International Journal of Approximate Reasoning 55 (2014) 84–98

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Anomaly detection in vessel tracks using Bayesian networks

Steven Mascaro [a,*], Ann Nicholson [b], Kevin Korb [b]

[a] Bayesian Intelligence Pty Ltd., 2/21 The Parade, Clarinda, Victoria 3169, Australia
[b] Clayton School of IT, Monash University, Wellington Road, Clayton, Victoria 3800, Australia

ARTICLE INFO

Article history:
Available online 2 April 2013

Keywords:
Machine learning
Bayesian networks
Models of normality
Anomaly detection
AIS
Maritime data

ABSTRACT

In recent years electronic tracking has provided voluminous data on vessel movements, leading researchers to try various data mining techniques to find patterns and, especially, deviations from patterns, i.e., for anomaly detection. Here we describe anomaly detection with data mined Bayesian Networks, learning them from real world Automated Identification System (AIS) data, and from supplementary data, producing both dynamic and static Bayesian network models. We find that the learned networks are quite easy to examine and verify despite incorporating a large number of variables. We also demonstrate that combining dynamic and static modelling approaches improves the coverage of the overall model and thereby anomaly detection performance.

1. Introduction

A wealth of information on vessel movements has become available through the use of the Automated Identification System (AIS), with much of it even filtering through to the public via the Internet. Surveillance authorities are interested in using this data to uncover threats to security, illegal trafficking or other risks. Whereas previously surveillance has suffered from a lack of data, electronic tracking has transformed the problem into one of overabundance, leading to a need for automated analysis.

The main goal of vessel behaviour analysis is to identify anomalies. As noted by Riveiro and Falkman [1], anomalies are detected either by using signature-based approaches or, as we do here, by developing a model representing normal behaviour, with anomalous behaviour being then identified by the extent of a vessel's deviation from normality. A common approach to creating normal models is to cluster the data around a set of points in a multi-dimensional feature space, with features such as longitude and latitude, speed and course [2]. Tracks that are within or near one of these clusters are considered normal, while the remainder are flagged as potential anomalies. Researchers use many different machine learning techniques to generate normality models from vessel movement data (typically AIS data), including the learning of Gaussian mixture models [2], support vector machines [3] and neural networks [4]. A disadvantage of these approaches is that they do not provide a transparent model that a human user, such as a surveillance officer, can understand, interact with and explore.

Here, we explore the use of Bayesian Networks (BNs) [5,6] for analysing vessel behaviour and detecting anomalies. While BNs have been widely applied for surveillance and anomaly detection (e.g., [7–10]), to date there have been only a few preliminary applications of BNs to maritime anomaly detection. As noted by Johansson and Falkman [11], however, BNs potentially have two substantial advantages in this domain over other types of models: (1) BN models are easily understood by people who are not BN specialists (which may include surveillance operators or other domain experts) and (2) they allow

* Corresponding author.
E-mail address: steven.mascaro@bayesian-intelligence.com (S. Mascaro).
URL: http://www.bayesian-intelligence.com (S. Mascaro).

0888-613X/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ijar.2013.03.012

for the straightforward incorporation of expert knowledge. They can also represent causal relations directly and, in that case, have the advantage of being more easily verified and validated, as we show in Section 3. We begin with a brief look at some of earlier approaches to anomaly detection.

1.1. Other approaches to anomaly detection

Support Vector Machines (SVMs) partition the multidimensional feature space, producing strict boundaries between clusters. In their simplest forms, SVMs suffer from a number of problems that have limited their use in vessel anomaly detection, including a lack of partial assignment, a restriction to binary classes, high computational complexity and difficulties in summarizing and communicating the learned models. Li et al. [3], 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 turning left or looping) 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 than an SVM trained on lower level features alone.

One commonly used model is the neural network [4,12], which consists of a network of processing nodes, input/output connections between nodes and weights attached to the connections. For 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, data mined neural networks of any moderate degree of complexity are almost completely opaque to human understanding, whereas their interpretation by surveillance operators is one of their primary purposes [13].

Gaussian Mixture Models (GMMs) have proven a popular choice for representing normality models of vessel behaviour [14, 2,15]. As its name implies, a GMM is a combination of multi-variate Gaussian distributions. These distributions aim to summarize how the training data cluster and spread in the multi-dimensional space. Kernel Density Estimators (KDEs) are a generalisation of GMMs, 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-experts to understand. Laxhammar et al. [15] 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.They found little extra value in using KDE methods over GMMs.

Das and Schneider [16,17] identify anomalous cases by finding unexpected dependencies between sets of attributes. They compare their approach to one using BNs and find their approach does better. We are skeptical inasmuch as their approach could very readily be adopted using Bayesian networks, while Bayesian network models also allow the identification of a large further class of anomalies not reflected simply by direct dependencies, for example, those represented only by conditional dependencies, which Das and his collaborators ignore.

1.2. BN-based approaches to anomaly detection

Given that anomalies just are events that are highly improbable under ordinary circumstances, Bayesian networks are a natural representation for reasoning about them. In particular, using a BN we can easily calculate:

$$P(e|m) \tag{1}$$

where e is an event (or evidence for an event) and m is the model. However, there is no generally accepted method of classifying an event as anomalous using a BN. Often (e.g., [8,11]), the probability above is tested against a threshold t:

$$P(e|m) < t \rightarrow \text{anomalous} \tag{2}$$

Or, if there is a sequence of events—as there is when trying to detect anomalous *behaviour*—these probabilities may be aggregated over time, as in:

$$\frac{1}{N}\sum_i P(e_i|m) < t \rightarrow \text{anomalous} \tag{3}$$

with N timesteps i. We can choose N either to range over the course of the entire behaviour (i.e., event sequence) or to restrict it to specific time windows.

An alternative approach to identifying anomalies is to check for conflicts within a set of evidence. This is similar to the approach that Das and Schneider [16] take, but within the context of BN inference. Jensen et al. [18] proposed a "conflict measure" to detect possible incoherence in evidence $\mathbf{E} = \{E_1 = e_1, \ldots, E_m = e_m\}$:

$$C(\mathbf{E}) = \log \frac{P(E_1 = e_1) \times \ldots \times P(E_m = e_m)}{P(\mathbf{E})}$$

This catches cases where each attribute is independently common but jointly uncommon. For example, Nielsen and Jensen [19] used this measure to identify when a coal based power plant has begun behaving abnormally. They do this by first learning a model of normal power plant operation using the plant's sensor information and then monitoring the sensors in an online environment, checking the probability of the readings to determine when to trigger an alarm. When the sensor readings are jointly less probable than what they would be independently (i.e., when $C(\mathbf{E}) > 0$), an alarm is triggered.

There have been a few cases in which BNs have been applied to maritime anomaly detection, although none are in deployment that we are aware of. Helldin and Riveiro [13] used BNs in anomaly detection with AIS data, focusing specifically on how the reasoning capabilities of a BN can assist surveillance system operators, such as by flagging potential anomalies.

Johansson and Falkman [11] used the constraint-based PC algorithm [20] to learn BNs from simulated data representing normal vessel behaviour. While they claimed their approach identifies a "reasonable proportion" of anomalous tracks, no specifics such as false (or true) positive rates were given, nor did they examine how their parameters affect anomaly detection.

Lane et al. [21] also apply BNs to maritime anomaly detection, focusing on their ability to fuse probabilities from different sources. They define five categories of anomalous ship behaviour: deviation from standard routes, unexpected AIS activity, unexpected port arrival, close approach and zone entry. They then define a threat (e.g., illegal exchange of goods at sea) as manifesting one or more of these behaviours, with each producing a certain probabilistic pattern. They produce a tree of conditional probabilities represented by a BN (that is, a threat causes certain behaviours, which in turn cause detections) to fuse together information about individual behaviour detections into an overall threat assessment. While a promising approach, much of the heavy lifting in terms of modelling is done prior to the use of the BN; the BN acts primarily as an aggregator.

