Abstract—The paper is devoted to statistical analysis of vessel motion patterns in the ports and waterways using AIS ship self-reporting data. From the real historic AIS data we extract motion patterns which are then used to construct the corresponding motion anomaly detectors. This is carried out in the framework of adaptive kernel density estimation. The anomaly detector is then sequentially applied to the real incoming AIS data for the purpose of anomaly detection. Under the null hypothesis (no anomaly), using the historic motion pattern data, we predict the motion of vessels using the Gaussian sum tracking filter.

Keywords: Maritime surveillance, Automatic identification system, motion patterns, novelty detection, motion prediction, kernel density estimation.

I. INTRODUCTION

Maritime surveillance of ports and coastlines is of paramount importance for island nations such as Australia. The risks historically include unauthorised maritime arrivals, prohibited imports/exports, maritime pollution, piracy and more recently maritime terrorism. Considering that more than 99% of imports and exports to Australia are carried by sea, potential terrorist attacks in ports or waterways could cause serious disruptions with significant economic implications.

The typical sensors used for maritime surveillance of ports and waterways include radars, infrared and video cameras, installed on fixed ground locations or mounted on border patrol vessels, aircraft and satellites. Recently, however, a number of self-reporting maritime systems have been introduced, mainly for the purpose of safety in navigation and collision avoidance. The messages transmitted by these self-reporting systems have thus become an abundant and inexpensive source of information for maritime surveillance. Due to its nature, however, this source of data is also unreliable and incomplete: on one hand it is fairly easy to spoof the self-reporting broadcasts and on the other hand, self-reporting is likely to be turned off during the illegal operations (Although the absence of self-reporting data over a long period of time and its subsequent reappearance is easy to detect and raise the alarm). Despite its drawbacks, the self-reporting data promises to be a useful supplement to the existing maritime surveillance sensors. The task of data fusion research is to determine what is the best way to exploit this massive amount of broadcasted information, in order to improve the situational awareness of the maritime domain.

The most important self-reporting maritime system is the Automatic Identification System (AIS), which has been recently made compulsory by the International convention for the Safety of life and sea (SOLAS) for most commercial ships (cargo, passenger, tankers, tugboats, etc). AIS messages are automatically broadcasted with a reporting frequency directly proportional to the speed of the vessel. There are various types of AIS messages which are broadly classified as static information (name, type, size, etc. of vessel) and dynamic (position in geodetic coordinates, speed, course, heading, destination, estimated time of arrival, etc). Not all of the listed pieces of information are compulsory.

Recently there has been an increased interest in maritime surveillance. A survey of NURC research activities for maritime surveillance, based on AIS, coastal radars, SAR imagery, is presented in [1]. In a series of papers [2], [3], Rhodes, Bomberger et al describe a system that learns the normal behavior of vessels, detects anomalies and predicts the motion using an artificial neural network trained with AIS data. Tun et al [4] developed an algorithm based on density maps that breaks the vessel motion paths collected by an AIS receiver into separate regions. The routes of ships are used as input to an HMM to learn the motion patterns. The subject of motion behaviour analysis, however, is not unique to maritime surveillance – it has initially been studied in computer vision in the context of traffic monitoring [5], [6], human activity monitoring [7] and unusual event detection [8].

This paper is devoted to statistical analysis of AIS data for the purpose of detecting anomalies in vessel motion and the prediction of vessel motion under the assumption of normal behaviour. The proposed algorithms for motion anomaly detection and motion prediction assume that relevant motion patterns have already been extracted from the historic (training) data during the data mining stage of data processing. Anomaly detection is carried out in the framework of adaptive kernel density estimation (KDE), and is applied sequentially to the new incoming AIS data. The anomaly detection threshold is related to the probability of false alarm. Under the null hypothesis (normal motion behaviour) and using the historic motion pattern AIS data, this paper proposes a simple algorithm for vessel motion prediction (location, velocity) within
the specified time window.

