Anomaly Detection in Vessel Tracks using Bayesian networks

Steven Mascaro
Bayesian Intelligence Pty Ltd,
www.bayesian-intelligence.com

Ann E. Nicholson
Clayton School of IT,
Monash University, Australia

Kevin B. Korb
Clayton School of IT,
Monash University, Australia

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 tackle anomaly detection with 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. Combining the mined models improves performance in a variety of cases, demonstrating that learning Bayesian Networks from track data is a promising approach to anomaly detection.

1 INTRODUCTION

A wealth of information on vessel movements has become available to authorities through the use of the Automated Identification System (AIS). Much of this data has even filtered through to the general public via the Internet. Surveillance authorities are interested in using this data to uncover threats to security, illegal trafficking or other risks. While in the past, surveillance has suffered from a lack of solid 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. This requires the development of a model representing normal behaviour, with anomalous behaviour being then identified by the extent of a vessel's deviation from normality. A common 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 (Laxhammar, 2008). Tracks that are within or near one of these clusters may be 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), 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. A disadvantage of these approaches is that they do not provide a causal 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) (Pearl, 1988; Korb and Nicholson, 2010) for analysing vessel behaviour and detecting anomalies. While BNs have been widely applied for surveillance and anomaly detection (e.g., Wong et al., 2003; Cansado and Soto, 2008; Wang et al., 2008; Loy et al., 2010), to date there have been only a few preliminary applications of BNs to maritime anomaly detection. As noted by Johansson and Falkman (2007), however, BNs potentially have two substantial advantages in this domain over other types of models: 1) BN models are easily understood by non-specialists and 2) they allow 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.

Johansson and Falkman (2007) used the constraint-based PC algorithm (Spirtes et al., 1993) to learn BNs from simulated data representing normal vessel behaviour. While they claimed their approach identifies a "reasonable proportion" of anomalous tracks, while missing others, no specifics such as false (or true) positive rates were given, nor did they examine how their parameters affect anomaly detection. Helldin and Riveiro (2009) also looked at the use of BNs in anomaly detection with AIS data, but focused specifically on how the reasoning capabilities of a BN can assist surveillance system operators, such as by flagging potential anomalies, but they did not look at learning BNs from the data.

Outside of the maritime domain, Wong et al. (2003)

use BNs to detect disease outbreaks by detecting anomalous patterns in health care data, such as an upswing in the number of people with flu or an unusual decrease in the number of people buying decongestants. Wong et al. use a method called WSARE (What's Strange About Recent Events) to detect when a temporal stream of such data begins deviating from its own baseline profile. This differs from our approach here in that we are concerned with tracks as a whole, rather than trying to identify if and when a track has begun deviating from a normal baseline.

In our study here 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 (Wallace and Korb, 1999), that flexibly allows various 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., Loy et al., 2010). Second, we also 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 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 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 APPROACH

While our basic approach is well known — applying a BN learner to produce normality models to be used in assessing degrees of anomaly — in practice the experimental workflow was complex, as shown in Figure 1.

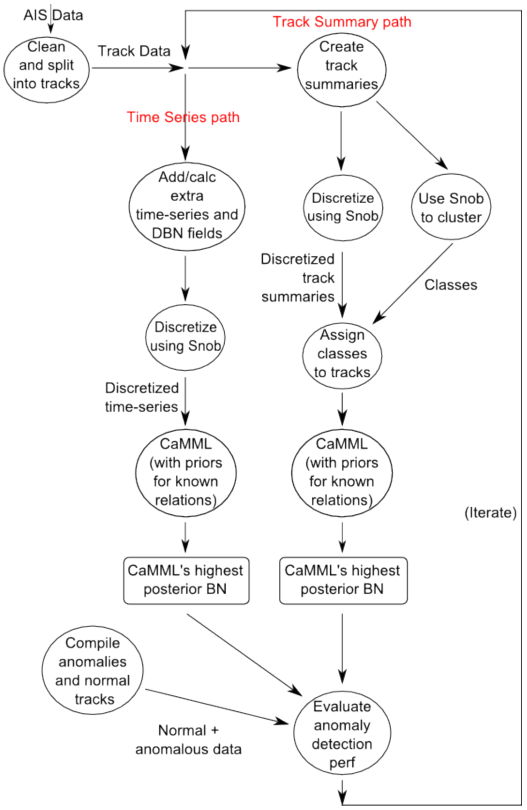

AIS Data
Clean and split into tracks
Track Data
Track Summary path
Create track summaries
Time Series path
Add/calc extra time-series and DBN fields
Discretize using Snob
Use Snob to cluster
Discretized track summaries
Classes
Discretize using Snob
Assign classes to tracks
Discretized time-series
CaMML (with priors for known relations)
CaMML (with priors for known relations)
(Iterate)
CaMML's highest posterior BN
CaMML's highest posterior BN
Compile anomalies and normal tracks
Normal + anomalous data
Evaluate anomaly detection perf

Figure 1: Workflow for the experiments.