1.3. Overview of our approach

In our study we data mined AIS data supplied by the Australian Defence Science and Technology Organisation (DSTO). Since many factors can contribute to the (ab)normality of a vessel's behaviour, in this study we also enhanced that data set by adding information such as weather and time, as well as vessel interactions. We used a metric BN learner, CaMML,[1] that flexibly allows many kinds of structural priors (e.g., directed arcs and tiers), aiding the learning of sensible models.

We investigated two approaches to model learning. First, we trained a model on the track data in its original time series form. For variables related to motion, we added new variables to represent the motion at both step k and $k + 1$, effectively making the data represent a dynamic Bayesian network (DBN), which have been used successfully for other kinds of anomaly detection (e.g., [10]). Second, we created single summary records of each track and learned static models from them. Summary data included average speed and course, number of stops, major stopping points and percentage of time travelling straight.

To assess the value of the networks in anomaly detection we took the common approach of using a measure for how probable a given track is according to the learned models of normality. This measure was applied to data sets representing both normal and synthetic anomalous tracks. In addition, we also mutated the presumed normal tracks to help us see how the network's probability estimates change. This led to a very interesting understanding of both the network's behaviour and the nature of the normal data set.

Next we describe our approach to the main components of this study, including details of the time series and track summary methods, the variables used by the BNs and the learning algorithms. We first analyse interesting aspects of the learned BNs, then present experimental results in Section 3, which demonstrate the value of Bayesian networks for anomaly detection.

2. The knowledge engineering process

Our knowledge engineering process was in the end complex, as shown in Fig. 1. We began by cleaning the raw AIS track data, merging data from other sources (including ship and weather data) and then, for each row, calculating new values of interest (such as average acceleration and the number of ships in the vicinity). This enhanced track data was used for both the time series and track summary models. At this point, the knowledge engineering process broke into two paths. For the time series path, we added one time-step delayed fields for the DBN, discretized each continuous field and then fed the data into the CaMML BN learner. For the track summary path, we created initial summaries, clustered the tracks and applied class labels to each track based on the clustering. We then discretized each continuous field in the summaries and fed the data into CaMML. For both paths, we chose priors supporting relations we already knew about (e.g., DBN arcs between two time steps, ship speed cannot cause weather). CaMML produces a range of possible BNs; we selected the highest posterior BNs of each type and performed evaluation with normal and anomalous tracks. Evaluation involved manual inspection of the networks as well as empirical testing using real and generated test data. The results of the evaluation were then used to refine each of the steps of the process. In the following sections we will describe the more interesting steps of this process in greater detail.

[1] CaMML is software developed over the past 15 years, see [6, Ch. 9] for an overview; here, we used the version available at `https://github.com/rodneyodonnell/CaMML`, described in [22].

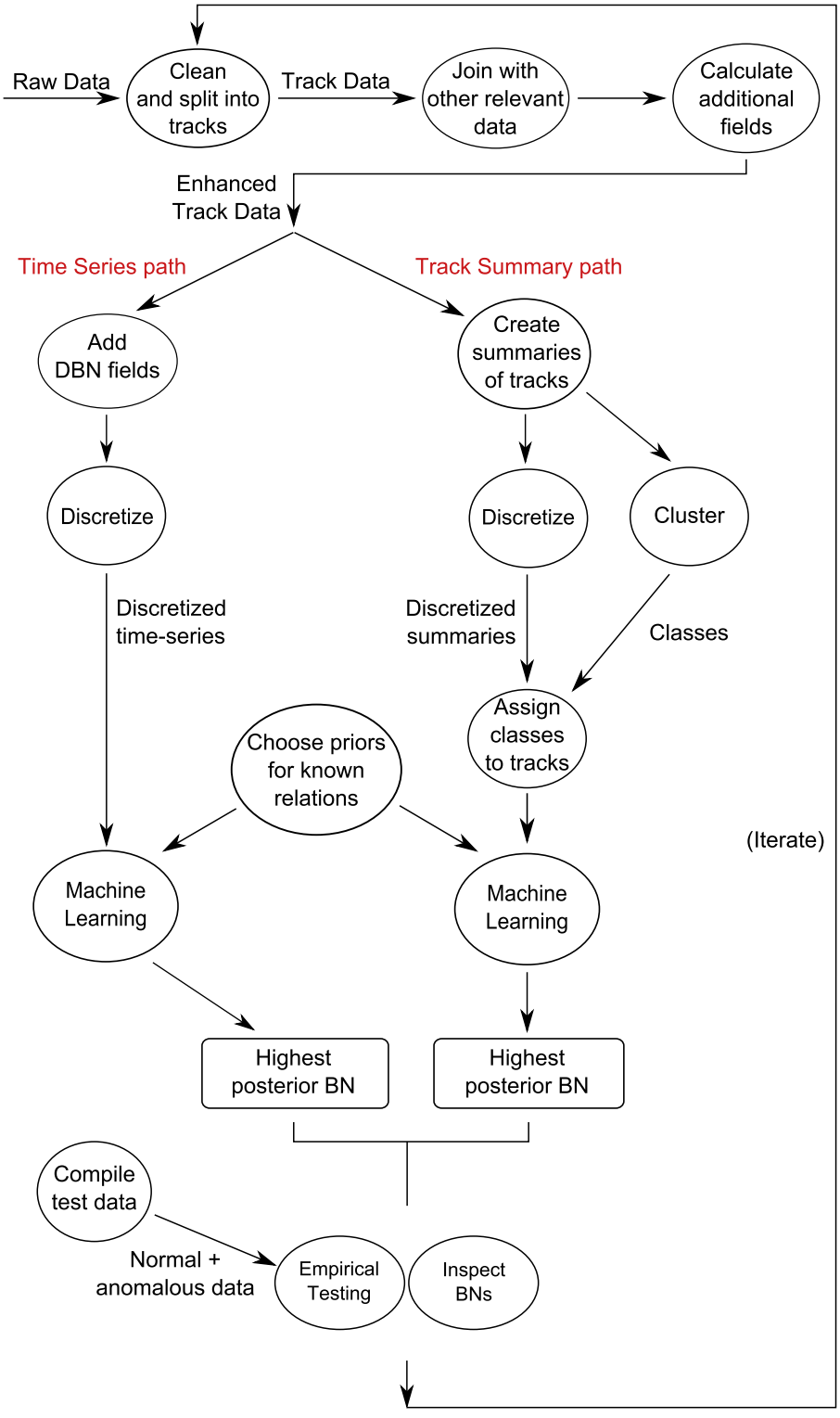


Fig. 1. The knowledge engineering process for the experiments.