The paper is organised as follows. Sec. II describes the problem of statistical analysis of motion patterns. Sec. III presents the algorithm for detection of anomalies, based on adaptive kernel density estimation.

II. PROBLEM DESCRIPTION

The historic (training) AIS data sets are of growing size and complexity. Typically the first step in the analysis of huge size data sets is to perform data mining in order to extract motion patterns. As an illustration of the importance of this step, Fig.1.a displays the ensemble of trajectories of all vessels coming in and going out of Port Adelaide (Gulf St. Vincent), over a period of three months. The figure is rather messy and before we can apply any of the algorithms (to be presented in the paper), it will be assumed that AIS data has been processed and patterns have been extracted. An example of a simple pattern is shown in Fig.1.b, which represents all paths originating from a dock in the Port of Adelaide (at latitude $138^\circ30'23''$, longitude $-34^\circ49'48''$), initially moving north and then turning southwest (destination Edithburgh, SA).

Next we try to define a motion pattern. This is not obvious, and can be quite complex if one deals with a network of origins/destinations with multiple connecting paths. See for example Fig. 2, which shows graphically a network of vessel paths in a harbour, where node 1 may for example be the entry or the exit point of a harbour, while nodes 2, 3 and 4 are the docking stations in the harbour. In order to simplify matters, we will define a motion pattern by kinematic and attribute information, with only one compulsory attribute - its origin. Other useful attributes, if available, can be the vessel type, season of the year, etc. The kinematic information will include the ship location (in two-dimensions) and velocity (also in two-dimensions). The origin of a motion pattern is defined by the location-velocity vector and its associated uncertainty ellipsoid. In Sec.IV we will argue that it is also very useful for a pattern to contain the elapsed time information in the form of the interval of time since the vessel was at the origin of the pattern. By adopting the described framework for a motion pattern, we will not need to worry about the topology of the network of paths (see Fig. 2).

Consider again Fig. 2. For such a network of vessel paths one defines the set of motion pattern origins, in this example $\mathbb{O} = \{1, 2, 3, 4\}$. Let a trajectory belonging to the motion pattern $P_j$ with origin $j \in \mathbb{O}$ be denoted as

$$X_j(t) = \{x_j^i(t); t = t_1, t_2, \ldots, t_{N_j}\}$$

(1)

where $i = 1, 2, \ldots, N_j$ is the index of the trajectory ($N_j$ is the total number of training trajectories from $P_j$) and $x_j^i(t)$ is the kinematic state of the vessel traversing trajectory $X_j$ at time $t$. The kinematic state consists of positional information $(\zeta, \eta)$ and velocity information $(\dot{\zeta}, \dot{\eta})$, i.e. $x_j^i(t)$ is a four-dimensional vector $[\zeta \ \eta \ \dot{\zeta} \ \dot{\eta}]^T$. We will assume that kinematic states $x_j^i(t)$ are independent in $t$ and along $i$ (Note that if $x_j^i(t)$ are the outputs of a tracking filter, then they are correlated in time. Hence we will use raw AIS data).

The motion pattern displayed in Fig.1.b will be referred to as being simple, because all vessels originating from the docking station at lat $138^\circ30'23''$, lon $-34^\circ49'48''$ have the same destination.

Using all available trajectories from motion pattern $P_j$ (which serve as the training data set for normal or usual behaviour), the paper deals with:

- The problem of motion anomaly detection, where we need to determine sequentially (on-line) if the state-vectors of a test trajectory $y_j(t) \in Y_j$ comply with the normal behaviour (normalcy is the null hypothesis);
- The problem of motion prediction, where having established that the state of a test vessel at time $t$, $y_j(t)$, complies with normalcy, we need to predict the state of this vessel at time $t + T$, ($T > 0$), under the assumption that it will continue to follow the pattern of normal behaviour.

