Quantitative Assessment of Anomaly Detection Algorithms in Annotated Datasets from the Maritime Domain

Mathias Anneken, Yvonne Fischer and Jürgen Beyerer

Abstract The early detection of anomalies is an important part of a support system to aid human operators in surveillance tasks. Normally, such an operator is confronted with the overwhelming task to identify important events in a huge amount of incoming data. In order to strengthen their situation awareness, the human decision maker needs an support system, to focus on the most important events. Therefore, the detection of anomalies especially in the maritime domain is investigated in this work. An anomaly is a deviation from the normal behavior shown by the majority of actors in the investigated environment. Thus, algorithms to detect these deviations are analyzed and compared with each other by using different metrics. The two algorithms used in the evaluation are the Kernel Density Estimation and the Gaussian Mixture Model. Compared to other works in this domain, the dataset used in the evaluation is annotated and non-simulative.

1 Introduction

In order to be able to make the best decision, an operator in surveillance tasks needs an overview about all incoming data. Therefore, only if operators are able to understand and interpret the whole data correctly, they will be able to make the best possible next move. As stated by Fischer and Beyerer [7], the main problem is not the acquisition of the data, but the large amount of data. In order to aid human decision makers, a support system is needed. This system needs to provide the aid to identify important

M. Anneken (✉) · Y. Fischer · J. Beyerer
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), Karlsruhe, Germany
e-mail: mathias.anneken@iosb.fraunhofer.de

Y. Fischer
e-mail: yvonne.fischer@iosb.fraunhofer.de

J. Beyerer
Vision and Fusion Laboratory, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
e-mail: juergen.beyerer@iosb.fraunhofer.de

Y. Bi et al. (eds.), *Intelligent Systems and Applications*,
Studies in Computational Intelligence 650, DOI 10.1007/978-3-319-33386-1_5

events. Without such a system, the operator needs to analyze the whole data by hand. This is tedious work and might result in a reduced concentration of operators. Hence, the chance is high to overlook crucial events.

The crucial events are most often deviations from the normal behavior. Therefore, this events can be classified as anomalies. If a support system is able to provide an operator reliable with information about anomalies, this will increase the situation awareness of the operator. Thus, algorithms to identify anomalies are evaluated in this work. The idea is to reduce the workload of an operator by providing information about the important events. While this reduction helps an operator to concentrate on the important tasks, it is crucial for the acceptance, that the algorithms will identify the anomalies as good as possible. In order to be able to assess the performance of an algorithm in real life, the algorithm has to be evaluated with real data. Else, the support system might not be able to detect some anomalies, which are not covered in a simulated dataset.

Here, the detection of anomalies is investigated in the maritime domain. In this special case, the normal traffic drives on sea lanes. Anomalies can, e.g., be seen as the deviation from these lanes or a different speed compared to the normal traffic. Thus, the algorithms in the evaluation must be able to assess spatio-temporal data in form of trajectories.

2 Related Work

Chandola et al. [6] give a wide overview about different tasks and algorithms used to detect anomalies. The applications range from sensor network and cyber security to fraud detection and image processing. Morris and Trivedi [13] give a survey especially for anomaly detection using vision-based algorithms. For each application appropriate algorithms are introduced. The underlying concepts for the algorithms vary, depending on the use case, e.g., classification-, clustering- or nearest neighbor-based algorithms are used. Each of these algorithms has its advantages and disadvantages.

Especially in the maritime domain, the detection of anomalies is an important field of research. Several different approaches were introduced to identify abnormal behavior of vessels and to incorporate expert knowledge to correctly assess specific situations.

Laxhammar et al. [11] compare the Kernel Density Estimation (KDE) and the Gaussian Mixture Model (GMM). The models are trained with a real life dataset comprising the *position*, *heading* and *speed* of each vessel. For the evaluation, artificial anomalies are simulated. The models' performances to resemble the normal behavior is evaluated by comparing the log-likelihood with the 1st percentile as well as the median log-likelihood. For the anomaly detection performance, the needed number of observations for detecting an anomaly is compared. Anneken et al. [2] evaluate the same algorithms by using an annotated dataset.

Brax and Niklasson [5] introduce a state-based anomaly detection algorithm. The different discrete states are *heading*, *speed*, *position* and *relative position to the next vessel*. Different roles (here called agents) are developed and incorporate the states. These roles comprise, e.g., smuggler and raid agents. The probability for each role is calculated using the prior defined states. For the evaluation, different scenarios resembling specific situations are generated to obtain an accurate ground truth. The algorithms are only tested using this simulated ground truth.

Andersson and Johansson [1] use an algorithm based on a Hidden Markov Model (HMM) to detect abnormal behavior. They train the HMM with simulated normal behavior of ships in a certain area. For the evaluation, the data is divided into discrete states. The states resemble the change of specific values, i.e., *distance to other objects*, *vessel size*, *identification number*, *speed* and *heading*. Afterwards, the model is evaluated by using a simulated pirate attack.

Laxhammer and Falkman [10] introduce the sequential conformal anomaly detection. The underlying conformal prediction framework is, e.g., explained further by Schafer and Vovk [16]. The algorithm provides a reliable lower boundary for the probability that a prediction is correct. This threshold is given as a parameter for the algorithm and directly influences the false positive rate. The similarity between two trajectories is calculated by using the Hausdorff distance. The model is trained with real life data, but the anomalies for the evaluation are simulated.

De Vries and van Someren [17] use *piecewise linear segmentation* methods to partition trajectories of maritime vessels. The resulting trajectories are grouped in clusters. Afterwards, the anomaly detection is performed by using kernel methods. Additionally, expert domain knowledge like geographical information about harbors, shipping lanes and anchoring areas is incorporated. The algorithms are validated with a dataset from the Netherlands' coast near Rotterdam.

Guillarme and Lerouvreur [9] introduce an algorithm consisting of three main steps. They first partition the training trajectories into stops and moves segments using the *Clustering-Based Stops and Moves of Trajectories* algorithm. Afterwards, a similarity measure and a clustering algorithm based on the density of the data is used to cluster the resulting sub-trajectories. The clusters discovered by the algorithm need to be assessed by hand. With this results, motion patterns and junctions for the trajectories are defined. For the evaluation, satellite AIS data is used. No information about the performance of the algorithm compared to other algorithms is given.

