomaly score — we call this the false ship effect. We can see that in most cases, the false ship effect is positive (i.e. the anomaly score increases), which is in line with what we would expect. In this narrow sense, the time series model performs better than the track summary model: the false ship effect is positive in around 87.2% of the cases as opposed to the 69.4% of cases for the track summary model. Combining the two networks' scores performs very slightly better (87.5% showing a positive false ship effect).

It does seem odd that a track would become more probable given incorrect ship information under either model. It should first be noted that many of the ship types are in fact quite similar. There are several sub-categories of cargo ships and tankers, and switching between these may not produce a noteworthy false ship effect. However, this does not account for all the cases. A closer look at these tracks shows, in fact, that many of them are highly anomalous. There is a good chance that either they have been

mislabelled or that they do indeed behave anomalously according to their type.

Figure 9c and 9d show scatter plots of the anomaly score versus the false ship effect. With the time series model, we can see that as the anomaly score grows, the false ship effect falls (with a correlation of -0.70, $p \ll 0.01$). This also occurs with the track summary model, but to a much less extent (with a correlation of -0.31, $p \ll 0.01$).

This suggests another approach to anomaly detection based on a form of *abduction*. Abduction (due to Peirce, 1955) is the process of positing an *explanation* for some observation. In a sense, abduction is the reverse of deduction. Whereas in deduction we start from a given set of premises and infer conclusions (i.e. moving from the set P to C), with abduction we start with a given conclusion (that is, some accepted fact or observation) and propose some or all of the premises that might lead to that conclusion as potential explanations of the conclusion (i.e. moving from C to some or all of the set P). While deduction always leads to certain conclusions given certain premises, abduction frequently leads to many possible explanations (sets of premises) that can be ranked probabilistically prior perhaps to gathering new evidence in priority order (i.e., prior to performing inductive statistical inferences over the set of alternative explanations).¹³

This probabilistic ranking of explanations maps very naturally to the world of Bayesian networks. If A causes B , and we know that $B = s_i$, one possible way to abduce a cause of B is by looking at which of the states of A give the highest probability to $B = s_i$. Applying this to the present case, if a track is assigned a certain probability given the normalcy model, but would have a sufficiently higher probability given a different ship type, this may be reasonable grounds for flagging the vessel as possibly behaving anomalously. Indeed, it appears that in the present case, we may have found something of a natural threshold for identifying potential anomalies (i.e. when the false ship effect is less than 0) — we believe that such a possibility warrants a more detailed investigation.

4.1.3 Track splices

We also created anomalous tracks by splicing random tracks together. Specifically, we selected 140 tracks at random and replaced their tails with those of other tracks. We spliced half of the tracks with those created by ships of a different type and we spliced the other half with tracks created by ships of the same type. When assessing these tracks using the track summary model, tracks forged from different types yield an average anomaly score of 121.3 while those forged from the same type yield an anomaly score of 115.4 (this difference being statistically significant with $p \ll 0.01$). Both scores are significantly different to the average anomaly score for all data of 89.0.

With the spliced tracks, we would expect the track summary model to perform better than the time series approach. This is because the time series model will not be able to detect unusual behaviour across the track as a whole. Indeed, the time series model does appear to perform less well. Tracks put together from ships of different types produce an average anomaly score of 48.9 while those put together from the same types produced a score of 45.6. This difference is not statistically significant ($p > 0.01$). In addition, while the higher score was significantly different ($p < 0.01$) to the average of the full data set (43.8), the lower score was not ($p \gg 0.01$). Here we can see the advantage of the higher level view embodied by the track summary approach.

4.1.4 Manually drawn anomalies

Here we test the models by passing in a set of presumed normal tracks along with a manually created set of anomalous tracks. The normal tracks are simply the 20% of our

¹³Thus, abduction per se is arguably not *inferential*, as no conclusion is drawn, but instead heuristic in guiding subsequent inductive inferences.