III. DETECTION OF ANOMALY IN MOTION

In order to detect an anomaly in vessel motion we will use the training data from pattern $P_j$ to determine a detection
threshold that will partition the state space into two regions: one region to correspond to hypothesis \( H_0 \) (normal behaviour), the other to \( H_1 \) (anomaly). This threshold will be applied sequentially in time to the incoming test data \( y_j(t) \). In the classification literature similar problems have been referred to as one-class classification or novelty detection [9, Ch.8], [10], and solved using support vector machines (SVM). In the following we will perform anomaly detection using the adaptive kernel density estimator (also known as the Parzen window method).

In order to simplify notation we will introduce index \( k \) to enumerate pairs \((i, t)\) in (1). The set of unlabeled training data corresponding to motion pattern \( j \in \mathcal{O} \) is then:

\[
\mathcal{X}_j = \{ x_{j,k}; k = 1, \ldots, K_j \}
\]

(2)

where \( K_j = \sum_{i=1}^{N_j} T_i \). \( \mathcal{X}_j \) is an iid random sample from the underlying multi-variate pattern pdf under the normalcy hypothesis \( H_0 \): \( p_j(x|H_0) = p_j(\zeta, \eta, \zeta, \eta|H_0) \). For motion anomaly detection we need first to approximate \( g_j(x) = p_j(x|H_0) \); this will be carried out using the adaptive KDE approximation [11].

### A. Kernel density estimation

The KDE approximation constructs the density \( g_j(x) \) by placing a kernel function \( \phi \) on every observed datum \( x_{j,k} \). The kernel is parametrised by its width \( h \), which can be either fixed (identical for all observed data) or adaptive. For simplicity we will drop index \( j \) from notation in the reminder of the section. Assuming \( x \in \mathbb{R}^d \), (in our case \( d = 4 \)) the fixed KDE approximation is given by:

\[
g(x) \approx \hat{g}(x) = \frac{1}{K h^d} \sum_{k=1}^{K} \phi \left( \frac{x - x_k}{h} \right)
\]

(3)

The kernel must satisfy \( \phi(x) \geq 0 \) and \( \int_{\mathbb{R}^d} \phi(x)dx = 1 \). Adopting the Gaussian kernel with zero-mean and the covariance matrix \( \Sigma \):

\[
\phi(x) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}} \exp \left\{-\frac{1}{2} x^T \Sigma^{-1} x \right\}
\]

(4)

we have:

\[
\hat{g}(x) = \frac{1}{K h^d (2\pi)^{d/2} \sqrt{|\Sigma|}} \sum_{k=1}^{K} e^{-\frac{(x - x_k)^T \Sigma^{-1} (x - x_k)}{2 h^2}}.
\]

(5)

The optimal fixed bandwidth (under the assumption that the underlying pdf is Gaussian) for the Gaussian kernel is computed as [11]:

\[
h^* = AK^{-\frac{1}{d+4}}
\]

(6)

where

\[
A = \left[4/(d+2)\right]^{1/(d+4)}.
\]

(7)

Covariance \( \Sigma \) needs to be estimated from the data as the sample covariance.

The fixed KDE is unable to deal satisfactorily with the tails of distributions: since the observed data in the tails are rare, the window widths in the tails need to be broader. An obvious problem is deciding whether or not an observation is in a region of low density. The adaptive KDE approach [11, p.101] copes with this by a two-stage procedure, where the first stage is typically the fixed KDE. The second stage initially computes adaptive window widths \( h_k \), \( k = 1, \ldots, K \) as \( h_k = h^* \lambda_k \) where \( h^* \) is given in (6) and

\[
\lambda_k = \left( \frac{\hat{g}(x_k)}{\ell} \right)^{-\gamma}
\]

(8)

where \( \ell \) is the geometric mean of a sequence \( \{ \hat{g}(x_k) \} \) that is:

\[
\log \ell = \frac{1}{K} \sum_{k=1}^{K} \log \hat{g}(x_k),
\]

(9)

and \( \gamma \) is the sensitivity parameter such that \( 0 \leq \gamma \leq 1 \), typically set to 0.5.