Fischer et al. [8] present an approach based on dynamic Bayesian networks. Different situations and their relationship with each other are modeled in a situational dependency network. With this network, the existence probability for each defined situation, e.g., *a suspicious incoming smuggling vessel*, can be estimated. This estimated probability is used to detect unusual behavior. The algorithm is tested with simulated data.

Anneken et al. [3] reduce the complexity of trajectories by using b-splines estimation. The control points of the b-splines are used as the feature vector and the normal model is trained by using different machine learning algorithms. For the evaluation an annotated dataset is used. The results of the different algorithms are compared with the results of a KDE and a GMM as previously shown in [2, 11].

The majority of the previous work in the maritime domain uses simulated data to evaluate their proposed algorithms. E.g., the evaluations in [1, 8] rely entirely on simulated data, and in [10, 11] the anomalies are created artificially. In this work, the same annotated dataset as in [2, 3] is used. Additionally to the previously used areas, an additional area is introduced and the algorithms are compared by using a different set of metrics.

3 Dataset

The dataset for the evaluation was recorded by using the automatic identification system (AIS). The AIS provides different kinds of data like *navigation-status*, *estimated time of arrival*, ..., *destination*. For the analysis only a subset of the whole available data is used, namely *position*, *speed*, *heading*, *maritime mobile service identity (MMSI)*, *timestamp* and *vessel-type*. The whole dataset comprises more than 2.4 million unique measurements recorded during a time span of seven days.

A depiction of the dataset in form of a heat-map is shown in Fig. 1. The map encodes the traffic density with colors ranging from green for low density to red for high density. Thus sea lanes and harbors are easily recognizable for their higher traffic density. Geographically, the recorded area comprises the western parts of the Baltic Sea, the Kattegat and parts of the Skagerrak. Temporally, it spans a whole week starting from 16th May 2011. Altogether, 3,702 different vessels (unique MMSIs) grouped into 30 different vessel types were detected. In the first step, clearly wrong measurements as well as measurements generated by offshore structures (e.g., lights) are removed. Afterwards, 3,550 unique vessels remain. For further processing the data points created by each ship are grouped by their corresponding MMSI and connected to tracks. If the time between two measurements with the same MMSI is too large (here, larger than 30 min), the track of the ship is split. Therefore, each vessel can generate more than one track and altogether 25,918 different tracks are detected.

For further investigations, only cargo vessels and tankers are used. In the prepared dataset, there are 1,087 cargo vessels and 386 tankers. The two types have similar movement behavior; therefore, they are treated as one. This means, there will always be one compound model for both types instead of a single one for each.

Due to the huge amount of data and the necessary effort to annotate the whole dataset, only three subareas (called Fehmarn, Kattegat and Baltic) are annotated which reflect specific criteria. A description of the Kattegat and the Fehmarn area can be found in [2].

Figure 2 depicts the normal behavior within the Baltic area. This area consists of two main sea lines. In the east both lines are starting in the north west of the Danish island Bornholm. One line goes in the direction of Copenhagen, the other in the direction of Fehmarn and Lolland. Furthermore, the sea lanes of vessels calling the port at Trelleborg can be seen in the north. As the traffic density of the other possible lanes (e.g. the traffic to Ystad) is low, all other traffic is defined as abnormal behavior

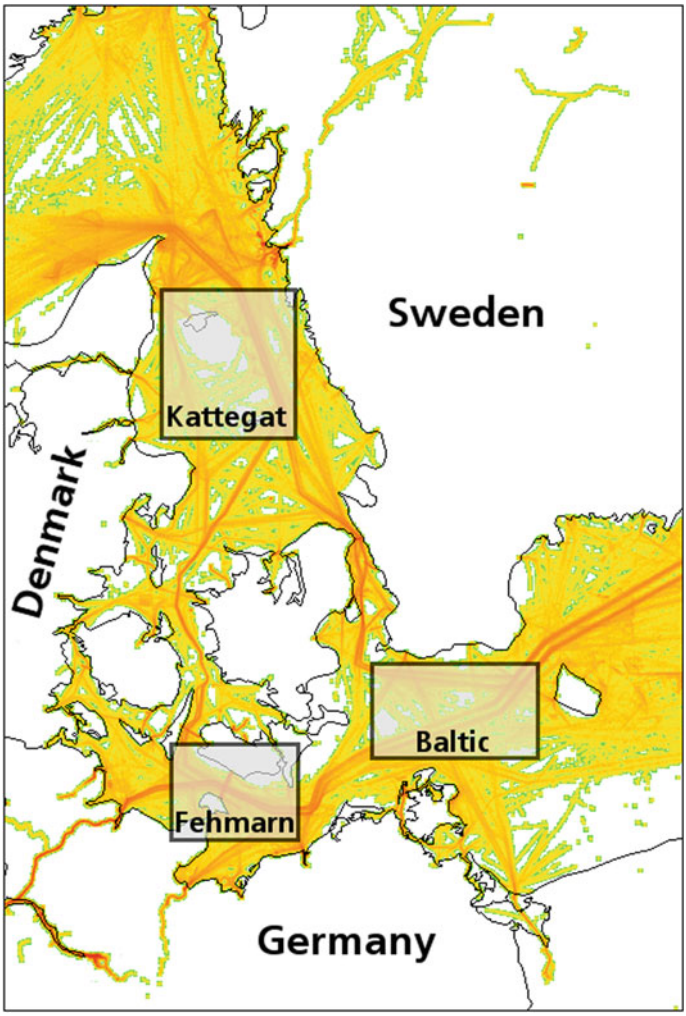

Fig. 1 Heat-map of the vessel traffic in the dataset. The traffic density is encoded by the color, whereas the gradient from *red* to *green* represents the gradient from high density to low density. The *marked areas* (namely Fehmarn, Kattegat and Baltic) are further analyzed

(compare Fig. 3). The resulting area has 698 unique tankers and cargo vessels which generate a total of 26,808 data points.