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 Figure 2). The raw data initially contained just under 9.2 million rows and each row consisted of seven fields: the vessel's MMSI (a nine digit numerical vessel identifier), a timestamp, the latitude and longitude of the vessel, and its reported speed, course and heading (see Table 1). We did not use the MMSI directly in learning, but did use it in pre-processing and to locate additional information about the vessel.

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 rounding (and interpolating) each row to the nearest 10 second interval and eliminating duplicate data. However, since the raw data contained many cases in which a single vessel transmits for much of the three month period of the data, further track splitting was required. We split a track record into multiple records when the vessel was stopped or not transmitting for 6 hours or more.[1] This yielded 2,473 tracks across 544 unique MMSIs averaging 1,995 rows each.

Vessel track anomaly detection models have been lim-

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

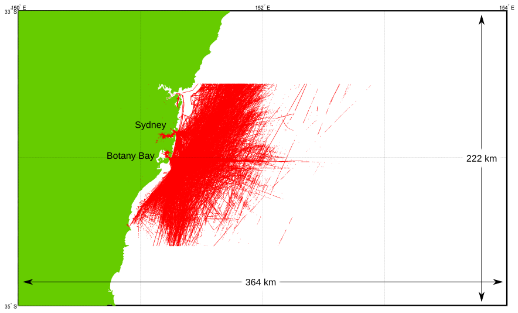


Figure 2: Example AIS tracks.

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

ited to kinematic variables, such as location, speed and course, coupled with the type of the vessel (e.g., Johansson and Falkman, 2007). 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 three locations: 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 website, based on observation stations around Sydney harbour (Bureau of Meteorology, 2010).

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**, uses the data in its original time series form. Each timestep in a track was 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 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 based on summaries of each track — e.g., identifying for a given track the number of times the vessel stops, the main stopping locations, etc. While track summaries cannot be used as 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 Figure 3 (see Mascaro et al., 2010).

2.3 CLASSIFICATION AND DISCRETIZATION

We were interested in whether the pre-processing summarization might help us directly to identify types of tracks and anomalies. To test this, we classified the summary tracks using Snob (Wallace and Freeman, 1992), an unsupervised clustering tool comparable to AutoClass (Cheeseman et al., 1988), producing a class variable for each track (see 'Class' node in Figure 3(b)).

Discretization of variables in the data set was needed for technical reasons: (1) the version of CaMML that allows structural priors requires discrete data and (2) we used Netica, which also requires discrete variables. To perform discretization, we again used Snob to classify each continuous variable in one dimension, with each discovered class becoming a state. Using Snob in this way allowed us to recover any hidden regularities 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 3a contains lower probability states wedged amongst higher probability states. One might expect to see a more even distribution, however Snob has identified 12 underlying classes corresponding to these 12 states — some of which are much more frequent than their neighbours.

2.4 THE CaMML BN LEARNER

In this work, we make use of the CaMML BN learner (Wallace and Korb, 1999). CaMML (Causal discovery via MML) learns causal BNs from data using a stochastic search (MCMC) and score approach. After learning the structure, we parameterized the model with a standard counting-based procedure (Heckerman, 1998), as did Johansson and Falkman (2007).

CaMML allows one to specify different types of expert priors (ODonnell et al., 2006). These can be hard priors (e.g., an arc *must* be present or absent) or soft priors that specify the probability of certain arcs connecting pairs of variables; other soft priors for more indirect dependencies can also be specified. Here, we used some simple hard priors in the time series model

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

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 and so 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 1,978 tracks) for training and 20% for testing. As is common with anomaly detection models (e.g. Das and Schneider, 2007; Johansson and Falkman, 2007), 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, using different seeds, taking CaMML's reported "best" (highest posterior) network each time, from which we derived the reported results.