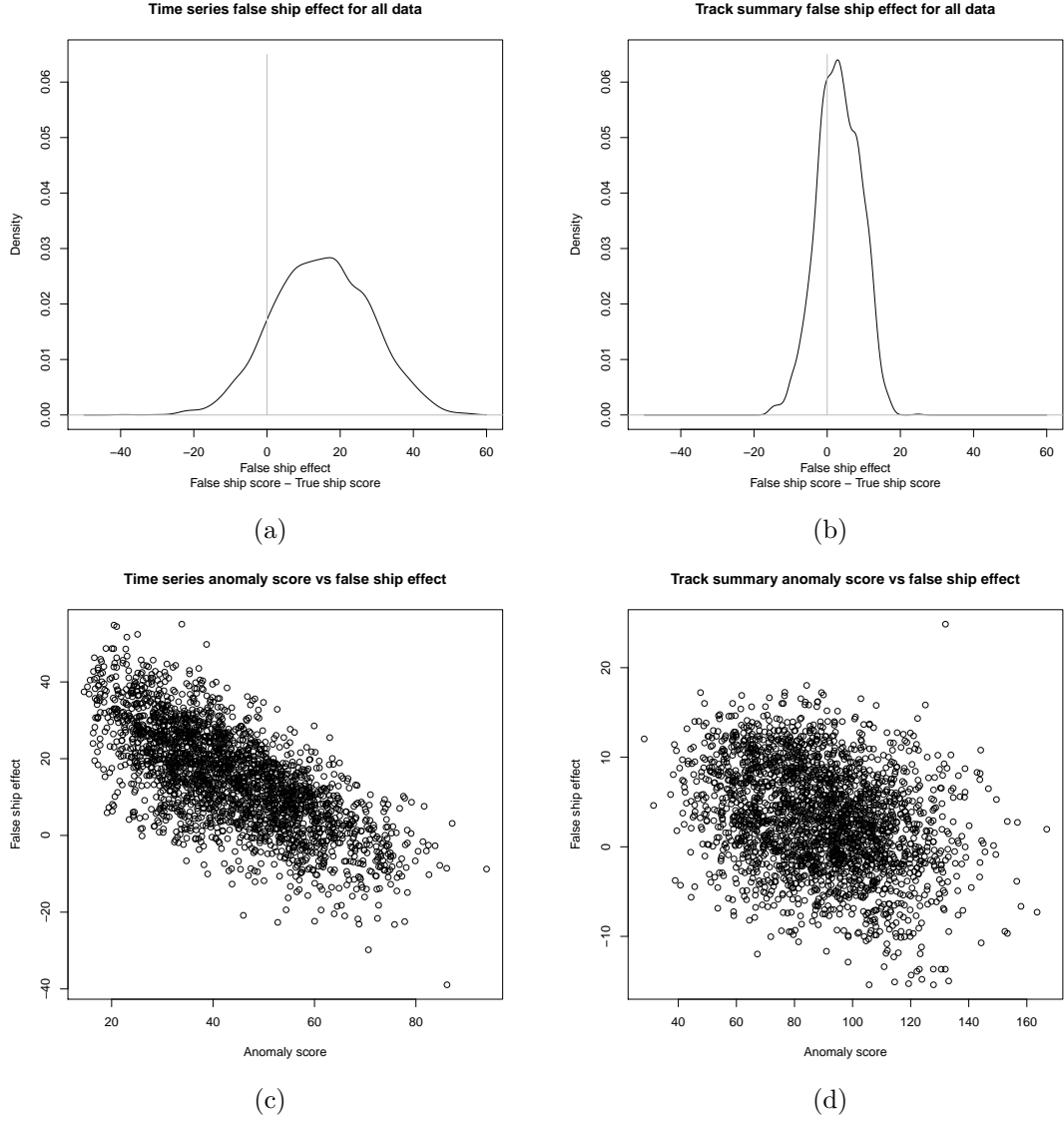


Figure 9: The difference between the anomaly scores for tracks when they contain false ship information versus correct (original) ship information (the false ship effect) for the (a) time series and (b) track summary networks, sorted by score; and a scatter plot of each track's score in the (a) time series and (b) track summary networks vs their respective false ship effects

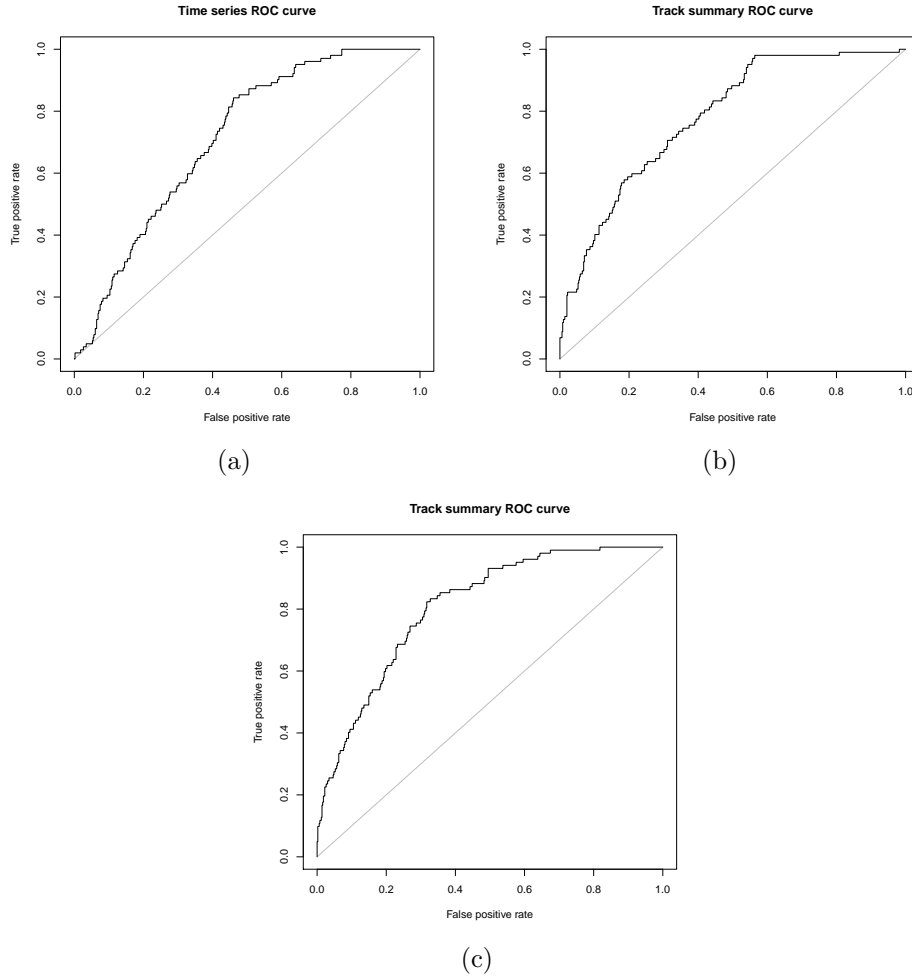


Figure 10: ROC curves for test data, containing both normal tracks and manually created anomalous tracks, given the (a) time series, (b) track summary and (c) combined models