Finally the adaptive KDE approximation is similar to (3), except that for each datum \( x_k \) we apply the kernel of width \( \tilde{h}_k \):

\[
\hat{g}(x) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{(h_k)^d} \phi \left( \frac{x - x_k}{h_k} \right)
\]

(10)

### B. Anomaly detection

Computation of the decision boundary in the state space is prohibitively expensive and therefore we propose to perform the detection using the values of density \( g(x) = p(x|H_0) \). Let \( y \) be a test datum from a test trajectory \( Y \) which originates from the same node as the training data (node \( j \), subscript being supressed). Then anomaly (i.e. hypothesis \( H_1 \)) is declared if

\[
g(y) > \alpha g(x_r),
\]

(11)

where

\[
r = \arg \min_k g(x_k)
\]

(12)

and \( \alpha > 0 \) is a detector parameter which will be related to the probability of false detection. Note that if we want all training data \( \{ x_k; k = 1, \ldots, K \} \) to fall inside the anomaly detection boundary, then \( \alpha < 1 \).

We select \( \alpha \) so that

\[
Pr\{g(y) < \alpha g(x_r)|H_0\} = P_{fa}
\]

(13)
where \( P_{fa} \) is a specified probability of false alarm (incorrect anomaly detection). Eq. (13) can be written as:

\[
\int \chi_{[0,\alpha g(x_m)]}(g(y))g(y)dy = P_{fa},
\]

where \( \chi_A(z) \) is an index function defined as

\[
\chi_A(z) = \begin{cases}
1, & z \in A \\
0, & \text{otherwise}.
\end{cases}
\]

For a given \( P_{fa} \), we can compute \( \alpha \) numerically. First we generate a random iid sample \( \{y_m \sim g(y) : m = 1, \ldots, M\} \). Substitution of approximation \( g(y) \approx \frac{1}{M} \sum_{m=1}^{M} \delta(y - y_m) \) into (14) yields:

\[
P_{fa}(\alpha) \approx \frac{1}{M} \sum_{m=1}^{M} \chi_{[0,\alpha g(x_m)]}(g(y_m)).
\]

We expect that \( P_{fa}(\alpha) \) will depend on the number of training data points \( K \).

C. Numerical results

First we test the anomaly detector using simulated data. For this purpose we generate 100 vessel trajectories (in \( \zeta, \eta, \zeta, \eta \) space) originating from node \( j = 1 \), as shown in Fig.3. The motion pattern is complex, because there are multiple (three) destinations (they are equally likely in this example). The total number of training data points \( x_k \sim p(x|H_0) \) is \( K = 3726 \). The kernel covariance matrix was set to \( \Sigma = \text{diag} \left[ \sigma_p^2, \sigma_p^2, \sigma_p^2, \sigma_r^2 \right] \), with \( \sigma_p = 1400 \) m and \( \sigma_r = 2.3 \) m/s. The resulting anomaly detection boundary is shown in \( (\zeta, \eta) \) plane in Fig.3.a for \( \alpha = 0.8 \) (brown line). Note however that for anomaly detection, according to (11), we do not need to compute the boundary in the state space (which is a computationally very intensive operation), we only compute the value of \( g(x_r) \), where \( r \) was defined by (12). The decision boundary in Fig.3.a is shown only for the sake of illustration.

The test trajectory for anomaly detection is shown in Fig.3.a with green and red dots. The vessel following this trajectory enters the surveillance region at node 1 and is initially moving eastwards in accordance with the motion pattern (towards either node 3 or 4) for about 1 hour and 20 minutes. The anomaly detector correctly classifies the motion as \( H_0 \) during this period of time (see Fig.3.b); the segment of the test trajectory corresponding to \( H_0 \) is indicated by the green dots in Fig.3.a. The first time when the anomaly is declared, the vessel position is actually inside the decision boundary for the positional information, but its heading (velocity vector) is incompatible with the training data (at this stage the vessel is heading north-west). From then on, the vessel is making a turn, joins the path from node 2 to 1 and continues to travel westwards. The anomaly detector is always deciding \( H_1 \) in this segment, either due to velocity or positional incompatibility with the training data. The test trajectory when \( H_1 \) is declared is indicated using red dots in Fig.3.a. This example illustrates the importance of velocities in the definition of a motion pattern. We emphasize again that there is no need to keep the time information in a motion pattern.