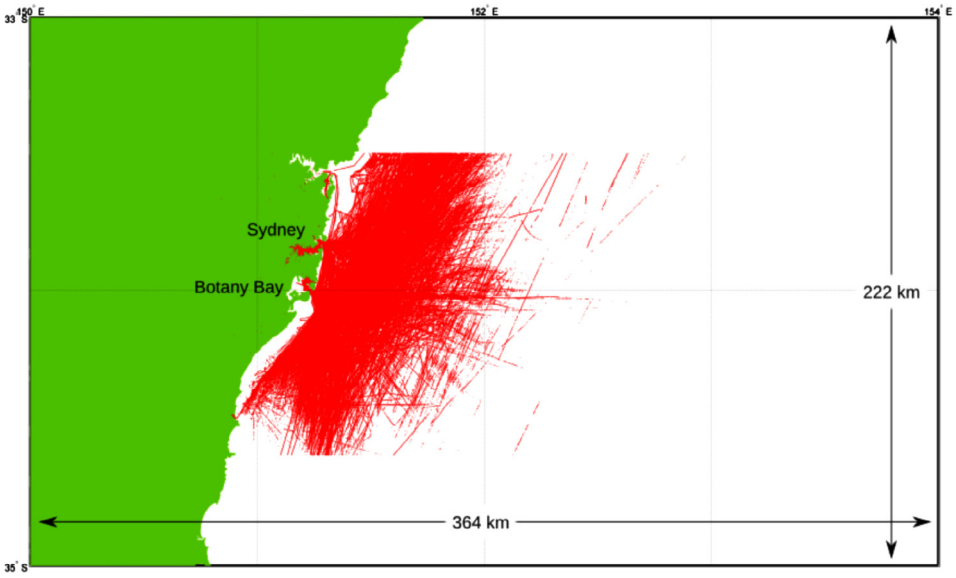


Fig. 2. The 2473 tracks in the AIS dataset.

Table 1
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.

MMSI	Timestamp	Lat	Lon	Speed	Course	Hdng
X	200905X	-33.X	151.X	18.7	49.9	46
X	200905X	-34.X	151.X	2.1	218	80
X	200905X	-33.X	151.X	0	0	511
X	200905X	-34.X	151.X	17.5	183	179
X	200905X	-33.X	151.X	1.2	28	64

2.1. The data

We used AIS data from May 1st to July 31st, 2009 for a section of the NSW coast framing Sydney harbour (see Fig. 2). The raw data consisted of just under 9.2 million rows with the vessel's MMSI (a numerical vessel identifier), a timestamp, the latitude and longitude, and the reported speed, course and heading (see Table 1).

The AIS data was cleaned and separated into 'tracks', first by assigning each record to a separate track based on the MMSI. We then cleaned the data in each track by interpolating each row's values to the nearest 10 second interval and eliminating duplicate data. The raw data contained many cases in which a single vessel transmits for much of the three month period of the data, with no indication of how many journeys it took within that period. Track splitting was performed so that each single track better corresponded to a single journey. We split a track record into multiple records when the vessel was stopped or not transmitting for 6 hours or more.[2] This yielded 2473 tracks across 544 unique MMSIs averaging 1995 rows each.

Vessel track anomaly detection models have been limited to kinematic variables, such as location, speed and course, coupled with the type of the vessel (e.g., [11]). One aim of our study was to investigate the possible advantages of considering additional factors. We added variables related to the ship itself (including type, dimensions and weight), the weather (such as temperature, cloud cover and wind speed), natural temporal factors (including hour of day and time since dawn or dusk), kinematic DBN nodes and elementary information on vessel interactions for both the time series and track summary models. Information about each ship was obtained from the public websites `marinetraffic.com` and `digital-seas.com` and also from the DSTO. Coverage was generally excellent; for example, only 13 of the 544 vessels lacked ship type information. On the few occasions in which data was missing we used a "missing" value. Weather information for the period was retrieved from the Australian Bureau of Meteorology, based on observation stations around Sydney harbour [23].

2.2. The models

We investigated two kinds of model based on two different forms of the training data. The first, the *time series model*, used the data in its original time series form. Each timestep in a track was associated with a set of variables, such as latitude,

[2] We note, however, that since such stops may themselves indicate an anomaly, deciding what constitutes a track warrants future investigation.

Table 2
Causal tiers for the variables in the time series model, given as hard priors to CaMML.

1st Tier	ShipType, ShipSize, Rainfall, MaxTemp, EstWindSpeed, EstOktas
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

longitude and speed, used directly in the BN. This approach, of course, has the advantage that learned models can be used in online analysis, but it may miss patterns at a broader time scale.

The second model, the *track summary model*, was learned from summaries of each track—e.g., identifying the number of times the vessel stopped, the stopping locations, etc. While track summaries cannot be used easily in real-time surveillance, they can capture patterns that occur at the time scale of the track as a whole. For example, if a vessel heads straight out to sea, turns around at a constant rate, then returns directly home, each timestep in the track may appear perfectly normal to any time series-based normality model. However, the behaviour embodied by the track as a whole may be anomalous and worthy of attention. The variables for each type of model are in Fig. 3 (see [24]).

2.3. Classification and discretization

Though not a major goal, we were additionally interested in whether the pre-processing summarization might help us directly to identify types of tracks and anomalies through clustering. We used Snob [25], a clustering and unsupervised classification tool comparable to AutoClass [26], to do this clustering, which gave a class variable for each track.

Discretization of variables was needed for both CaMML and Netica. To perform discretization, we used Snob in a second way, this time to classify each continuous variable in one dimension, with each discovered class becoming a state. Using Snob in this way allowed us to recover hidden regularities (for instance, common values for the Lat and Lon variables, due to commonly visited locations) and is similar to the attribute clustering approach taken by Li et al. [3].

2.4. The CaMML BN learner

CaMML (Causal discovery via MML) learns causal BNs from data using a stochastic search (MCMC) and score approach [27], where the score handles the trade-off between model simplicity and goodness of fit to the data. After learning the structure, we parameterized the model with a standard counting-based procedure [28], as did Johansson and Falkman [11].

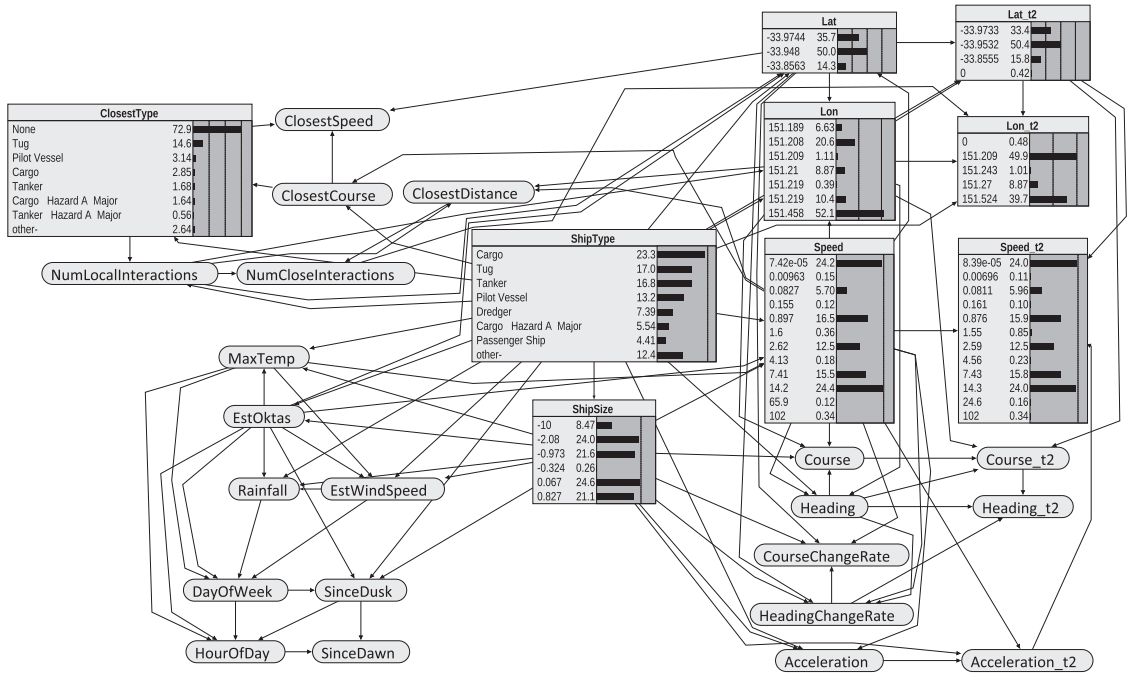
CaMML allows one to specify different types of expert priors [29]. These can be hard priors (e.g., an arc *must* be present or absent) or soft priors that specify the probability of certain arcs being present or, again, for more indirect dependencies. We used hard priors in the time series model to guarantee that the right DBN relationships held across time steps. We also specified priors in the form of "temporal tiers", putting a temporal partial order over variables, indicating which variables could *not* be ancestors of which others (Table 2).