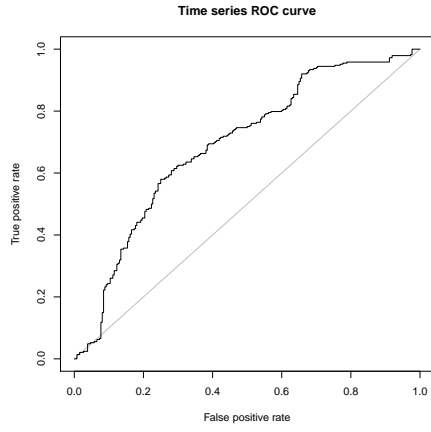
Type	Track Summary		Time Series	
	Score	Delta	Score	Delta
Normal test tracks	90.8	(0)	45.7	(0)
Random movement in the middle of water	102.4	+11.7	50.8	+5.1
Closed tracks in the middle of water	101.7	+10.9	53.7	+8.0
Very short tracks	95.5	+4.7	62.7	+17.0
Unusual stops	119.1	+28.3	48.6	+2.9
Tracks with many interactions	139.9	+49.1	75.8	+30.1
Tracks with many loops	126.2	+35.4	52.7	+7.0
Travel over land	122.2	+31.4	60.2	+14.5
Appearing at edges of observable area only	103.5	+12.7	54.2	+8.6
Very noisy observations	135.2	+44.4	54.6	+8.9
Tracks behaving against type	113.7	+22.9	57.8	+12.0
Multiple anomalies (e.g. combination of noise, no landfall, loops, etc.)	126.9	+36.1	53.9	+8.2

Table 2: Average anomaly scores for various forms of anomaly. Columns headed ‘Delta’ indicate the difference to the average score for normal test tracks

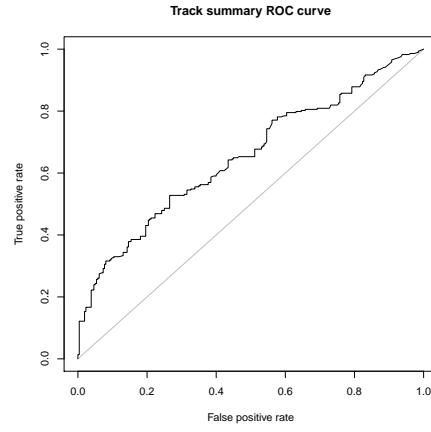
data that we set aside earlier and that were not involved in training. The anomalous tracks were “drawn” by hand. Thus, by drawing a track with the mouse over a map of the NSW coast, mouse location correlated with vessel location, and the speed of mouse movement correlated with vessel speed. Other factors were generated randomly, including the time and duration of the track, noise in the data, vessel details and maximum speed.

Anomalous behaviour in these tracks included very noisy data, close interactions with many other vessels, vessels that circle in unusual patterns or that make no landfall, vessels travelling over land, overly short tracks in the middle of the sea and vessels behaving against their type. In all, 107 tracks were created.

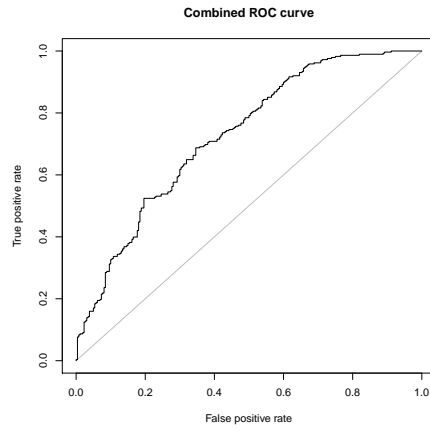
When combined with the normal track test data, and scored using the two models both independently and combined, the ROC (receiver operating characteristic) curves of Figure 10 are the result. We can see here that the track summary model (with an AUC — area under the curve — of 0.780) performs better than the time series model (with an AUC of 0.712). Combining the scores from the two models performs better again (with an AUC of 0.809). Table 2 shows the average scores each model yielded for the various kinds of anomalous tracks created. We can see that both models easily detected the tracks containing too many close interactions (averaging scores of 139.9 and 75.8, against the test averages of 90.8 and 45.7, giving a difference of +49.1 and +30.1 for track summary and time series models, respectively); the time series model detected overly short tracks best (track summary: +4.7; time series: +17) while the track summary model substantially outperformed the time series model for tracks containing unusual stops, as would be expected (track summary: +28.3; time series: +2.9). In most cases, the track summary model outperformed the time series model.



(a)



(b)



(c)

Figure 11: ROC curves for the Johansson and Falkman data using the (a) time series, (b) track summary and (c) combined models

4.1.5 Testing on other data sets

We also applied our methods to the simulated data used by Johansson and Falkman (2007), both normal and anomalous. While we will not describe the full details of those investigations here, we would like to note that the models performed reasonably well. In particular, with the track summary model, anomalous tracks received an average anomaly score of 22, while normal tracks averaged 17; while in the time series model, anomalous tracks received an average score of 29, with normal tracks averaging 25. In addition, an examination of the scores for normal tracks would suggest setting an anomaly score threshold of about 30 for both the track summary and time series cases. In the track summary case, this would result in 4% of the normal tracks being flagged as anomalous, while 17% of the anomalous tracks would have been flagged as such. In the time series case, this increases to 17% of the normal tracks being flagged anomalous, and 43% of the anomalous tracks. The full ROC curves can be seen in Figure 11; as can be seen, the time series model performs better in this case with an AUC of 0.691, over the track summary AUC of 0.652. The combined model again performs better than both individually with an AUC of 0.727. (It should also be noted that the tracks flagged as anomalous amongst the notionally normal tracks do indeed appear anomalous.)

We also examined what happens when the ship type of the tracks is altered. Interestingly, the only cases in which this change created a notable *negative* false ship effect (i.e. increased the probability of the track) again involved high anomaly scores. These scores were 25 and above for the track summaries and 36 and above for the time series — both higher than the respective average scores for the anomalous tracks.