Next we compute (in the context of the described simulated data framework) the probability of false alarm as a function of the threshold parameter \( \alpha \) and the number of training points \( K \). Fig.4 shows the result obtained with \( K = 4440 \) (red line) and \( K = 9560 \) (blue line) training points. The size of the random sample \( y_m \) from \( g(y) \) in (16) was set to \( M = 80000 \). Covariance matrix \( \Sigma \) was specified above. We observe that for a larger training set (larger \( K \)), we make a better approximation of \( g(y) \) and hence can use a higher threshold value \( \alpha \) for the fixed value of \( P_{fa} \). This can be explained as follows. For larger \( K \), the value of \( g(x_r) \) is likely to be smaller; in order to obtain the fixed value of \( P_{fa} \) (which corresponds to the area under \( g(y) \) s.t. \( g(y) < \alpha g(x_r) \)), when \( g(x_r) \) is smaller, \( \alpha \) needs to be greater.

The last two numerical examples in this section deals with real AIS data. The first set of data was collected in Golf St Vincent (Port Adelaide). Fig.5 shows the results of the anomaly detector with \( \alpha \) parameter set to 0.6, applied to the motion pattern data displayed previously in Fig.1.b. As before, in Fig.5 we only show the 2D positional data (decision boundary, training data, and the test trajectory), although the actual state-space is four-dimensional. Fig.5.a demonstrates a trajectory of a vessel which comes from the same motion pattern family as the training data. The motion of vessel in
where p(i.e. y density with mean likelihood function. For the initial pdf we adopt the Gaussian p we need to specify: (1) the pdf of the initial vessel state, design a particle filter (PF) for this nonlinear filtering problem, j filtering. Let us drop subscript velocity motion model: is given by the Gaussian density under the nearly constant y probable state of vector the adopted framework we can usually only predict the most p we want to predict the state y A. Using training data X Suppose we have at our disposal only the unlabeled training data X, as in (2). Having decided for the state of a test vessel at time t0, y(t0) that hypothesis H0 holds, we want to predict the state y(t0 + T), where T > 0. Ideally we would like to predict the pdf p(y(t0 + T)), however, in the adopted framework we can usually only predict the most probable state of vector y(t0 + T).

There are several possible solutions all based on nonlinear filtering. Let us drop subscript j from notation. In order to design a particle filter (PF) for this nonlinear filtering problem, we need to specify: (1) the pdf of the initial vessel state, p(x0); (2) the motion transition model; (3) the measurement likelihood function. For the initial pdf we adopt the Gaussian density with mean y(t0) and a very small covariance P0, i.e. p(x0) = N(x0; y(t0), P0). The motion transition model is given by the Gaussian density under the nearly constant velocity motion model:

\[ p(x_{n+1}|x_n) = \mathcal{N}(Fx_n, Q) \]  

where n is a discrete-time index which indicates time t = t0 + n\Delta, (\Delta is a conveniently chosen time increment); transition matrix F is given by

\[
F = \begin{pmatrix}
1 & 0 & \Delta & 0 \\
0 & 1 & 0 & \Delta \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

and Q is the process covariance matrix.

Since we do not actually collect the measurements (we are predicting the future vessel state), instead of the measurement likelihood we will use the KDE approximation \(\hat{y}(x)\) of (10), which summarizes the historic measurements. Let us denote the particles at discrete-time n by \(\{x^n_i\}_{i=1}^{N_p}\), \(N_p\) being the number of particles. The steps of a single cycle of the PF are summaries in Table I. The resampling and regularisation steps are explained in, for example, [12]. The PF runs until \(n\Delta\) approximately equals the required prediction interval T.