2.5. Experimental methodology

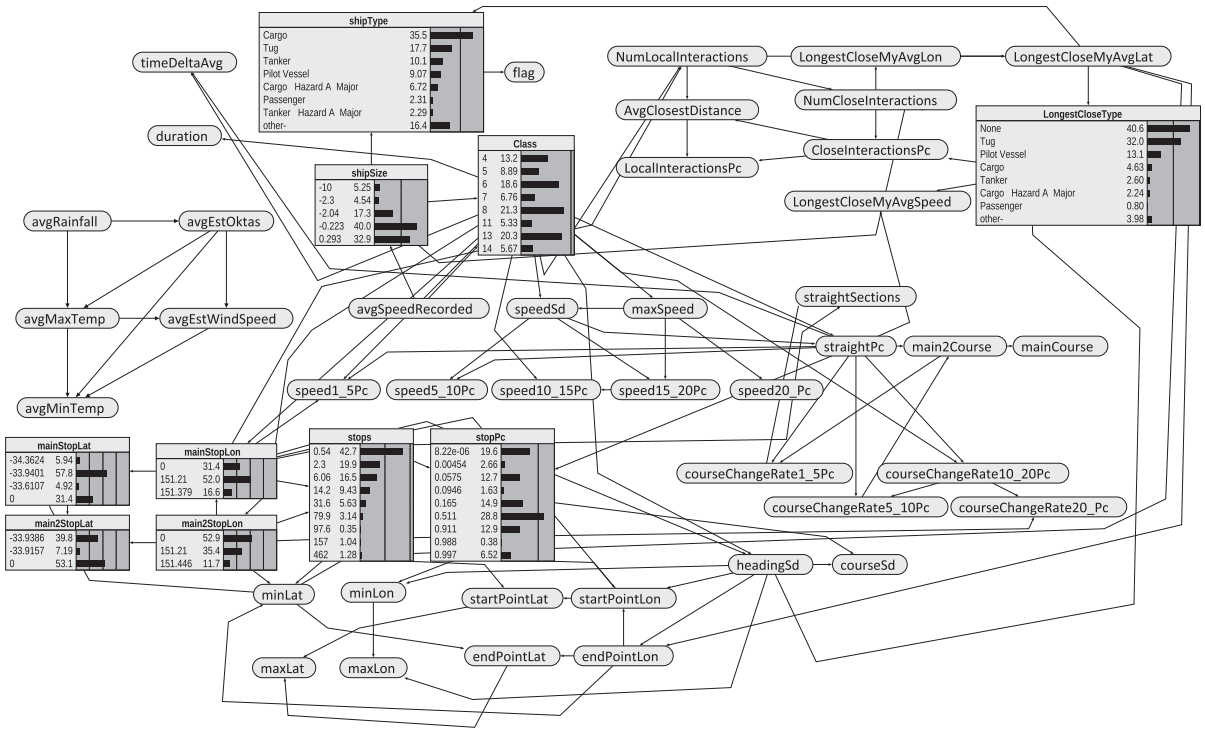
After pre-processing the data, we ran experiments using CaMML. We divided the data randomly (both time series and track summaries) into 80% (or 1978 tracks) for training and 20% for testing. As is common with anomaly detection models (e.g., [16,11]), the training data consisted of unfiltered real or 'normal' data in order to produce a model of normality against which we could assess deviations. We did a set of 10 runs of CaMML, taking CaMML's reported "best" (highest posterior) network each time, from which we derived the reported results.

3. Evaluation

We took two complementary methods to evaluating the learned models: inspection and empirical testing. The first method takes advantage of the transparency of Bayesian networks. Normally, this inspection would be done by domain experts to ensure that the network structure, parameters and model behaviour accord with expectation; here we had to take on this role ourselves. For cases in which the created model deviates from expectation, something may have gone wrong in an earlier part of the model development process, in which case we have cause to go back and fix the error (i.e., to iterate as per Fig. 1). This happened on occasion, though the cause was always due to something trivial, such as a mislabelled variable. Alternatively, particularly in cases where we learn from data, as we did here, we may have hit upon new knowledge about the domain. The second, more empirical, method of testing systematically examines model behaviour by feeding data from normal and anomalous cases into the model and comparing the results. While this is more systematic, it may miss things that human inspection of the BN can easily turn up.



(a)



(b)

Fig. 3. Example BNs produced by CaMML for the (a) time series data and (b) track summary data. For definitions of all the variables, see Appendix A of [24].

3.1. Inspecting the learned models

Fig. 3(a) shows an example BN produced by CaMML from the time series data, while Fig. 3(b) shows an example learned from the track summary data. First, note that there is an isolated subnetwork on the left in the track summary model, which

indicates that information about the weather (cloud cover, temperature and so forth) has no effect on the remainder of the network; this was found in all the track-summary networks learned. A possible explanation is that the limited time period didn't include enough examples of different weather patterns that might influence track behaviour. By contrast, learned time series models *included* weather variables, although their influence on kinematics variables was relatively weak.

It is clear that few arcs in the learned networks represent intuitive *direct* causal relations, other than the DBN arcs (given as hard priors) and the weather variables. Many of the other variables are simultaneous properties of the vessel, which will be correlated by hidden common ancestors. For example, while we would expect a ship's speed, size and course to be related, it isn't obvious what the underlying causes might be. They may be such things as the business the vessel belongs to, the purpose of its trip or the nature of its crew and contents. Some of these hidden causes will be partly captured by the ShipType, e.g., the purpose of a trip employing a cargo ship is almost always transport. This explains why that variable is the common cause of so many others in the time series models. In the track summary network this common cause role is assumed by the 'Class' variable instead.

Causal discovery relying on joint sample data very often gets arc directions wrong, in the anti-causal direction, because it is dependent upon sparse information about any uncovered collisions (where two parents of a child node are not themselves directly connected) to infer all arc directions. For example, Fig. 3(a) shows ShipType $\rightarrow$ Weather, for a variety of weather variables. Of course, ship type cannot affect the weather. A plausible interpretation of this is that weather conditions *do* affect which types of ship put to sea, so the arc directions are reversed. A simple and very effective method of dealing with this problem would be to introduce extra prior constraints, such as putting weather variables into a zeroeth Tier.

Interactively exploring Bayesian networks is very natural and here turned up many points of interest. In confirming the reasonableness of the time series model, we found that entering 'Tug' or 'Pilot Vessel' into the 'ShipType' variable significantly increases the chance of another vessel being nearby. Cargo ships, on the other hand, travel mostly solo and tankers almost exclusively so. Ship sizes (i.e., the 'ShipSize' variable) are highly correlated with position via the 'ShipType' variable, with larger vessels especially appearing 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. The track summary model shows that cargo ships and tankers spend most of their time travelling straight, while tug directions are variable. Tugs also tend to stop in different locations from cargo ships, and for longer periods.

We can gain greater insight into the 10 best models that CaMML found for both approaches by using network summary matrices (used previously with CaMML in [30]). These summary matrices show how often CaMML produces an arc for each pair of variables across the 10 generated models, shading the cell in proportion to the arc frequency. This gives us a strong visualisation of the overall variability in the structure across the generated networks, while also allowing us to quickly drill down to examine the variability of individual arcs. Summaries of the 10 generated models for both approaches are shown in the arc frequency matrices of Figs. 4 and 5. 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—bright green indicates every network contains the arc, pale green indicates that half of the networks contain it and white indicates no networks contain it. There are two items to note about these arc frequency matrices: they are relatively sparse 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%. Therefore, CaMML did not have much trouble converging upon a BN structure. 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.