During the annotation of the tanker and cargo vessels in the designated areas, moored vessels moored in a harbor are removed from the dataset, for the behavior in harbors is out of scope for this work. For the Fehmarn area, 14.5 % of the data points by cargo vessels and 8.7 % of the data points by tankers are marked as unusual

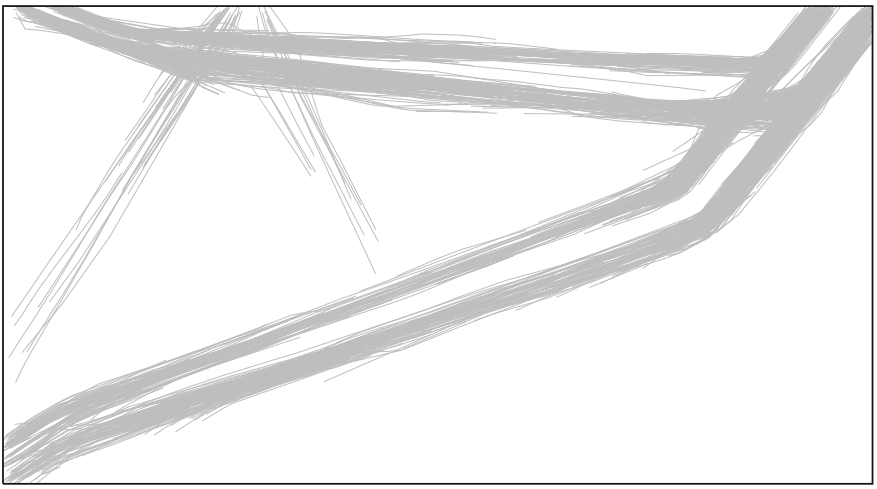

Fig. 2 Normal trajectories (*grey lines*) in the annotated and evaluated area consisting of a part of the Baltic Sea. The *white* background represents water

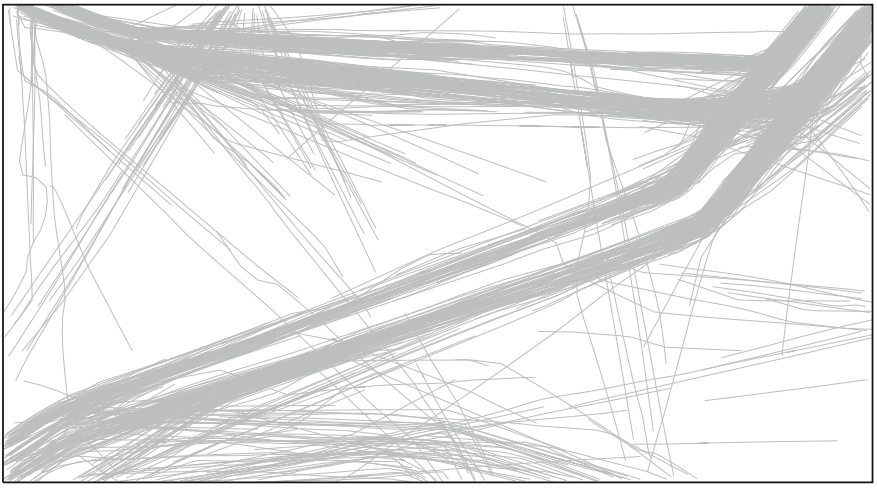

Fig. 3 All trajectories (*grey lines*) in the annotated and evaluated area consisting of a part of the Baltic Sea. The *white* background represents water

ones. In the Kattegat area, 5.4 % of the data points by tankers and 6.6 % of the data points by cargo vessels were annotated as abnormal. For the Baltic area, 15.2 % of data points generated by cargo vessels and 14.4 % of the ones by tankers are marked as anomalies.

4 Test Set-Up

In this section, the two evaluated algorithms, the metrics and the general process of detecting algorithms are introduced. The Gaussian Mixture Model (GMM) and the Kernel Density Estimation (KDE) are chosen as algorithms for the evaluation. The algorithms themselves and the possible parameters are described. As feature vector, the position in latitude p_{lat} and longitude p_{lon} as well as the speed vector split into its latitude v_{lat} and longitude v_{lon} components are used, resulting in

$$x_i = \{p_{\text{lat}}, p_{\text{lon}}, v_{\text{lat}}, v_{\text{lon}}\}$$

for each data point i. A new model has to be trained for each area. Further, each area can be divided by a grid and for each cell in the grid a distinct model has to be trained. The optimal grid-size as well as the parameters of the models have to be estimated. For these purposes, the Python package *scikit-learn* [15] is used.

4.1 Algorithms

4.1.1 Gaussian Mixture Model

A GMM consists of n superimposed multivariate normal distributions called components. Each distribution i has its own mean vector μ_i and covariance matrix Σ_i. Together, they form the parameter set $\theta_i = \{\mu_i, \Sigma_i\}$ for each component i. The dimension of μ_i and Σ_i depend on the number of observed features k. The probability density function is then given by

$$f(x) = \sum_{i=1}^{n} f_g(x, \theta_i)$$

with the density function for each component given by

$$f_g(x, \theta_i) = \frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{|\Sigma_i|}} \exp\left(-\frac{1}{2}(x - \mu_i)^T {\Sigma_i}^{-1}(x - \mu_i)\right).$$

In order to estimate the parameter sets θ_i, the expectation-maximisation (EM) algorithm is used. Prior to this estimation, the number of components n must be available. More details on the GMM and the EM algorithm is given, e.g., by Barber [4].

4.1.2 Kernel Density Estimator

The KDE or Parzen-window density estimation estimates the probability density function (PDF) of a dataset with n data points. Each of these data points is assigned a kernel function $K(x)$ with the bandwidth h. For each kernel function the same bandwidth is chosen. As kernel function, any valid PDF may be chosen. By taking the sum of all kernels evaluated at the point x, the PDF is estimated resulting in

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^k} K\left(\frac{x - x_i}{h}\right).$$

The Gaussian kernel

$$K(x) = \frac{1}{(2\pi)^{\frac{k}{2}} \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right)$$

with the covariance matrix set to the identity matrix $\Sigma = I_k$ is used as kernel function. The bandwidth has a huge impact on the resulting PDF. If it is chosen too small, the resulting estimation will overfit the problem; if the chosen bandwidth is too large, underfitting will occur. Further information on the KDE is available, e.g., by Murphy [14].

4.2 Anomaly Detection

For both algorithms, the detection of abnormal behavior is defined as depicted in Fig. 4. First, a normal model is estimated by using only data with normal behavior. Then, the minimum log-likelihood l_{min} for each model for the training data is calculated. The log-likelihood is the natural logarithm of the likelihood function which is defined as the conditional probability that an outcome is generated by a specific model. By using only normal data for the training, it can be expected that abnormal data will generate a lower log-likelihood.