5 Conclusion

We have seen in the above how promising a contribution Bayesian Networks can make to the cause of detecting anomalies. By using the machine learner CaMML on AIS data, combined with additional real world data, we produced networks at two different time scales, in the form of the time series and track summary models. We found that adding further real world attributes helped to create a better model of normalcy, but weather variables, at least in our data set, had either no or only a weak effect. By contrast, information on vessel interactions and vessel details had a substantial influence on network behaviour.

Neither level of abstraction performed better in all cases. In some cases — particularly when false ship data was used — the time series model better detected anomalies, while in others, such as when vessels changed behaviour mid-track or when high level behaviour appeared anomalous, the track summary approach proved more suitable. We were able to improve the performance of either model alone by combining their assessments. This suggests that additional networks for time scales other than that of the track and that of the moment may improve anomaly detection still further.

We feel that there is still a great deal of room to improve the normalcy model and the approach to anomaly detection, not just through networks at additional time scales, but also in terms of attribute selection and discretisation, identifying track starting and ending points, and investigating variations on approaches to calculating track probabilities. We would additionally like to evaluate our models using the notion of Bayesian Information Reward (Hope and Korb, 2002), which will also aid in model optimisation.

The investigation of the false ship effect suggests that abduction may also improve anomaly detection. A simple example may serve: if a vessel claims to have a particular weight or we assume a weight typical for that vessel, we can see if alternative weights provide a better explanation. If our model decrees the track more probable given more heft, the vessel may be harbouring cargo it should not.

References

- Bureau of Meteorology (2010). Daily weather observations, May – July 2009. <http://www.bom.gov.au/climate/dwo/IDCJDW2124.latest.shtml>.
- Cheeseman, P., J. Stutz, M. Self, J. Kelly, W. Taylor, and D. Freeman (1988, August). Bayesian classification. In *Proceedings of the 7th National Conference of Artificial Intelligence (AAAI-88)*, pp. 607–611.
- Cooper, G. and E. Herskovits (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learning* 9, 309–347.
- Heckerman, D. and D. Geiger (1995). Learning Bayesian networks: a unification for discrete and Gaussian domains. In P. Besnard and Hanks (Eds.), *Proceedings of the eleventh conference on uncertainty in artificial intelligence*, San Francisco, pp. 274–84. Morgan Kaufman.
- Helldin, T. and M. Riveiro (2009). Explanation methods for Bayesian networks: Review and application to a maritime scenario. In *Proceedings of the 3rd Annual Skövde Workshop on Information Fusion Topics (SWIFT 2009)*, pp. 11–16.
- Hope, L. R. and K. B. Korb (2002). Bayesian information reward. In *AI 2002: Advances in Artificial Intelligence*, Volume 2557, Berlin, pp. 272–283. Springer.
- Johansson, F. and G. Falkman (2007). Detection of vessel anomalies — a Bayesian network approach. In *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 395–400.
- Korb, K. B. and A. E. Nicholson (2004). *Bayesian Artificial Intelligence*. Chapman & Hall/CRC Press.
- Kraiman, J. B., S. L. Arouh, and M. L. Webb (2002). Automated anomaly detection processor. In *Proceedings of SPIE*, Volume 4716, pp. 128–137.
- Lam, W. and F. Bacchus (1993). Learning Bayesian belief networks: an approach based on the MDL principle. *Computational Intelligence* 10, 269–293.
- Laxhammar, R. (2008). Anomaly detection for sea surveillance. In *The 11th International Conference on Information Fusion*, pp. 55–62.
- Laxhammar, R., G. Falkman, and E. Sviestins (2009). Anomaly detection in sea traffic — a comparison of the Gaussian Mixture Model and the Kernel Density Estimator. In *Proceedings of the 12th IEEE International Conference on Information Fusion (FUSION 2009)*, pp. 756–763.
- Li, X., J. Han, and S. Kim (2006). Motion-Alert: Automatic anomaly detection in massive moving objects. In *Proceedings of the 2006 IEEE Intelligence and Security Informatics Conference (ISI 2006)*, Berlin, pp. 166–177. Springer.
- Maritime Safety Committee (1994). Goal-based standards under development at IMO’s Maritime Safety Committee. http://www.imo.org/About/mainframe.asp?topic_id=848&doc_id=4574#ais.
- Nielsen, U., J. Pellet, and A. Elisseeff (2008). Explanation trees for causal Bayesian networks. In *Proceedings of the 24th Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pp. 427–434.

- Peirce, C. (1955). Abduction and induction. In *Philosophical writings of Peirce*, pp. 150–156. New York: Dover Books.
- Rhodes, B., N. Bomberger, M. Seibert, and A. Waxman (2005). Maritime situation monitoring and awareness using learning mechanisms. In *Military Communications Conference*, pp. 646–652.
- Rhodes, B. J., N. A. Bomberger, T. M. Freyman, W. Kreamer, L. Kirschner, A. C. L’Italien, W. Mungovan, C. Stauffer, L. Stolzar, A. M. Waxman, and M. Seibert (2007). SeeCoast: Persistent surveillance and automated scene understanding for ports and coastal areas. In *Proceedings of SPIE*, Volume 6578, pp. 65781M.1–65781M.12.
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica* 14(5), 465–471.
- Spirites, P. and C. Glymour (1991). A fast algorithm for discovering sparse causal graphs. *Social Science Computer Review* 9, 62–72.
- Wallace, C. and D. Boulton (1968). An information measure for classification. *Computer Journal* 11(2), 185–194.
- Wallace, C. S. and P. R. Freeman (1992). Single factor estimation by MML. *Journal of the Royal Statistical Society B* 54(1), 195–209.