The last five columns and the last five rows of Fig. 5 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.

3.2. Empirical testing

To test our models, we first needed to define some method for assigning 'anomalousness' to each track given a model, corresponding to low probabilities attributed to them by the normality model [8]. However, unlike Cansado and Soto, we think choosing any particular threshold for deciding when tracks are anomalous would be arbitrary. In real applications a specific threshold may present itself as most suitable, but in general we feel it is better to represent the probability itself to surveillance operators, albeit in a more convenient form. [3]

Thus, for track summary data, we first computed each track's prior probability given the normality model. Since these probabilities are usually very low (around 10^{-11}) we took the negative log (base 2) to produce an "anomaly score" (i.e., the number of bits required to describe the data, given the model). Put simply, the higher the anomaly score, the less probable the track.

For time series networks we took a similar approach, but fed each timestep of the track into the network to yield a probability estimate for that timestep. We then took the average probability over all timesteps to generate a negative log anomaly score. For time series data it is possible, of course, to base anomaly criteria upon *changes* in the track over time.

[3] If operators are unable to monitor each track probability, then it would be best to extend the networks into decision networks, thereby avoiding the artificial problem of threshold selection for probabilities altogether.

	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	0.80	1.00	0.10	0.20	0.20	0.10	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.20	0.10	0.10	0.00	0.00	0.00
Rainfall	0.70		0.80	0.80	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	0.10	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	0.10	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	0.00	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	0.20
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	0.00	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	0.00	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	0.00	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	0.00	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	0.60	0.00	0.10	0.00		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	0.00	0.50	0.00	0.00	0.60
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	0.10	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	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
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
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	0.00		0.70	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	0.00	0.30		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	0.00	0.20	0.20	0.00	0.00	0.00		0.00	0.00	0.10
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	0.00	0.20	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
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	

Fig. 4. 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.

Johansson and Falkman [11], for example, used sliding windows across a track, looking for any anomalous *windows*. For this study, however, we focused on criteria for assessing the tracks as wholes, leaving this kind of alternative for future investigation.

Calculating anomaly scores for all the tracks in our data set and plotting the distribution of the results (using a Gaussian Kernel Density Estimator [KDE]), we obtained Fig. 6. These show a fair amount of diversity among anomaly scores, i.e., they do not simply clump around the lowest possible score. Note that the scores produced by the time series model are quite distinct from those of the track summary model. One likely reason is that the track summary scores are simply based on more variables, making each instance more specific and less probable. However, there is a surprisingly small correlation between the two sets of scores ($r = 0.159$; $p < 0.001$),[4] showing that the two models look at different aspects of each track. Indeed, as we will see below, they complement each other when performing anomaly detection.

To perform empirical testing we needed test data. Unfortunately, we did not have any access to known anomalous tracks nor are there any standardized or publicly available vessel track data sets containing anomalies (or otherwise). Nevertheless, there are many ways to create anomalous data. Cansado and Soto [8] generated anomalies by modifying selected attributes to random values within their ranges. Johansson and Falkman [11] generated anomalous data using anomalous models.[5] Here, we tried three approaches, partly inspired by these previous methods: modifying instances by swapping incorrect ship type information, splicing tracks together, and drawing anomalous tracks.

[4] Earlier iterations with cruder discretizations and more variables in common showed a stronger correlation—however, as models grew more detailed, the correlation shrank.

[5] Wang et al. [9], without known anomalous data, simply weakened their threshold to find "anomalies", whether they were there are not!

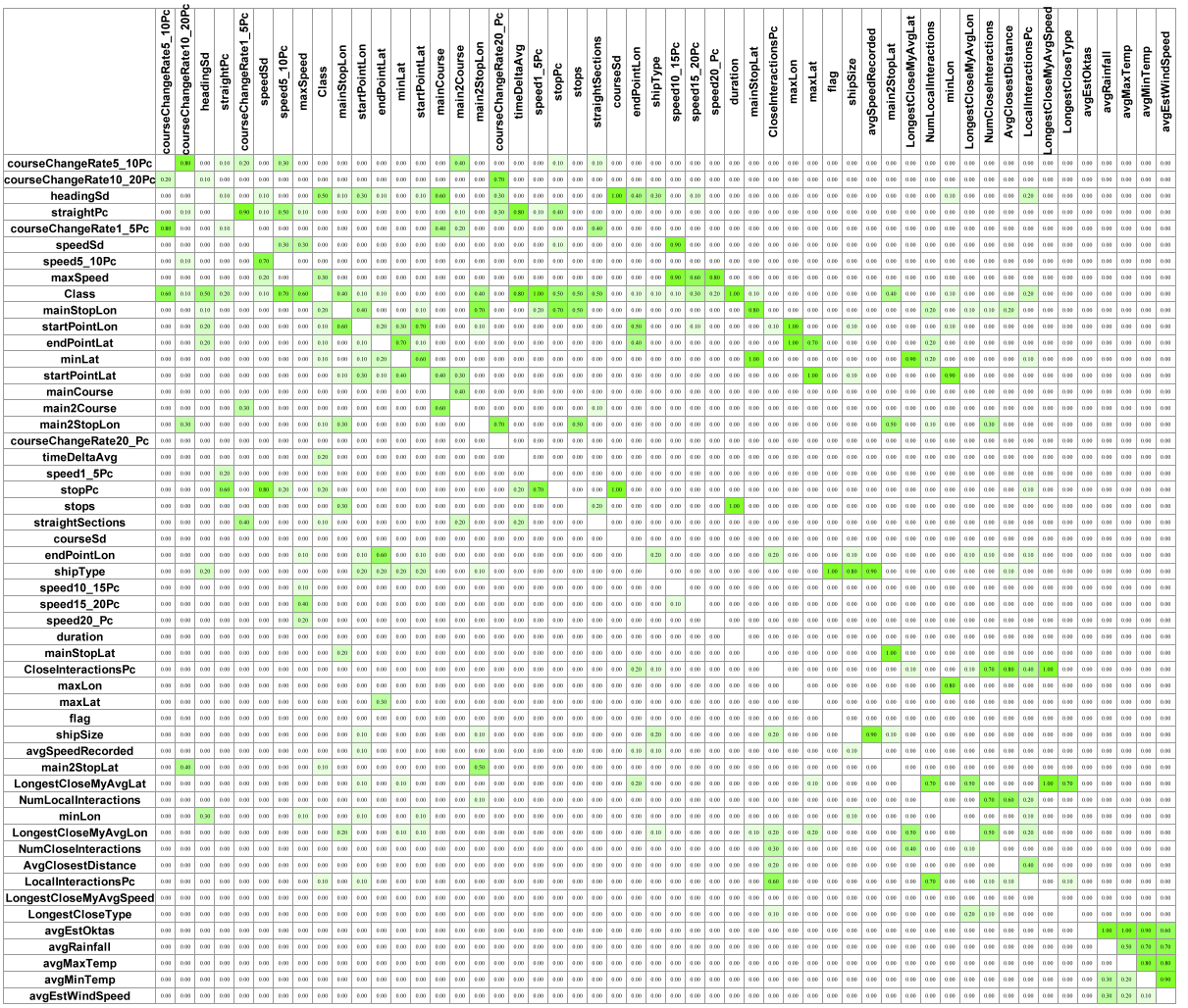


Fig. 5. A summary arc matrix of the 10 BNs generated for the track summary data. Each cell represents an arc (row to column) and displays the proportion of networks containing that arc.