For each new data point x, the log-likelihood l_{x} for the estimated model is calculated. If $l_{\text{min}} > l_{\text{x}}$ holds true, the data point is considered an anomaly. Thus l_{min} is the boundary between abnormal and normal behavior.

4.3 Metrics

For the evaluation, precision, recall, f1-score, accuracy, false positive rate (FPR), receiver operating characteristic (ROC) and area under ROC are used as metrics to

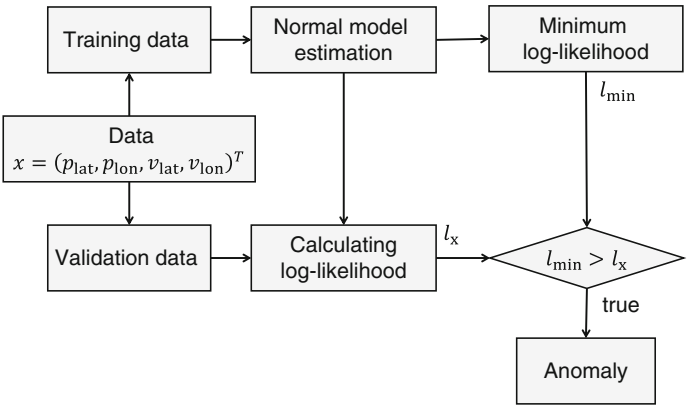

Fig. 4 Flow chart of the anomaly detection algorithm

compare and assess the algorithms. Here, only a brief introduction to the metrics is given. Further explanation are, e.g., given by Manning et al. [12].

The precision is defined as

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}},$$

whereas the recall is defined as

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}.$$

The f1-score is the harmonic mean of the precision and recall. It is defined as

$$\text{f1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

The recall describes the fraction of the positives which are actually classified as positive (true positives). Thus, a small recall means, that there are lots of false negative classifications. The precision describes the fraction of all positively classified results which are actually positives. Hence, a small precision equals a great number of false positive classifications. The accuracy is defined as

$$\text{accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{true negatives} + \text{false negatives} + \text{false negatives}}.$$

It describes the amount of correctly identified object compared to all available objects.

The FPR is given by

$$\text{FPR} = \frac{\text{false positives}}{\text{false positives} + \text{true negatives}}.$$

It describes the probability, that the classifier falsely identifies an object as positive. These metrics are commonly used in classification tasks. For an optimal classifier, the value of each metric except the FPR should be "1". For the FPR, the optimal classifier should yield the value "0".

The ROC curve is a graphical tool to describe the performance of a classifier while varying the discrimination threshold between to classes. It is a plot of the recall (true positive rate) against the false positive rate. If the ROC curve of a classifier is a diagonal line starting in the origin of the coordinate system, it will be equal to a random guess. This is the worst result for a classifier. For further comparison, the area under ROC (AUROC) is used as a metric. The worst value for the AUROC is "0.5", the best is "1". A value of "0.5" would equal a random guessing strategy and an optimal classifier would be described by an AUROC of "1".

5 Empirical Evaluation

Before the results of both algorithms are compared with each other, the optimal parameters for both algorithms have to be determined. Therefore, a k-fold cross-validation as, e.g., described by Witten and Frank [18] is conducted for different parameter combination. The parameters to estimate are the bandwidth for the KDE and the number of components as well as the optimal grid-size for the GMM.

For the cross-validation, each of the validation fold consists of the same ratio of normal and abnormal data in order to ensure that the folds are comparable to each other. The training fold has no anomalies at all. All in all, the available data is divided into k folds. In each step of the cross-validation, the model is trained with $k - 1$ folds and validated by using the remaining fold. The results for the optimal parameters using a 3-fold cross-validation are shown in Table 1. For each step in the cross-validation, the precision, recall, and f1-score are calculated. Finally, the cumulated means of these scores are determined and the parameter set with the highest f1-score is chosen as the best.

In Fig. 5a, b the precision, recall and f1-score for different bandwidths using the data in the Baltic area are shown. The difference between those two figures is the underlying grid. The model for Fig. 5a has no grid, while the model for Fig. 5b has a 3×3 grid. Comparing those figures, the one without grid performs better. Thus the parameter as shown in Table 1 are used in the evaluation. For the KDE, the grid-size is not an important parameter to optimize. Due to the main principle behind the KDE, only data points from the training set which are close to the evaluated data point will have an influence on the resulting probability density function. If the distance between data points from the training set and the evaluated data point is large, the

Table 1 Optimal Parameter

Area	KDE	GMM	
	Bandwidth	# Components	Grid Size
Fehmarn	0.06	75	5×5
Kattegat	0.09	50	3×3
Baltic	0.085	70	1×5

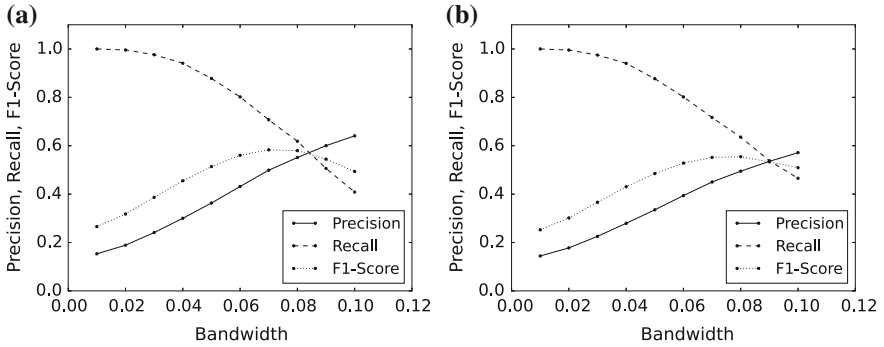

Fig. 5 Precision, recall, and f1-score for different bandwidths and different grids in the Baltic area