3 EVALUATION

3.1 INTERPRETING THE LEARNED MODELS

Figure 3(a) shows an example BN produced by CaMML from the time series data, while Figure 3(b) shows an example learned from the track summary data. 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, Figure 3(a) shows ShipType→Weather, for a variety of weather variables. Of course, ship type cannot affect the weather. A plausible interpretation of this result is that weather conditions *do* affect which types of ship put to sea, so, if anything, the arc directions here are reversed. The simplest and very effective method of dealing with this problem is to introduce extra prior constraints, such as putting some weather variables into a zeroeth Tier.

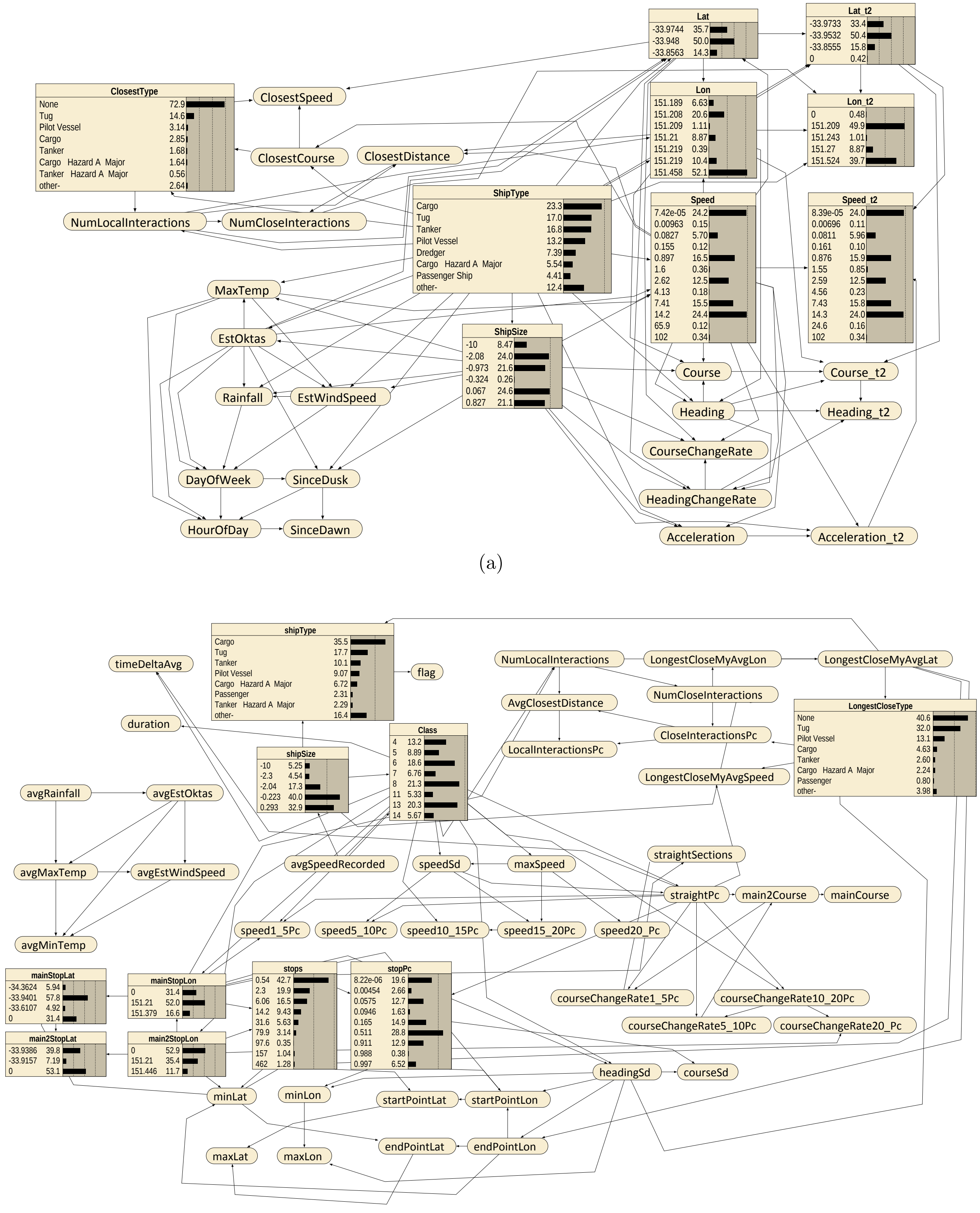
Exploring Bayesian networks is very easy and natural and here turned up many points of interest. In confirming the reasonableness of the time series model, we found that entering 'Tug' or 'Pilot Vessel' into the 'ShipType' variable significantly increases the chance of another vessel being nearby. Cargo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