3.2.1. The false ship effect

For each track in the training set, the ship type information was swapped with that of another randomly selected ship of a different type, leaving the track data alone. The intention was to create anomalous tracks that were a clear mismatch with their labelled type. Fig. 7(a) shows how this affected the anomaly score. In most cases the false ship effect (that is, the anomaly score for the false track minus the score for the true original track) is positive, increasing the anomaly score. The false ship effect for the time series model is positive in around 87.2% of the cases as opposed to 69.4% of cases for the track summary model. Fig. 7(b) and (c) 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 ($r = -0.70$, $p \ll 0.01$). This also occurs with the track summary model, to a smaller extent ($r = -0.31$, $p \ll 0.01$).

Sometimes tracks have become *more* probable given incorrect ship information, which itself seems anomalous! To be sure, many of the ship types are in fact quite similar (e.g., there are several sub-categories of cargo ship) so switching between these may randomly produce a more likely track. However, this does not account for all the cases. Some might be due simply to a regression from a high anomaly score towards the mean. But others may actually have been mislabelled or, more intriguingly, may simply have indeed been behaving anomalously according to their type. This suggests another approach to anomaly detection based on a form of *abduction*. Abduction (due to [31]) is the process of positing an *explanation* for some observation. 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 to gathering new evidence.[6]

If a track is assigned a certain probability given the normality 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,

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

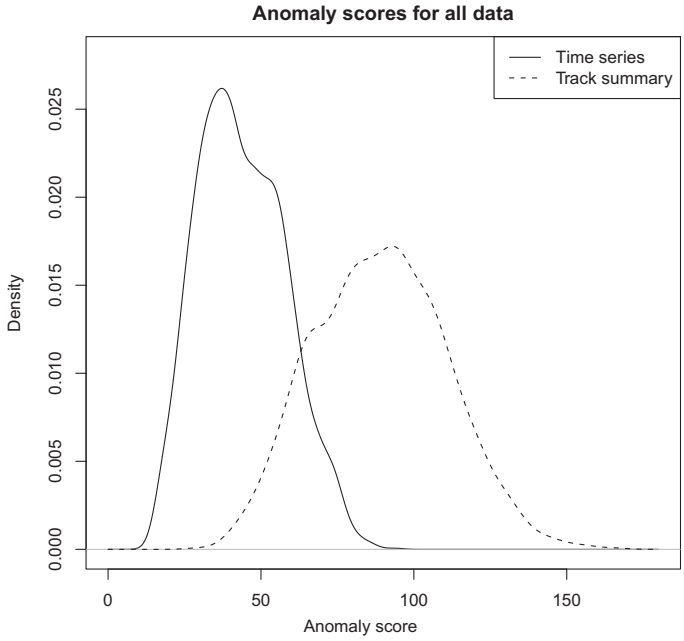


Fig. 6. The KDE distributions of anomaly scores for all tracks in the data set according to the (a) time series and (b) track summary networks.

the false ship effect simply being less than 0 may be a natural threshold for identifying potential anomalies; such a possibility warrants a more detailed investigation. This has some of the spirit of the Jensen et al. [18] conflict measure, while clearly being different.

3.2.2. Track splices

We also created anomalous tracks by splicing random tracks together. This allowed us to test our models for their ability to detect discontinuities as well as major changes in behaviour. Specifically, we selected 140 tracks at random and replaced their tails with those of other tracks (retaining the times and types of the original track). 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 ($p \ll 0.01$). Both scores are significantly different from the average anomaly score for all data of 89.0.

With the spliced tracks, as we expected the track summary model performed slightly better than the time series approach, because the time series model is not able to detect unusual behaviour across the whole track. Tracks put together from ships of different types produced an average score of 48.9 while those spliced from same types had a score of 45.6; while a small difference, this was statistically significant ($p < 0.01$). In addition, while the higher score was significantly different ($p < 0.01$) from 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 of the track summaries.

3.2.3. Manually drawn anomalies

Finally, we tested models using anomalous tracks drawn with a mouse over a map, where the mouse location and speed generated the vessel location and speed respectively. Other factors were created randomly, including the time and duration, noise in the data, vessel details and maximum speed. This allowed us to compare the performance of both models across several different categories of anomalous behaviour, thereby shedding light on the strengths and weaknesses of each model. Anomalous behaviour in these tracks included very noisy data, close interactions with many other vessels, vessels that circle in unusual patterns, vessels travelling over land, overly short tracks in the middle of the sea and vessels behaving against their type. In all, 107 such 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 Fig. 8 are the result. The ROC curves demonstrate the tradeoffs that can be made (if we were to settle on specific thresholds for anomalies) between false positives and true positives; the greater the area under the curve (AUC), the less severe the tradeoff needs to be. We can see that the track summary model (with an AUC of 0.780) performs better than the time series model (AUC 0.712). Adding the anomaly scores from the two models together (in effect, creating a combined model with equal weight given to each individual model) performs better again (AUC 0.809). Table 3 shows the average scores each model yielded for various kinds of anomalous tracks. We can see that both models easily detected the tracks containing too many close interactions (average scores of 139.9 and 75.8, against the