A Variables

Variables	Description
Lat, Lon	Location of the vessel
Speed, Course, Heading	As per original data
Acceleration	Calculated from the speed and time step delta
DayOfWeek	Current day of the week (Mon-Sun)
HourOfDay	Current hour of the day
CourseChangeRate, HeadingChangeRate	Rate at which course (heading) is changing, in degrees per minute
Lat-t2, Lon-t2, Course-t2, Heading-t2, Speed-t2, Acceleration-t2	The state of each corresponding variable one minute from the current time
SinceDawn, SinceDusk	The amount of time that has passed since dawn (dusk) as a fraction of the total length of daylight (night)
MinTemp, MaxTemp	The minimum and maximum day temperatures
Rainfall	The rainfall for the day
EstOktas	The estimated cloud cover for the day (taken at 9am and 3pm, and averaged)
EstWindSpeed	The estimated wind speed for the day (taken at 9am and 3pm, and averaged)
ShipType	The type of vessel (Cargo, Tanker, Tug, etc.) as given by a combination of vessel tracking websites and the DSTO. Set to NA if missing.
Flag	The vessel's country of origin. Set to NA if missing.
ShipSize	A combination of the vessel's length, width, draught and dead weight. (For each attribute, a z-score is calculated based on all the known vessels, and then the average of the z-scores is taken, ignoring missing attributes.) If no information about size is available, is set to -10.
NumCloseInteractions	Number of other vessels close by (i.e. within 100m)
NumLocalInteractions	Number of other vessels in the locality (i.e. within 1km)
ClosestType	The type of the closest vessel
ClosestSpeed, ClosestCourse	The speed and course of the closest vessel
ClosestDistance	The distance to the closest vessel

Table 3: Variables used in the time series model

Variables	Description
duration	The length of time of the track
startPointLat, startPointLon	The starting latitude and longitude of the track
endPointLon, endPointLat	The ending latitude and longitude of the track
maxLat, maxLon	The maximum latitude and longitude reached
minLat, minLon	The minimum latitude and longitude reached
speedSd	Standard deviation of the speed
courseSd	Standard deviation of the course
headingSd	Standard deviation of the heading
maxSpeed	The maximum speed reached
stops	The number of stops (defined as the number of times the vessel moves at a rate less than 1 knot after having moved at a greater speed or at the start of the track)
stopPc	The percentage of time for which the vessel is stopped (i.e. moving at a rate less than 1 knot)
mainStopLat, mainStopLon	(If the vessel has stopped at least once) The location at which the vessel has been stopped for the longest time
main2StopLat, main2StopLon	(If the vessel has stopped at least twice) The location at which the vessel has been stopped for the second longest time
timeDeltaAvg	The average time difference between steps in the original track
speed1-5Pc (speed5-10Pc, speed10-15Pc, speed15-20Pc, speed20-Pc)	The percentage of time that the vessel has spent travelling between 1 and 5 knots (or 5 and 10 knots, etc.)
straightSections	The number of times the vessel has travelled straight (defined as the course changing by less than 1 degree per minute when not stopped)
straightPc	The percentage of time the vessel has spent travelling straight
mainCourse	(If the vessel has travelled straight at least once) The course for which the vessel has travelled straight the longest
main2Course	(If the vessel has travelled straight at least twice) The course for which the vessel has travelled straight the second longest
courseChangeRate1-5Pc (courseChangeRate5-10Pc, courseChangeRate10-20Pc, courseChangeRate20-Pc)	The percentage of time that the vessel has spent changing course at a rate of between 1 and 5 degrees per minute (or between 5 and 10 degrees per minute, etc.)
shipType	The type of vessel (Cargo, Tanker, Tug, etc.) as given by a combination of vessel tracking websites and the DSTO. Set to NA if missing.
flag	The vessel's country of origin. Set to NA if missing.

shipSize	A combination of the vessel's length, width, draught and dead weight. (For each attribute, a z-score is calculated based on all the known vessels, and then the average of the z-scores is taken, ignoring missing attributes.) If no ship size information is available, it is set to -10.
avgSpeedRecorded	The average speed recorded for the vessel by marine-traffic.com
avgMinTemp	The minimum temperature on the date of the track (or if the track runs over multiple days, the average minimum temperature)
avgMaxTemp	The maximum temperature on the date of the track (or multiple day average)
avgRainfall	The rainfall on the date of the track (or multiple day average)
avgEstOktas	The estimated cloud cover on the date of the track (or multiple day average)
avgEstWindSpeed	The estimated wind speed on the date of the track (or multiple day average)
NumCloseInteractions	The number of vessels that have appeared close to this vessel (where "close" is defined as within 100m)
NumLocalInteractions	The number of vessels that have appeared within the locality of this vessel (where "local" is defined as within 1km)
CloseInteractionsPc	The percentage of time at least one other vessel is close to this vessel
LocalInteractionsPc	The percentage of time at least one other vessel is in the locality of this vessel
AvgClosestDistance	(If a vessel has appeared in the locality of this vessel) The average distance of the vessel that has gotten closest to this vessel
LongestCloseType	(If another vessel has passed close to this vessel) The type of vessel that has stayed close (i.e. within 100m) to this vessel for the longest time
LongestCloseMyAvgSpeed	(If another vessel has passed close to this vessel) The average speed of this vessel during the period in which the closest vessel stayed close
LongestCloseMyAvgLat, LongestCloseMyAvgLon	(If another vessel has passed close to this vessel) The average latitude and longitude of this vessel during the period in which the closest vessel stayed close
Class	The Snob assigned class for this track

Table 4: Variables used in the track summary model

B Bayesian network basics

A **Bayesian network** is a graphical structure that allows us to represent and reason about an uncertain domain. The nodes in a Bayesian network represent a set of random variables, $\mathbf{V} = v_1, \dots, v_i, \dots, v_n$, from the domain. A set of directed **arcs** (or links) connects pairs of nodes, $v_i \rightarrow v_j$, representing the direct dependencies between variables. Assuming discrete variables, the strength of the relationship between variables is quantified by conditional probability distributions associated with each node. The only constraint on the arcs allowed in a BN is that there must not be any directed cycles: you cannot return to a node simply by following directed arcs. Such networks are called directed acyclic graphs, or simply **dags**.