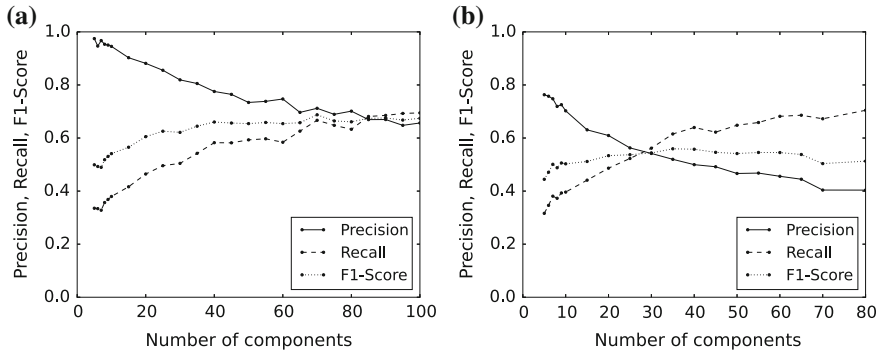

Fig. 6 Precision, recall, and f1-score for different numbers of components and different grids in the Baltic area

resulting value of the kernel function will tend to zero. Thus, omitting some points will only decrease the calculation time, which is not important for this work.

The results for varying the number of components in the Baltic area is shown in Fig. 6a, b. The first has a 1×5 grid, the second a 5×5 grid. These figures are only exemplary to show the different behavior of the algorithm with different parameter configurations. The best performance is achieved with the 1×5 grid. The number of components is chosen like shown in Table 1.

Table 2 Evaluation results

	Fehmarn		Kattegat		Baltic	
	KDE	GMM	KDE	GMM	KDE	GMM
Precision	0.5128	0.5607	0.3844	0.4675	0.5690	0.7013
Recall	0.6405	0.5428	0.4040	0.5250	0.5655	0.6227
F1-Score	0.5696	0.5515	0.3940	0.4946	0.5673	0.6597
Accuracy	0.8705	0.9144	0.9224	0.9376	0.8707	0.9037
FPR	0.0940	0.0456	0.0431	0.0369	0.0755	0.0467
AUROC	0.8719	0.8165	0.8514	0.7771	0.8656	0.8981

As the main task is the detection of anomalies, a data point which is detected and annotated as anomaly is a true positive. The results for the different areas using the different metrics are given in Table 2. Furthermore, the ROC for each area and algorithm combination is depicted in Fig. 7. For each fold, the ROC is drawn, with the AUROC value stated in the legend. Furthermore, the mean of the fold is depicted.

Comparing the scores of the algorithms as depicted in Table 2, it is clear, that neither of the algorithms delivers a good performance for the detection of anomalies. The overall performance of the GMM measured by the f1-score is always higher as the one of the KDE. In the Fehmarn area, both algorithms have nearly an equal score, while in the Kattegat area the GMM's f1-score is 25.5 % higher and in the Baltic area it is 16.3 % higher.

The accuracy of the GMM in all areas is higher than the KDE's. In the Fehrmarn area, it is 5 %, in the Kattegat area 1.6 %, and in the Baltic area 3.8 % higher. The FPR of the GMM is lower in all areas than the one of the KDE. Thus, the GMM's performance is always better for these metrics in the annotated areas.

Comparing the AUROC of the two algorithms, the KDE performs better in the Fehmarn area and the Kattegat area, while the GMM has a better result in the Baltic area. These results can also be derived by comparing the shape of the ROC curves in Fig. 7. The ROC curve of the GMM is less steep than the one of the KDE in the Fehmarn area and the Kattegat area, resulting in the lower AUROC value of the GMM in these particular areas.

Figures 8, 10, and 12 show the results for the GMM algorithm for the different folds in the Baltic area. The equivalent figures for the KDE algorithm are Figs. 9, 11, and 13. Similar figures for the other areas can be found in [2]. Both algorithms will perform quite well, if the data point which is to be analyzed is far away from the normal sea lanes as it can be seen in the southern part of each figure. Most of the points in the southern part are correctly identified as either anomalies or normal data points. A problem for both algorithms are the more sparsely and less dense trajectories to and from Trelleborg. These normal trajectories are found as abnormal behavior as seen, e.g., in Figs. 8 and 9.