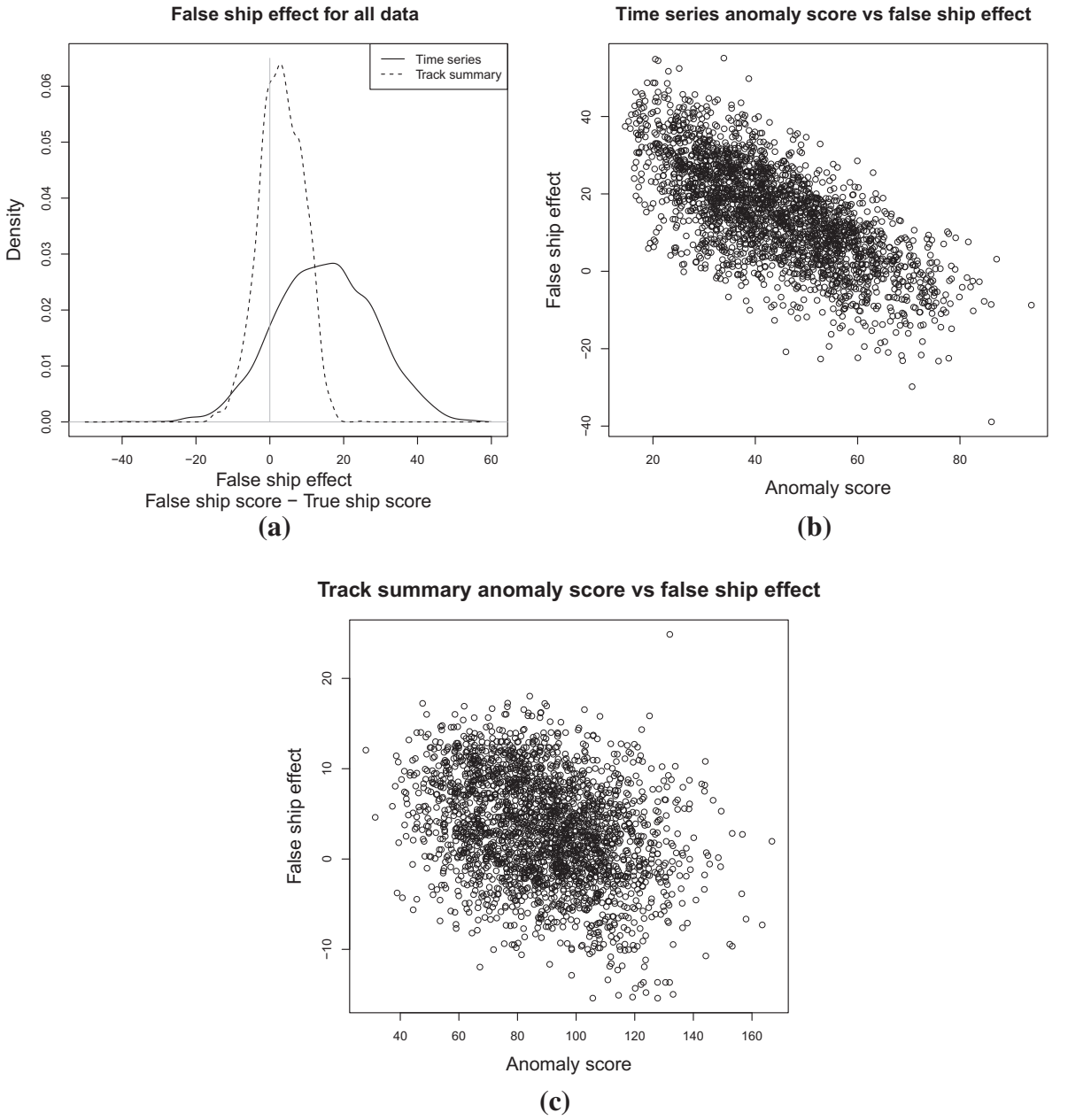


Fig. 7. False ship effect: (a) anomaly score differences for false ship information versus correct ship information, sorted by score; and scatter plots for (b) time series and (c) track summary networks.

test averages of 90.8 and 45.7, giving Deltas 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.

3.2.4. Testing on Johansson and Falkman's simulated data

We also applied our methods to the simulated data used by Johansson and Falkman [11], both normal and anomalous. Our models are not well suited to the simulated data since our approach depends heavily on having much more external data available (such as weather, sensible vessel interactions, time of day and year, etc.) but nonetheless performed reasonably well. In particular, with the track summary model, anomalous tracks received an average anomaly score of 22, while normal tracks averaged 17 ($p \ll 0.01$); while in the time series model, anomalous tracks received an average score of 29, with normal tracks averaging 25 ($p \ll 0.01$). When we calculated the ROC curves (shown in Fig. 9), we found that the time series

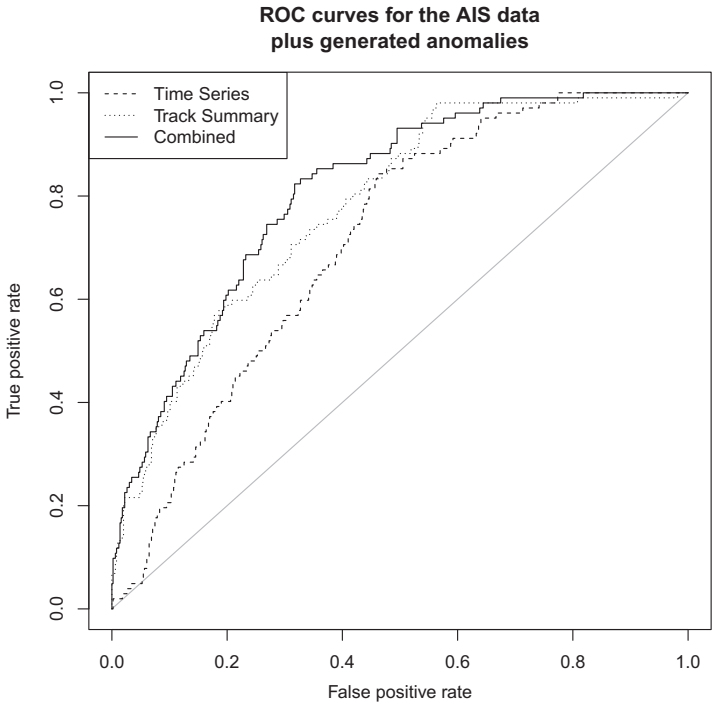


Fig. 8. ROC curves for test data, containing both normal tracks and manually created anomalous tracks, given the time series, track summary and combined models.

Table 3
Average anomaly scores for various categories of anomaly. Columns headed 'Δ' indicate the difference from the average score for normal test tracks.

	Track summary score	Δ	Time series score	Δ
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	126.9	+36.1	53.9	+8.2

model performed better with this data set with an AUC of 0.691, over the track summary AUC of 0.652. This was likely due to a lack of extended ship type information. The combined model again performs better than both individually, with an AUC of 0.727. Fig. 10 shows the ROC curves from both the AIS data and the Johansson and Falkman data using the combined models.

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 much higher than the respective average scores for the anomalous tracks.

4. Conclusion

We have demonstrated Bayesian Networks are a promising tool for detecting anomalies in vessel tracks. By using a BN learner on AIS data supplemented by additional real world data, we produced both dynamic and static networks, which demonstrated distinct and complementary strengths in identifying anomalies. Thus, we were able to improve anomaly detection by combining the dynamic and static networks.

Our work suggests the technique of combining models at different time scales should be an important part of the knowledge engineering process for anomaly detection. Normality models for tracks produced by cars, planes and humans

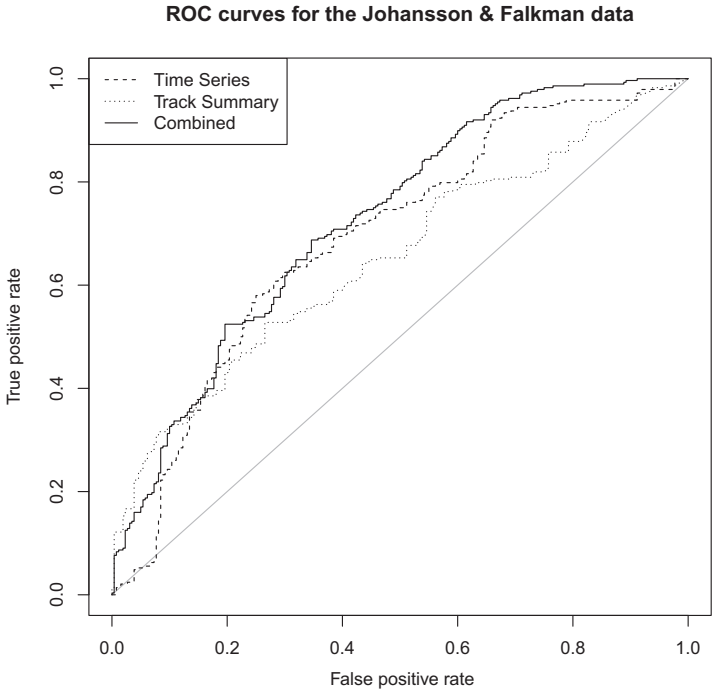


Fig. 9. ROC curves for the Johansson and Falkman data using the time series, track summary and combined models.

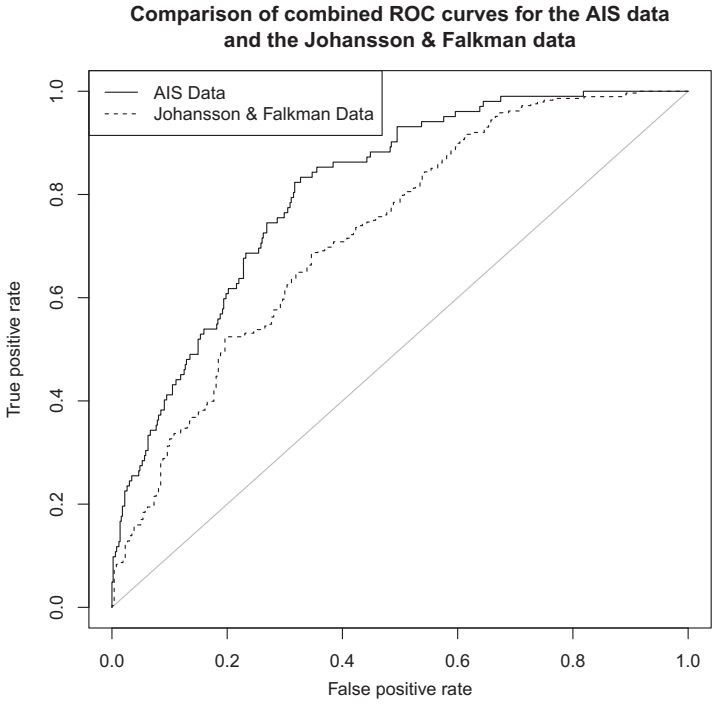


Fig. 10. Comparison of ROC curves for the AIS data and Johansson and Falkman data using the combined models.

could all benefit from the approach. And the benefits of this technique are not limited to anomaly detection models. Many kinds of dynamic model stand to benefit by the addition of a higher level summary static model—some obvious cases that come to mind include behaviour recognition and motion prediction. Our work further suggests that learning networks at still additional time scales, intermediate between the full track summary and each timestep, may improve performance even further.