The described PF was applied to the simulated motion pattern shown (in 2D) in Fig.3. The initial state vector was \(y_1(t_0) = [1.8 \text{ km} \ 21 \text{km} \ 6.4 \text{m/s} \ 0.4 \text{m/s}]^T\), \(\Delta = 1 \text{ min}, N_p = 1000\). The PF predicted vessel state is shown in Fig.7 in 2D plane (velocities not shown), after T = 10, 32 and 70 minutes. An obvious problem with the PF predicted state is
that it fails to estimate the predicted pdf \( p(y_j(t_0 + T)) \) after the motion pattern paths branch. This pdf after 60 minutes should be bi-modal, with one (smaller) bump along the path towards node 2 and another (larger) along the path towards nodes 3 and 4. The PF captures only the second bump; the particles tend to follow the most probable path. 

Improvements to the proposed PF can be envisaged, involving some kind of managing multiple branching hypotheses (for multiple paths). This however appears to be unnecessary, as we discuss next.

### B. Indexed training data \( X_j \)

The main cause of problems with the above approach is that it is using unlabeled training data \( X_j \) from pattern \( P_j \), as in (2). If we instead index each data point \( x_k \) by (1) its trajectory of origin \( i = 1, 2, \ldots, N_j \) and (2) its sequence index \( \ell = 1, 2, \ldots, T_i \), then the problem of motion prediction becomes much easier. In this notation, \( x_k^j(\ell) \) is an AIS data point (state vector at discrete time \( t_\ell \)) from trajectory \( X_j^i \) of pattern \( j \), see eq(1). The pattern training data set is \( X_j = \bigcup_{i=1}^{N_j} X_j^i \).

The predicted state pdf \( p(y_j(t_0 + T)) \) can be now approximated as follows:

**Step 1.** Compute the contribution of all training data from \( x_k \in X_j \) to adaptive KDE approximation at the initial test data point \( y_j(t_0) \). This follows directly from (10). The contribution of \( x_k \), referred to as its weight \( q_k \), is then

\[
q_k = \frac{1}{K(h_k)} \phi \left( \frac{x - x_k}{h_k} \right)
\]

Let \( q^* = \max_k q_k \).

**Step 2.** Extract a subset \( X_j^* \subset X_j \) as:

\[
X_j^* = \{ x_k \in X_j | q_k > \beta q^* \}
\]

where \( \beta < 1 \).

**Step 3.** \( \forall x_k \in X_j^* \) find trajectory index \( i \) and time index \( \ell \). Then using trajectory \( X_j^i \), estimate the state of a vessel that is following this trajectory, at the prediction time \( t_0 + T \).

The resulting random sample can be seen as an approximation of the required density \( p(y_j(t_0 + T)) \). Fig.8 shows the result (positional 2D information only) after \( T = 10, 32 \) and 70 minutes, using the same training data and parameters that were used in Sec.IV-A. Parameter \( \beta \) was set to \( 10^{-5} \). In order to improve the estimate of the pdf (and enhance the visual effect), the obtained random samples are resampled (using weights \( q_k \)) and regularised. We observe from Fig.8 that for \( T = 32 \) min, the predicted pdf branches, thus indicating two possible future vessel paths. After \( T = 70 \) min, the pdf is distinctly bi-modal.

### V. Summary

The paper presented some preliminary results of ongoing research into the behaviour analysis of vessels in ports and waterways using AIS broadcasts (both as training and testing
Fig. 8. The predicted vessel state (position shown only) after 10, 32 and 60 minutes (method 2)

data). A simple and fast detector of anomaly in vessel motion is proposed based on the adaptive kernel density estimation. The probability of false alarm of this detector can be evaluated numerically, providing thus its quantitative measure of performance. Vessel motion prediction attempts to compute the density of the vessel state in the future. Using effectively the training data, the paper presents a fairly straightforward solution to motion prediction.

REFERENCES