Example: Ship size. We wish to consider the implications of ship size on vessel behaviour. We know that ship building costs will be an important factor, as well as whether the ship carries any cargo. We also recognise that vessel motion information such as speed, acceleration and turning may be part of the model.

B.1 Nodes and values

The first step to building a Bayesian network, is identifying the variables of interest. This involves answering the question: what are the nodes to represent and what values can they take, or what state can they be in?

The values should be both **mutually exclusive** and **exhaustive**, which means that the variable must take on exactly one of these values at a time. Common types of discrete nodes include:

- Boolean nodes, which represent propositions, taking the binary values true (T) and false (F). In a vessel behaviour domain, the node *Cargo* would represent the proposition that the ship carries cargo.
- Ordered values. For example, a node *Building Costs* might represent the costs involved in building a ship and take the values $\{low, medium, high\}$.
- Integral values. For example, a node called *Length* might represent a ship's length and have possible values from 1 to 400.

Below are some possible first choices of nodes and values for the ship size example.

Table 5: Node and value choices for the ship size example

Node name	Type	Values
<i>Cargo</i>	Boolean	$\{True, False\}$
<i>Building Costs</i>	Binary	$\{Low, High\}$
<i>Ship Size</i>	Binary	$\{Small, Large\}$
<i>Acceleration</i>	Binary	$\{High, Low\}$
<i>Rate of Turn</i>	Binary	$\{High, Low\}$

B.2 Structure

The structure, or topology, of the network should capture qualitative relationships between variables. In particular, two nodes should be connected directly if one affects or causes the other, with the arc indicating the direction of the effect. So, in our vessel behaviour example, we might ask what factors affect the size of a ship? If the answer is “building costs and whether the ship carries cargo,” then we should add arcs from *Cargo* and *Building Costs* to *Ship Size*. Similarly, ship size will affect the ship’s ability to accelerate and turn. So we add arcs from *Ship Size* to *Acceleration* and *Rate of Turn*. The resultant structure is shown in Figure 12. It is important to note that this is just one possible structure for the example.

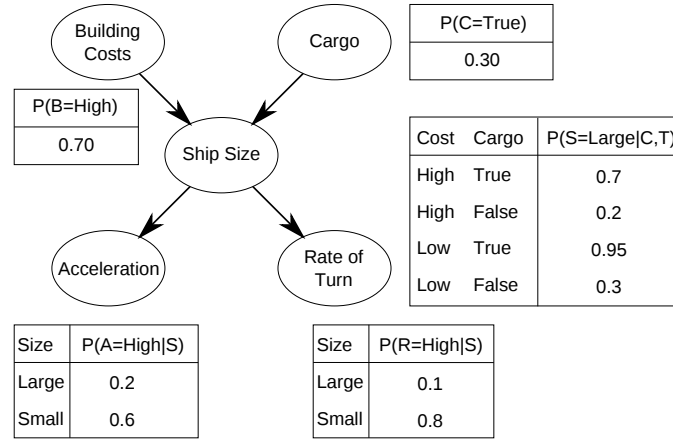


Figure 12: A BN for the ship size example

Structure terminology and layout

In talking about network structure it is useful to employ a family metaphor: a node is a **parent** of a **child**, if there is an arc from the former to the latter. Extending the metaphor, if there is a directed chain of nodes, one node is an **ancestor** of another if it appears earlier in the chain, whereas a node is a **descendant** of another node if it comes later in the chain. In our example, the *Ship Size* node has two parents, *Building Costs* and *Cargo*, while *Cargo* is an ancestor of both *Acceleration* and *Rate of Turn*. Similarly, *Rate of Turn* is a child of *Ship Size* and descendant of *Building Costs* and *Cargo*. The set of parent nodes of a node X is given by $Parents(X)$.

Another useful concept is that of the **Markov blanket** of a node, which consists of the node’s parents, its children, and its children’s parents. Other terminology commonly used comes from the “tree” analogy (even though Bayesian networks in general are graphs rather than simple trees): any node without parents is called a **root** node, while any node without children is called a **leaf** node. Any other node (non-leaf and non-root) is called an **intermediate node**. Given a causal understanding of the BN structure, this means that root nodes represent original causes, while leaf nodes represent final effects. In our example, the causes *Building Costs* and *Cargo* are root nodes, while the effects *Acceleration* and *Rate of Turn* are leaf nodes.

By convention, for easier visual examination of BN structure, networks are usually laid out so that the arcs generally point from top to bottom. This means that the BN “tree” is usually depicted upside down, with roots at the top and leaves at the bottom.

B.3 Conditional probabilities

Once the topology of the BN is specified, the next step is to quantify the relationships between connected nodes – this is done by specifying a conditional probability distribution for each node. As we are only considering discrete variables, this takes the form of a conditional probability *table* (CPT).

For each node we need to look at all the possible combinations of values of those parent nodes. Each such combination is called an **instantiation** of the parent set. For each distinct instantiation of parent node values, we need to specify the probability that the child will take each of its values.

For example, consider the *Ship Size* node of Figure 12. Its parents are *Building Costs* and *Cargo* and take the possible joint values $\{ \langle H, T \rangle, \langle H, F \rangle, \langle L, T \rangle, \langle L, F \rangle \}$. The conditional probability table specifies in order the probability of a large ship for each of these cases to be: $\langle 0.7, 0.2, 0.95, 0.3 \rangle$. Since these *are* probabilities, and must sum to one over all possible states of the *Ship Size* variable, the probability of a small ship is already implicitly given as one minus the above probabilities in each case; i.e., the probability of a small ship in the four possible parent instantiations is $\langle 0.3, 0.8, 0.05, 0.7 \rangle$.