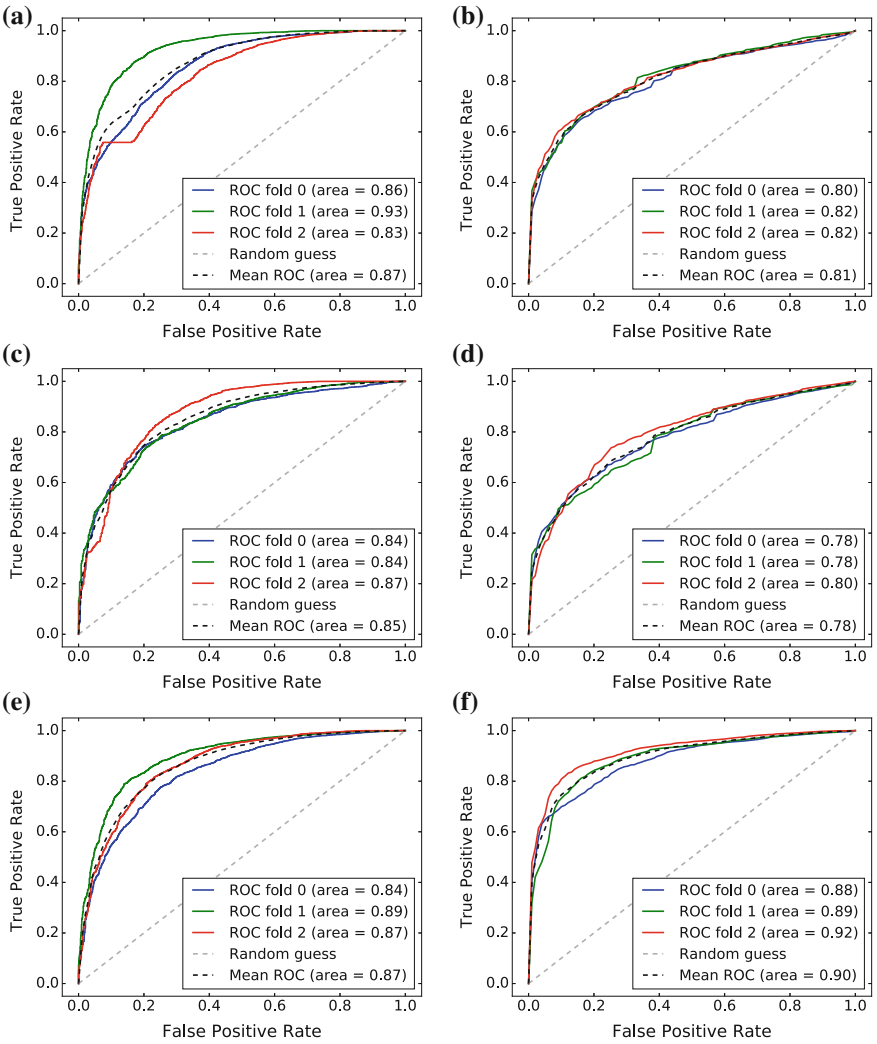

Fig. 7 The ROC for the KDE and GMM in the different areas. In each figure, the ROC curve for each fold as well as the mean ROC is shown. In the legend, the area under each curve is stated. **a** KDE—Fehmarn. **b** GMM—Fehmarn. **c** KDE—Kattegat. **d** GMM—Kattegat. **e** KDE—Baltic. **f** GMM—Baltic

If a trajectory is quite near to the normal behavior, and if the speed and heading of the vessel is similar to the normal model, the trajectory will not be identified as anomaly. In Figs. 12 and 13, this problem can be seen in the eastern region, where several vessels are entering the sea lanes at no distinct point.

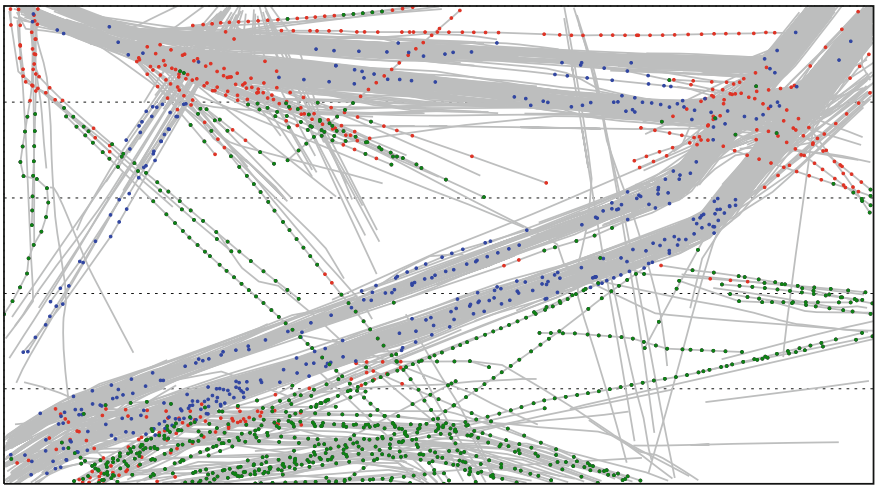

Fig. 8 GMM results for the Baltic area for one fold. The *grey lines* correspond to all trajectories, the *grey dotted lines* represent the used grid, the *dots* represent some evaluated data points. *Green dots* represent correctly found anomalies, *red dots* missed anomalies and *blue dots* normal points which are falsely declared as anomaly

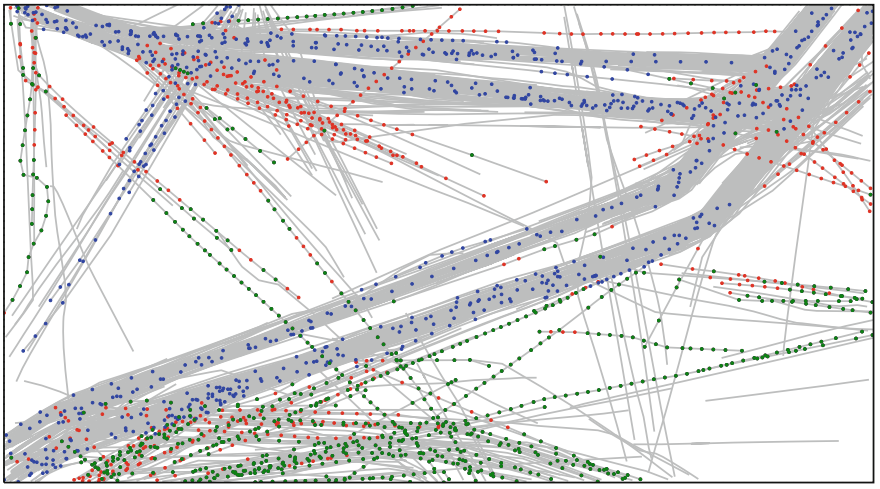

Fig. 9 KDE results for the Baltic area for one fold. The *grey lines* correspond to all trajectories, the *dots* represent some evaluated data points. *Green dots* represent correctly found anomalies, *red dots* missed anomalies and *blue dots* normal points which are falsely declared as anomaly

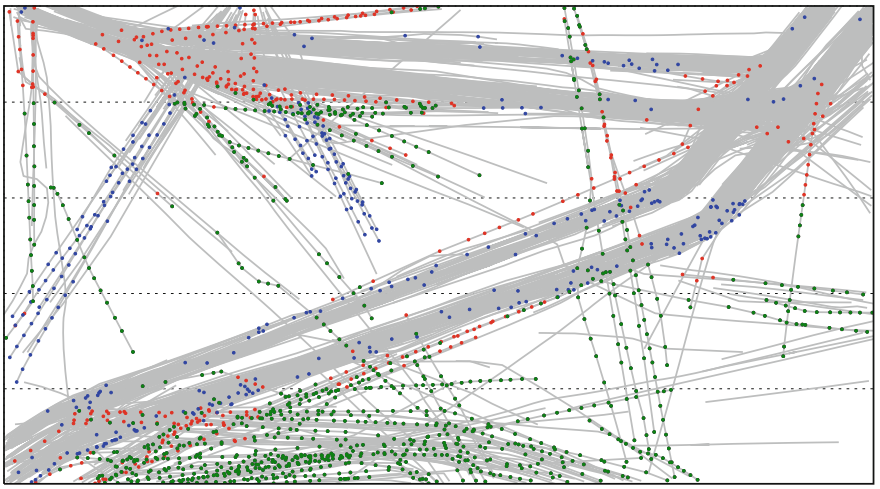

Fig. 10 GMM results for the Baltic area for one fold. The *grey lines* correspond to all trajectories, the *grey dotted lines* represent the used grid, the *dots* represent some evaluated data points. *Green dots* represent correctly found anomalies, *red dots* missed anomalies and *blue dots* normal points which are falsely declared as anomaly