Other opportunities for improving anomaly detection (some of which we have begun exploring) include better attribute selection, improved discretization, incorporating better expert priors, and identifying track starting and ending points. Variations on our simple approach of gauging anomalies via track probabilities are also worth looking at. For example, for the time series model, it is possible to calculate anomaly scores for a small window around time t as a way of detecting anomalous portions of a track. As another example, our investigation of the false ship effect suggested that modifications of records meant to reduce their probabilities may signal anomalies when they fail to do so. We think an abductive approach may also aid anomaly detection. A simple example may serve: if a vessel claims to have a particular weight, we can see if alternative weights provide a better explanation of its track data. If our model decrees the track more probable given more heft, the vessel may be harbouring cargo it should not.

Finally, while there are a handful of approaches tackling anomaly detection with uncertainty models, none as yet take advantage of the decision-making capabilities of BNs. This would be especially useful in issuing automated alerts—by categorizing threats and attaching utilities to each type of warning, assessing levels of warning to issue could be done in a systematic and transparent way.

Acknowledgements

The authors would like to thank the DSTO for supporting this work, 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.

References

[1] M. Riveiro, G. Falkman, Evaluating the usability of visualizations of normal behavioral models for analytical reasoning, in: Computer Graphics, Imaging and Visualization (CGIV), 2010 Seventh International Conference on, IEEE, pp. 179–185.
[2] R. Laxhammar, Anomaly detection for sea surveillance, in: The 11th International Conference on Information Fusion, pp. 55–62.
[3] X. Li, J. Han, S. Kim, Motion-Alert: Automatic anomaly detection in massive moving objects, Proceedings of the 2006 IEEE Intelligence and Security Informatics Conference (ISI 2006), Springer, Berlin, 2006, pp. 166–177.
[4] B.J. Rhodes, N.A. Bomberger, T.M. Freyman, W. Kreamer, L. Kirschner, A.C. L'Italien, W. Mungovan, C. Stauffer, L. Stolzar, A.M. Waxman, M. Seibert, SeeCoast: Persistent surveillance and automated scene understanding for ports and coastal areas, in: Proceedings of SPIE, vol. 6578, pp. 65781M.1–65781M.12.
[5] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, San Mateo, CA, 1988.
[6] K.B. Korb, A.E. Nicholson, Bayesian Artificial Intelligence, second ed., Chapman & Hall/CRC Press, 2010.
[7] W. Wong, A. Moore, G. Cooper, M. Wagner, Bayesian network anomaly pattern detection for disease outbreaks, in: International Conference on Machine Learning, vol. 20, pp. 808–815.
[8] A. Cansado, A. Soto, Unsupervised anomaly detection in large databases using Bayesian networks, Applied Artificial Intelligence 22 (2008) 309–330.
[9] X. Wang, J. Lizier, O. Obst, M. Prokopenko, P. Wang, Spatiotemporal anomaly detection in gas monitoring sensor networks, Wireless Sensor Networks (2008) 90–105.
[10] C. Loy, T. Xiang, S. Gong, Detecting and discriminating behavioural anomalies, Pattern Recognition 44 (2010) 117–132.
[11] F. Johansson, G. Falkman, Detection of vessel anomalies—a Bayesian network approach, in: International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2007, pp. 395–400.
[12] B. Rhodes, N. Bomberger, M. Seibert, A. Waxman, Maritime situation monitoring and awareness using learning mechanisms, in: Military Communications Conference, pp. 646–652.
[13] T. Helldin, M. Riveiro, Explanation methods for Bayesian networks: Review and application to a maritime scenario, in: Proceeings of the 3rd Annual Skövde Workshop on Information Fusion Topics (SWIFT 2009), pp. 11–16.
[14] J.B. Kraiman, S.L. Arouh, M.L. Webb, Automated anomaly detection processor, in: Proceedings of SPIE, vol. 4716, pp. 128–137.
[15] R. Laxhammar, G. Falkman, E. Sviestins, 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.
[16] K. Das, J. Schneider, Detecting anomalous records in categorical datasets, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 220–229.
[17] K. Das, J. Schneider, D. Neill, Anomaly pattern detection in categorical datasets, in: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 169–176.
[18] F. Jensen, B. Chamberlain, T. Nordahl, F. Jensen, Analysis in hugin of data conflict, in: Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence, Elsevier Science Inc., pp. 519–528.
[19] T. Nielsen, F. Jensen, On-line alert systems for production plants: A conflict based approach, International Journal of Approximate Reasoning 45 (2007) 255–270.
[20] P. Spirtes, C. Glymour, R. Scheines, Causation, Prediction and Search, Springer Verlag, 1993.
[21] R. Lane, D. Nevell, S. Hayward, T. Beaney, Maritime anomaly detection and threat assessment, in: Information Fusion (FUSION), 2010 13th Conference on, IEEE, pp. 1–8.
[22] R. O'Donnell, Flexible Causal Discovery with MML, Ph.D. thesis, Monash University, 2010.
[23] Bureau of Meteorology, Daily weather observations, May–July 2009, http://www.bom.gov.au/climate/dwo/IDCJDW2124.latest.shtml, 2010.
[24] S. Mascaro, K.B. Korb, A.E. Nicholson, Learning normal vessel behaviour from AIS data with Bayesian Networks at two time scales, Technical Report TR 2010/4, Bayesian Intelligence, 2010.
[25] C.S. Wallace, P.R. Freeman, Single factor estimation by MML, Journal of the Royal Statistical Society B 54 (1992) 195–209.
[26] P. Cheeseman, J. Stutz, M. Self, J. Kelly, W. Taylor, D. Freeman, Bayesian classification, in: AAAI 88, pp. 607–611.
[27] C.S. Wallace, K.B. Korb, Learning linear causal models by MML sampling, in: A. Gammerman (Ed.), Causal Models and Intelligent Data Management, Springer-Verlag, Heidelberg, 1999, pp. 89–111.
[28] D. Heckerman, A tutorial on learning with Bayesian networks, in: M. Jordan (Ed.), Learning in Graphical Models, MIT, 1998, pp. 301–354.
[29] R. ODonnell, A. Nicholson, B. Han, K. Korb, M. Alam, L. Hope, Causal discovery with prior information, in: A. Sattar, B.-H. Kang (Eds.), AI 2006, vol. 4304 of LNCS, Springer, Berlin, 2006, pp. 1162–1167.
[30] M. Julia Flores, A. Nicholson, A. Brunskill, K. Korb, S. Mascaro, Incorporating expert knowledge when learning bayesian network structure: a medical case study, Artificial Intelligence in Medicine (2011)
[31] C. Peirce, Abduction and induction, Philosophical Writings of Peirce, Dover Books, New York, 1955, pp. 150–156.