C Dynamic Bayesian networks

Bayesian and decision networks model relationships between variables at a particular point in time or during a specific time interval. Although a causal relationship represented by an arc implies a temporal relationship, BNs do not explicitly model temporal relationships between variables. And the only way to model the relationship between the current value of a variable, and its past or future value, is by adding another variable with a different name. Generally, it is important to be able to represent and reason about changes over time explicitly when performing such tasks as monitoring, diagnosis, prediction and decision making/planning. Dynamic Bayesian networks are a generalization of Bayesian networks that explicitly model change over time.

C.1 Nodes, structure and CPTs

Suppose that the domain consists of a set of n random variables $\mathbf{V} = \{V_1, \dots, V_n\}$, each of which is represented by a node in a Bayesian network. When constructing a DBN for modelling changes over time, we include one node for each V_i for each time step. If the current time step is represented by t , the previous time step by $t - 1$, and the next time step by $t + 1$, then the corresponding DBN nodes will be:

- Current: $\{V_1^t, V_2^t, \dots, V_n^t\}$
- Previous: $\{V_1^{t-1}, V_2^{t-1}, \dots, V_n^{t-1}\}$
- Next: $\{V_1^{t+1}, V_2^{t+1}, \dots, V_n^{t+1}\}$

Each time step is called a **time-slice**. The relationships between variables in a time-slice are represented by **intra-slice** arcs, $V_i^T \rightarrow V_j^T$. Although it is not a requirement, the structure of a time-slice does not usually change over time. That is, the relationship between the variables $V_1^T, V_2^T, \dots, V_n^T$ is the same, regardless of the particular T .

The relationships between variables at successive time steps are represented by **inter-slice** arcs, also called **temporal arcs**, including relationships between (i) the same variable over time, $V_i^T \rightarrow V_i^{T+1}$, and (ii) different variables over time, $V_i^T \rightarrow V_j^{T+1}$.

In most cases, the value of a variable at one time affects its value at the next, so the $V_i^T \rightarrow V_i^{T+1}$ arcs are nearly always present. In general, the value of any node at one

time can affect the value of any other node at the next time step. Of course, a fully temporally connected network structure would lead to complexity problems, but there is usually more structure in the underlying process being modeled.

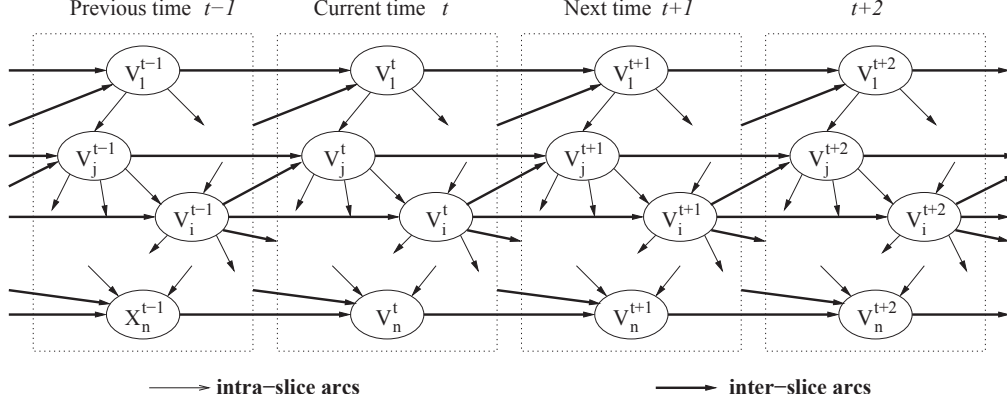


Figure 13: General structure of a Dynamic Bayesian Network

Figure 13 shows a generic DBN structure, with a sequence of the same static BNs connected with inter-slice arcs (shown with thicker arcs). Note that there are no arcs that span more than a single time step. This is another example of the **Markov assumption**, that the state of the world at a particular time depends only on the previous state and any action taken in it.

The relationships between variables, both intra-slice and inter-slice, are quantified by the conditional probability distribution associated with each node. In general, for node V_i^T with intra-slice parents Y_1^T, \dots, Y_m^T and inter-slice parents V_i^{T-1} and $Z_1^{T-1}, \dots, Z_r^{T-1}$, the CPT is

$$P(V_i^T | Y_1^T, \dots, Y_m^T, V_i^{T-1}, Z_1^{T-1}, \dots, Z_r^{T-1}).$$

Given the usual restriction that the networks for each time slice are exactly the same and that the changes over time also remain the same (i.e., both the structure and the CPTs are unchanging), a DBN can be specified very compactly. The specification must include:

- Node names
- Intra-slice arcs
- Temporal (inter-slice) arcs
- CPTs for the first time slice t_0 (when there are no parents from a previous time)
- CPTs for $t + 1$ slice (when parents may be from t or $t + 1$ time-slices).

Figure 14 shows how we can use a DBN to represent change over time explicitly. The ship's position (i.e. latitude and longitude) at time t will clearly be a critical factor in where the ship will be positioned at time $t + 1$ (assuming the length of time between t and $t + 1$ is small). Similarly, a ship's heading at t clearly affects its heading at $t + 1$, and the same can also be said for the course and rate of turn. The rate of turn at t will also affect the ship's heading at $t + 1$, which in turn will influence its position. There are also interactions *within* a single time slice; for example, the intended course at t will affect the heading of the vessel at t .