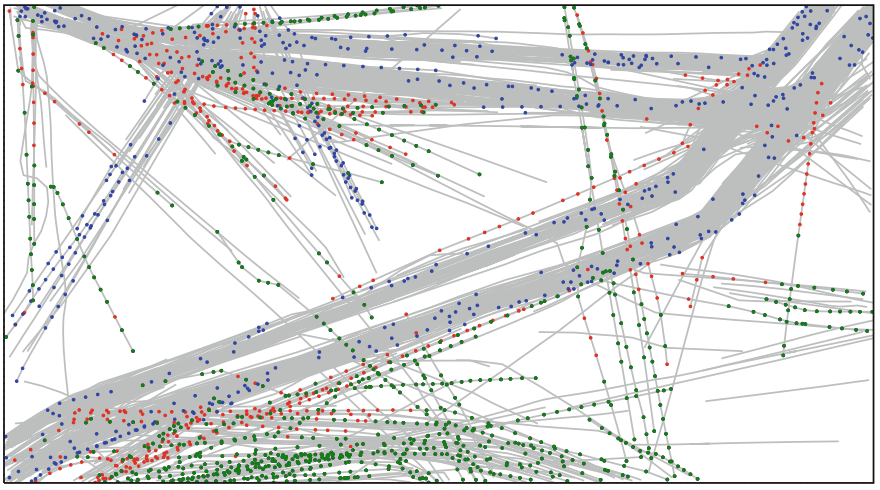

Fig. 11 KDE results for the Baltic area for one fold. The *grey lines* correspond to all trajectories, the *dots* represent some evaluated data points. *Green dots* represent correctly found anomalies, *red dots* missed anomalies and *blue dots* normal points which are falsely declared as anomaly

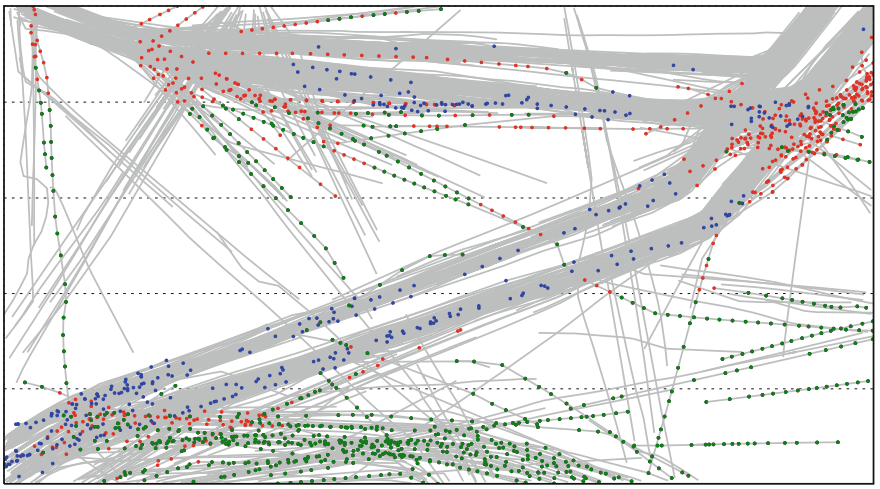

Fig. 12 GMM results for the Baltic area for one fold. The *grey lines* correspond to all trajectories, the *grey dotted lines* represent the used grid, the *dots* represent some evaluated data points. *Green dots* represent correctly found anomalies, *red dots* missed anomalies and *blue dots* normal points which are falsely declared as anomaly

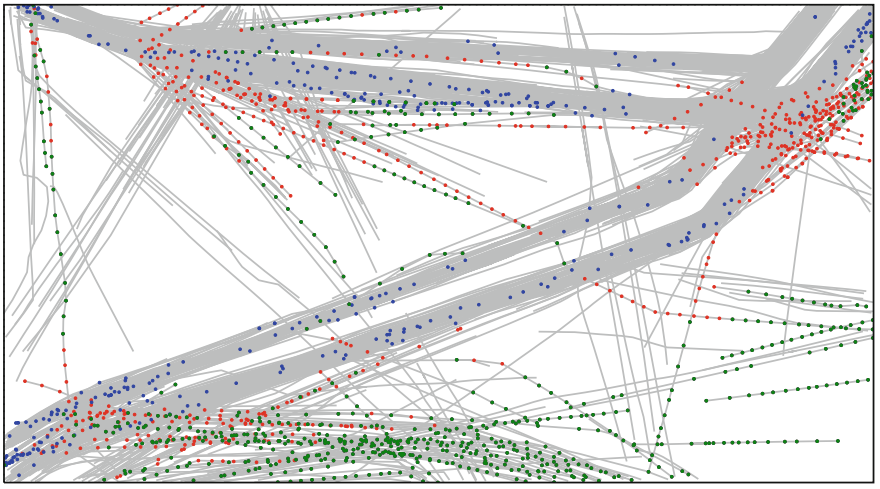

Fig. 13 KDE results for the Baltic area for one fold. The *grey lines* correspond to all trajectories, the *dots* represent some evaluated data points. *Green dots* represent correctly found anomalies, *red dots* missed anomalies and *blue dots* normal points which are falsely declared as anomaly

Another problem can occur, if there is not enough data in one of the grid-cells to build a normal model. In this case, there will be no valid model for the decision about abnormal behavior. Therefore, a different strategy has to be chosen; e.g., all data points in the cell are anomalies, or there are no anomalies in the cell. Here, all points in those cells are marked as anomaly. This problem arises especially in the Fehmarn area as shown in [2]. Furthermore, several falsely classified anomalies occur in all areas with both algorithms (blue dots).

6 Conclusion

The two algorithms generate a large amount of false positives and false negatives. Therefore, the results are not as good as it would be expected for a support system. Both algorithm estimate the underlying PDF of the sea traffic. For the GMM, the PDF is expected to consists of superimposed multivariate normal distributions, while the real PDF is unknown and might be of a different kind. This might be a reason for the result. A KDE is able to estimate an arbitrary PDF, if enough training data is available. If not enough data is available, the resulting PDF might differ significantly from the true PDF.