3.2 ANOMALY SCORES

There is no generally accepted method for detecting anomalies from BN models. Jensen and Nielsen (2007) proposed a "conflict measure" to detect possible incoherence in evidence $\mathbf{E} = \{E_1 = e_1, \ldots, E_m = e_m\}$:

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

Jensen and Nielsen use this to identify when a power plant begins behaving abnormally. Unfortunately, this will only catch cases where each attribute is independently common but jointly uncommon. Here, we're interested in any kind of joint uncommonness, even when variables are independently uncommon, which simply comes down to a difference in requirements. In other approaches Loy et al. (2010) used learned DBNs to calculate log-likelihoods and compare them against thresholds selected to maximize the accuracy, extended to detecting abnormal correlations between multiple objects. Cansado and Soto (2008) simply assumed that records with low probabilities given the learned BN are anomalies.



(a)

(b)

Figure 3: Example BNs produced by CaMML for the (a) time series data and (b) track summary data.

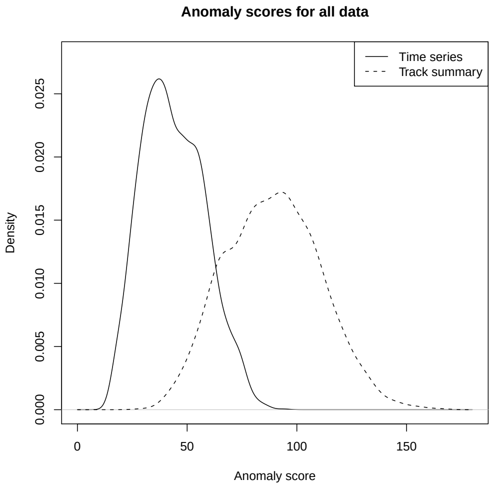


Figure 4: The KDE distributions of anomaly scores for all tracks in the data set according to the (a) time series and (b) track summary networks.

We started from the same assumption as Cansado and Soto, however 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 present the probability itself to surveillance operators, albeit in a more convenient form.

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 the order of 1^{-10}) 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 instead 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. Johansson and Falkman (2007), 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 Figure 4. 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. There is a surprisingly small correlation between the two sets of scores (r =0.159; $p < 0.001$).[2] The two models look at different aspects of each track, and, as we see below, reinforce each other when performing anomaly detection.

3.3 RESULTS ON ANOMALOUS DATA

Unfortunately, we did not have any access to known anomalous tracks nor are there any standardised or publicly available vessel track data sets containing anomalies (or otherwise). Nevertheless, there are many ways to create anomalous data. Cansado and Soto (2008) generated anomalies by modifying selected attributes to random values within their ranges. Johansson and Falkman (2007) generated anomalous data using anomalous models.[3] 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.

3.3.1 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. Figure 5(a) shows how this affected the anomaly score. In most cases this false ship effect 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. Sometimes, however, 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. A closer look at these showed that many are highly improbable (i.e., have high anomaly scores), suggesting that either they have been mislabelled or, more intriguingly, that they do indeed behave anomalously according to their type. This suggests a new criterion for anomalousness based not merely upon the probability of the given track but on what alterations might explain the track better. This has some of the spirit of Jensen and Nielsen's conflict measure, though is clearly quite different; we leave this possibility for future exploration.

Figures 5(b) and 5(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

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

[3] Wang et al. (2008), without known anomalous data, simply weakened their threshold to find "anomalies", whether they were there are not!