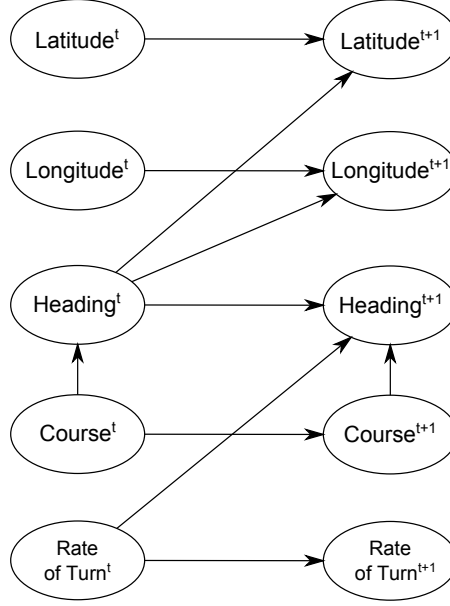


Figure 14: A DBN modelling vessel motion

D MML

The basic idea behind MML is to play a tradeoff between model simplicity and fit to the data by minimizing the length of a *joint* description of the model and the data assuming the model is correct. Thus, if a model is allowed to grow arbitrarily complex, and if it has sufficient representational power (e.g., sufficiently many parameters), then eventually it will be able to record directly all the evidence that has been gathered. In that case, the part of the message communicating the data given the model will be of length zero, but the first part communicating the model itself will be quite long. Similarly, one can communicate the simplest possible model in a very short first part, but then the equation will be balanced by the necessity of detailing *every* aspect of the data in the second part of the message, since none of that will be implied by the model itself. Minimum encoding inference seeks a golden mean between these two extremes, where any extra complexity in the optimal model is justified by savings in inferring the data from the model.

In principle, minimum encoding inference is inspired by Claude Shannon's measure of information (see Figure 15).

Shannon information measure

$$I(m) = -\log P(m)$$

Applied to joint messages of hypothesis and evidence:

$$I(h, e) = -\log P(h, e) \quad (1)$$

Shannon's concept was inspired by the hunt for an efficient code for telecommunications; his goal, that is, was to find a code which maximized use of a telecommunications channel by minimizing expected message length. If we have a coding scheme which satisfies Shannon's definition D, then we have what is called an **efficient code**. Since

efficient codes yield probability distributions (multiply by -1 and exponentiate), efficiency requires observance of the probability axioms. Indeed, in consequence we can derive the optimality of minimizing the two-part message length from Bayes' Theorem:

$$\begin{aligned} P(h, e) &= P(h)P(e|h) \\ -\log P(h, e) &= -\log[P(h)P(e|h)] \\ -\log P(h, e) &= -\log P(h) - \log P(e|h) \\ I(h, e) &= I(h) + I(e|h) \end{aligned}$$

A further consequence is that an efficient code cannot encode the same hypothesis in two different lengths, since that would imply two distinct probabilities for the very same hypothesis.

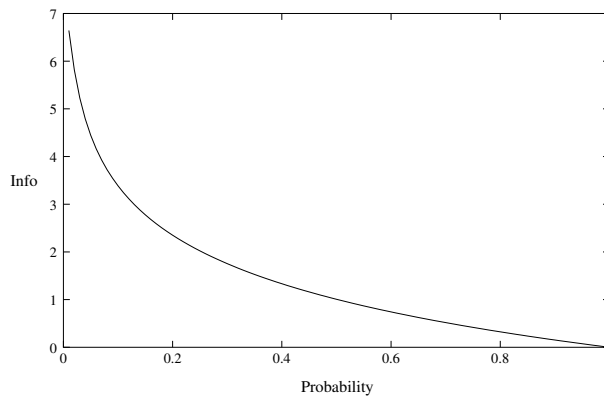


Figure 15: Shannon's information measure

Minimum encoding inference metrics thus can provide an estimate of the joint probability $P(h, e)$. Since at least one plausible goal of causal discovery is to find that hypothesis which maximizes the conditional probability $P(h|e)$, such a metric suffices, since maximizing $P(h, e)$ is equivalent to maximizing $P(h|e)$. It is worth noting that in order to compute such a metric we need to compute how long the joint message of h together with $e|h$ *would be* were we to build it. It is not actually necessary to build the message itself, so long as we can determine how long it would be without building it.

E CaMML: Causal discovery via MML

CaMML attempts to learn the best causal structure to account for an observational sample, using an MML metric and an MCMC search (using the Metropolis algorithm) (Wallace and Boulton, 1968). It does this by sampling from a posterior distribution over the causal model space, using the minimum message length (MML) probability. MML provides an information-theoretic metric which combines a model complexity penalty with a penalty for unexplained data values. Either linear path models or discrete Bayesian networks can be learned. Some versions are able to learn local structure (versus full conditional probability tables) in the form of logit models or classification trees. Alternative search algorithms include genetic algorithm search.

Other metric causal discovery algorithms include K2 (Cooper and Herskovits, 1992), the BGe and BDe metrics (Heckerman and Geiger, 1995), and MDL (minimum description length) (Lam and Bacchus, 1993). These metrics are implemented in algorithms which apparently require more prior information about the domain than CaMML. K2

requires a prior total ordering of variables; the MDL algorithm requires a useful partial order to perform well; BGe/BDe requires a prior preferred causal model.

A distinguishing feature of the MML metric from all others is that CaMML samples the space of totally-ordered models, rather than directed acyclic graphs (DAGs) or patterns (statistically equivalent DAGs). Because of this, CaMML considers, for example, a common-cause structure ($A \leftarrow B \rightarrow C$) to be twice as likely a priori as a directed chain ($A \rightarrow B \rightarrow C$), assuming there is no specific prior information to the contrary. The other metrics assume a uniform prior over either DAGs or patterns.