Both algorithms only evaluate a single point of the trajectory at a time. Therefore, the whole trajectory is never considered. Thus, Trajectories as shown in Fig. 14 will probably not be recognized correctly. Each data point of the orange trajectory for itself might be detected as normal behavior resulting in a normal label for the whole trajectory. An anomaly like this can only be recognized by evaluating the whole trajectory.

By using a grid to divide the area, the following problems might occur: In the border region between two cells, the grid can perform worse than using no grid at all. The EM algorithm fits the components of the GMM to the underlying data of each cell separately. The data in each cell abruptly ends at the grid border. Hence, it

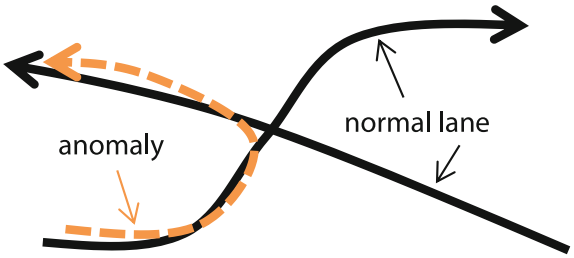

Fig. 14 Problem with point only evaluation. Two trajectories (each is only valid in the direction of the arrow) are crossing. An abnormal behavior is depicted as an *orange dashed line*. It starts on one trajectory and changes to the other during the crossing. Depending on the context, this can be considered as anomaly

is likely to be less dense at the border compared to the center of a cell. This might likely result in the placement of the components of a GMM in the center instead of the border, even though globally observed the data might have the same density at the center and at the border.

Furthermore, if the grid was chosen unfavorably, the resulting model will not be able to learn a sea lane properly, because the grid might divide a sea lane or cut out parts of a sea lane. A similar problem might occur, if there are not enough data points in a cell for the estimation of a normal model. For this case, different strategies are possible, e.g., every point in these areas is marked as anomaly. This might result in cutting sea lanes with normal behavior and detecting this normal behavior as anomaly.

7 Future Work

Even though, the optimal parameters are estimated, the same parameters are used for all grid-cells. Therefore, an improvement could be achieved by estimating the model parameters for each cell separately, respectively to use an adaptive approach for the bandwidth estimation for the KDE. Thus, the difference in density and complexity of each local area would be taken into account.

Currently, only quite simple algorithms for the anomaly detection are examined. The performance of these algorithms was suboptimal. Therefore, the next step is to compare more sophisticated algorithms. These algorithms should consider past points of a track while evaluating a new point. By incorporating the additional information provided using whole or partial trajectories, the results should improve compared to the density estimation using a GMM or a KDE.

Another open point is the annotation of the whole dataset and not only some artificial subsets. Currently, only a small subset of the whole dataset is annotated and inspected during the evaluation. By using the whole dataset, a better overview of the observed area can be used to get a better understanding of the normal behavior, and thus to improve the models. Due to the greater amount of data, the methods of annotating the data must be reconsidered and improved. e.g., to achieve better model results the tracks could be annotated by domain experts or by using another strategy to ensure a consistent and reliable annotation. Also, using sea-maps to gain a better understanding of the sea lanes, shoals etc. will help to improve the annotated ground truth.

Acknowledgments The underlying projects to this article are funded by the WTD 81 of the German Federal Ministry of Defense. The authors are responsible for the content of this article.

References

1. Andersson, M., Johansson, R.: Multiple sensor fusion for effective abnormal behaviour detection in counter-piracy operations. In: Proceedings of the International Waterside Security Conference (WSS), pp. 1–7 (2010)
2. Anneken, M., Fischer, Y., Beyerer, J.: Evaluation and comparison of anomaly detection algorithms in annotated datasets from the maritime domain. In: Proceedings of the SAI Intelligent Systems Conference (2015)
3. Anneken, M., Fischer, Y., Beyerer, J.: Anomaly detection using b-spline control points as feature space in annotated trajectory data from the maritime domain. In: Proceedings of the 8nd International Conference on Agents and Artificial Intelligence (2016). (accepted paper)
4. Barber, D.: Bayesian Reasoning and Machine Learning. Cambridge University Press, cambridge (2014)
5. Brax, C., Niklasson, L.: Enhanced situational awareness in the maritime domain: an agent-based approach for situation management. In: Proc. SPIE, vol. 7352, pp. 735,203–735,203–10 (2009)
6. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 15:1–15:58 (2009)
7. Fischer, Y., Beyerer, J.: Ontologies for probabilistic situation assessment in the maritime domain. In: Proceedings of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), pp. 102–105 (2013)
8. Fischer, Y., Reiswich, A., Beyerer, J.: Modeling and recognizing situations of interest in surveillance applications. In: Proceedings of the IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), pp. 209–215 (2014)
9. Guillarme, N.L., Lerouvreur, X.: Unsupervised extraction of knowledge from s-ais data for maritime situational awareness. In: Proceedings of the 16th International Conference on Information Fusion (FUSION), pp. 2025–2032 (2013)
10. Laxhammar, R., Falkman, G.: Sequential conformal anomaly detection in trajectories based on hausdorff distance. In: Proceedings of the 14th International Conference on Information Fusion (FUSION), pp. 1–8 (2011)
11. Laxhammar, R., Falkman, G., Sviestins, E.: Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density. In: Proceedings of the 12th International Conference Information Fusion (FUSION) (2009)
12. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
13. Morris, B., Trivedi, M.: A survey of vision-based trajectory learning and analysis for surveillance. IEEE Trans. Circuits Syst. Video Technol. **18**(8), 1114–1127 (2008)
14. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT Press, Cambridge (2012)
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
16. Shafer, G., Vovk, V.: A tutorial on conformal prediction. J. Mach. Learn. Res. **9**, 371–421 (2008)
17. de Vries, G.K.D., van Someren, M.: Machine learning for vessel trajectories using compression, alignments and domain knowledge. Expert Syst. Appl. **39**(18), 13,426 – 13,439 (2012)
18. Witten, I.H., Frank, E.: Data Minig: Practical Machine Learning Tools and Techniques with Java Implementations, 2nd edn. Morgan Kaufmann Publishers, Burlington (2005)