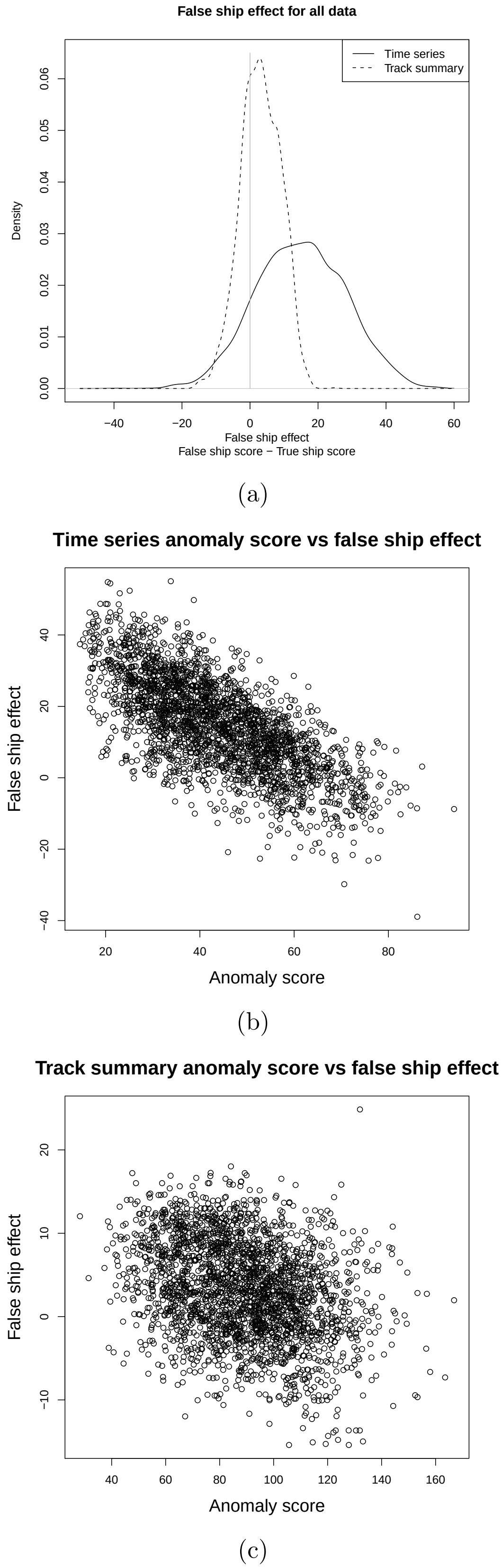


Figure 5: False ship effect: (a) anomaly score differences for false ship information versus correct ship information, sorted by score; and scatter plots for (b) time series and (c) track summary networks.

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, $p \ll 0.01$).

3.3.2 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.

3.3.3 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 Figure 6 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

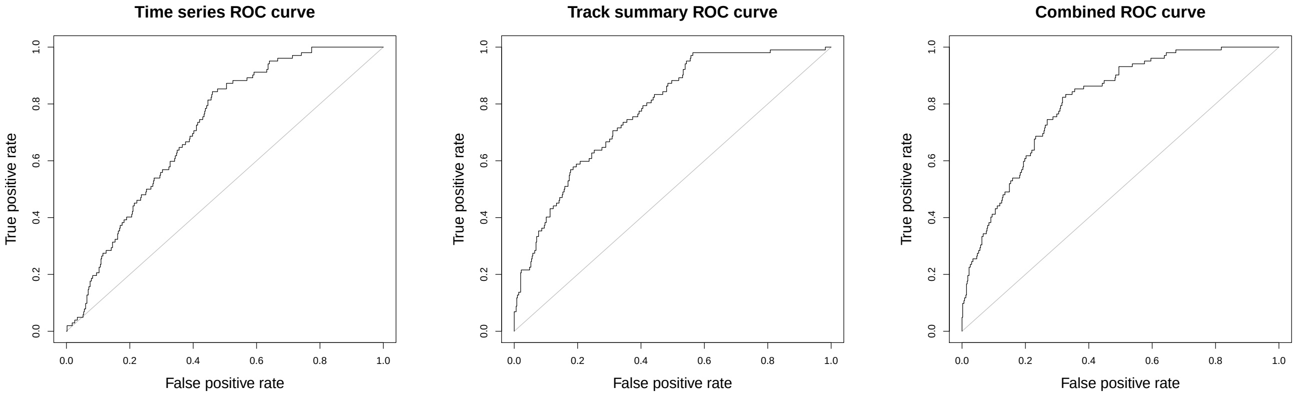


Figure 6: ROC curves for test data, containing both normal tracks and manually created anomalous tracks, given the (left) time series, (middle) track summary and (right) combined models.

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 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.

3.3.4 Testing on Johansson & Falkman's simulated data

We also applied our methods to the simulated data used by Johansson and Falkman (2007), both normal and anomalous. Our models, while not well suited to the simulated data, 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. When we calculated the ROC curves, we found that the time series 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 (whose ROC curve is shown in Figure 7) again performs better than both individually, with an AUC of 0.727.

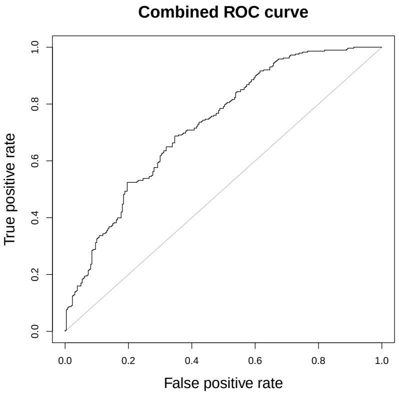


Figure 7: ROC curves for the Johansson and Falkman data using the combined models.

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]

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 their assessments. This sug-

[4] For further details of our comparison with Johansson & Falkman's work, see Mascaro et al. (2010).

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

Type	Track Summary Score	Delta	Time Series 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	126.9	+36.1	53.9	+8.2

gests that learning networks at still additional time scales, intermediate between the full track and each AIS snapshot, may improve anomaly detection even further. Such approaches may well generalize to other kinds of anomaly detection and can be extended to work with other kinds of track, such as those created by cars, planes and humans.

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

Bureau of Meteorology (2010). Daily weather observations, May – July 2009. `http://www.bom.gov.au/climate/dwo/IDCJDW2124.latest.shtml`.

Cansado, A. and A. Soto (2008). Unsupervised anomaly detection in large databases using Bayesian networks. *Applied Artificial Intelligence 22*(4), 309 – 330.

Cheeseman, P., J. Stutz, M. Self, J. Kelly, W. Taylor, and D. Freeman (1988, August). Bayesian classification. In *AAAI 88*, pp. 607–611.

Das, K. and J. Schneider (2007). Detecting anomalous records in categorical datasets. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 220–229. ACM.

Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In M. Jordan (Ed.), *Learning in Graphical Models*, pp. 301–354. MIT.

Helldin, T. and M. Riveiro (2009). Explanation methods for Bayesian networks: Review and application to a maritime scenario. In *Proc. of the 3rd Annual Skövde Workshop on Information Fusion Topics (SWIFT 2009)*, pp. 11–16.

Jensen, F. V. and T. D. Nielsen (2007). *Bayesian networks and decision graphs* (2nd ed.). New York: Springer Verlag.

Johansson, F. and G. Falkman (2007). Detection of vessel anomalies — a Bayesian network approach. In *Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 395–400.

Korb, K. B. and A. E. Nicholson (2010). *Bayesian Artificial Intelligence* (2nd ed.). Chapman & Hall/CRC Press.

Laxhammar, R. (2008). Anomaly detection for sea surveillance. In *The 11th Int. Conf. on Information Fusion*, pp. 55–62.

Li, X., J. Han, and S. Kim (2006). Motion-Alert: Automatic anomaly detection in massive moving objects. In *Proc. of the 2006 IEEE Intelligence and Security Informatics Conference (ISI 2006)*, Berlin, pp. 166–177. Springer.

Loy, C., T. Xiang, and S. Gong (2010). Detecting and discriminating behavioural anomalies. *Pattern Recognition*.

Mascaro, S., K. B. Korb, and A. E. Nicholson (2010). Learning normal vessel behaviour from AIS data with Bayesian networks at two time scales. Technical Report TR 2010/, Bayesian Intelligence.

ODonnell, R., A. Nicholson, B. Han, K. Korb, M. Alam, and L. Hope (2006). Causal discovery with prior information. In A. Sattar and B.-H. Kang (Eds.), *AI 2006*, Volume 4304 of *LNCS*, pp. 1162–1167. Berlin: Springer.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann.

Spirtes, P., C. Glymour, and R. Scheines (1993). *Causation, Prediction and Search*. Springer Verlag.

Wallace, C. S. and P. R. Freeman (1992). Single factor estimation by MML. *Journal of the Royal Statistical Society B 54*(1), 195–209.

Wallace, C. S. and K. B. Korb (1999). Learning linear causal models by MML sampling. In A. Gammerman (Ed.), *Causal Models and Intelligent Data Management*, pp. 89–111. Heidelberg: Springer-Verlag.

Wang, X., J. Lizier, O. Obst, M. Prokopenko, and P. Wang (2008). Spatiotemporal anomaly detection in gas monitoring sensor networks. *Wireless Sensor Networks*, 90–105.

Wong, W., A. Moore, G. Cooper, and M. Wagner (2003). Bayesian network anomaly pattern detection for disease outbreaks. In *Int. Conf. on Machine Learning*, Volume 20, pp. 808.