Graphical Abstract

Anomaly Detection in Ship Trajectories Using Machine Learning and Dynamic Time Warping

Valérian Mangé, Jean-Yves Tourneret, François Vincent, Laurent Mirambell, Fábio Manzoni Vieira

Highlights

Anomaly Detection in Ship Trajectories Using Machine Learning and Dynamic Time Warping

Valérian Mangé, Jean-Yves Tourneret, François Vincent, Laurent Mirambell, Fábio Manzoni Vieira

- Ship trajectories can be compared using appropriate similarity measures
- Unsupervised anomaly detection methods detect abnormal ship trajectories efficiently
- The proposed methods lead to F1 scores close to 95% for the considered datasets
- Radar and AIS data are complementary to detect abnormal ship trajectories

Anomaly Detection in Ship Trajectories Using Machine Learning and Dynamic Time Warping

Valérian Mangé[a,d,*], Jean-Yves Tourneret[b,d], François Vincent[c], Laurent Mirambell[a], Fábio Manzoni Vieira[a]

[a] *Hensoldt Nexeya France, 1 rond-point Général Eisenhower, Toulouse, 31100, France*
[b] *University of Toulouse, ENSEEIHT-IRIT-TéSA, TéSA, 7 bd de la gare, Toulouse, 31500, France*
[c] *ISAE-Supaéro, 10 avenue Édouard-Belin, Toulouse, 31055, France*
[d] *TéSA Laboratory, 7 boulevard de la Gare, Toulouse, 31500, France*

Abstract

This research paper proposes adaptations of three state-of-the-art anomaly detection algorithms, (One-Class Support Vector Machine, Isolation Forest and Local Outlier Factor), for detecting abnormal behavior in ship trajectories in an unsupervised way. These algorithms are adapted and tested using a wide range of similarity measures built specifically for time series, such as Dynamic Time Warping. The proposed methods are first applied on synthetic Automatic Identification System datasets with available ground truth. Then, they are generalized to handle pairs of Automatic Identification System and radar trajectories to detect unexpected activities, such as route deviations, delays and entering prohibited zones. The performances of the proposed methods are shown to be competitive when compared to the state-of-the-art for abnormal ship behavior detection.

Keywords:
Maritime Surveillance, Automatic Identification System, Radar, Anomaly detection, Dynamic time warping

---

*Corresponding author

Email addresses: `valerian.mange@tesa.prd.fr` (Valérian Mangé), `jean-yves.tourneret@toulouse-inp.fr` (Jean-Yves Tourneret), `francois.vincent@isae-supaero.fr` (François Vincent), `laurent.mirambell@hensoldt.fr` (Laurent Mirambell), `fabio.manzoni@hensoldt.fr` (Fábio Manzoni Vieira)

1. Introduction

1.1. Context

Maritime traffic is essential for the transportation of merchandise across the globe with ships carrying around 90% of all goods. Maritime activities can be subject to illicit actions leading to abnormal ship behavior. According to (Wolsing et al., 2022), there are 5 general types of anomalies: route deviation, unexpected activity, unexpected port arrival, close approach and zone entry. These actions include, but are not limited to, piracy, hijacking, illegal fishing, etc. The prevalent system for monitoring ships today is the Automatic Identification System (AIS). Initially developed to avoid collision between ships in 2004 (Androjna et al., 2021), AIS is mandatory for large ships transporting goods or passengers. Based on the Global Positioning System (GPS), this system is accurate and contains a wide variety of information on ships such as position, velocity and heading as well as the ship type and destination. AIS data is easily acquired by a receiver yielding signals covering large distances, including beyond the horizon when satellites are used as relays.

1.2. Related works

AIS datasets can be extremely large, and the data must be analyzed quickly in many practical applications in order to identify anomalous activities. This problem has been a topic of interest for many companies and researchers. Clustering algorithms such as DBSCAN (Ester et al., 1996) have been considered (Pallotta et al., 2013), (Zhang and Li, 2022), (Zhang et al., 2023), (Li et al., 2024), (Xie et al., 2024). The objective of DBSCAN when detecting abnormal trajectories is to identify the sets of normal behaviors, such as the different nominal maritime routes, and determine outliers with respect to these clusters, which is important for maritime surveillance. In other words, DBSCAN is able to find the routes from historical data while also performing anomaly detection (AD). In (Pallotta et al., 2013), DBSCAN is used directly on the AIS points (as opposed to the entire trajectories of ships) to find way points in routes. The classification of the normal points can then be used to reconstruct the routes and perform route prediction on ships. DBSCAN has also been applied to full ship trajectories using similarity measures such as the Longest Common Subsequence (LCSS) and Dynamic Time Warping (DTW) (Zhang and Li, 2022) and (Zhang et al., 2023).

AD methods based on machine learning algorithms (such as One-Class Support Vector Machines (OC-SVM) (Schölkopf et al., 2001), Isolation Forest (Liu et al., 2012) or Local Outlier Factor (Breunig et al., 2000)) have shown promising results in other industrial applications. OC-SVM has been used for detecting turboshaft engine (Zhao et al., 2020) and railway track anomalies (Ghiasi et al., 2024). An improved version of OC-SVM performing feature selection using wavelets has been investigated in (Sadooghi and Esmaeilzadeh Khadem, 2018). Isolation Forest has shown interesting results for optical emission spectroscopy (Puggini and McLoone, 2018) and has been combined with a Bayesian approach to perform active learning (Sartor et al., 2024). An interpretability of AD results obtained using isolation forest was studied in (Arcudi et al., 2024). Deep isolation forest (Xu et al., 2023) is a recent hybrid method between deep learning and standard machine learning, which uses random representations of the data generated by neural networks to train an isolation forest algorithm. Local outlier factor, coupled with a weighted Gaussian process, is able to monitor the state of lithium-ion batteries (Qian et al., 2024). A non-parametric version of local outlier factor studied in (Entezami et al., 2023) automatically selects the number of neighbors using the SANS algorithm (Zhu et al., 2016). Deep learning is currently receiving an increasing interest to solve AI problems including anomaly detection (Pang et al., 2021). A standard approach is to extract features using neural networks and to detect anomalies using these features as the input of a more standard anomaly detector (Bono et al., 2023) (Xu et al., 2023).

AD methods have also been investigated for maritime surveillance. The authors of (Laxhammar and Falkman, 2014) proposed detecting abnormal trajectories using conformal AD with the Hausdorff distance (method referred to as SHNN-CAD in this paper). The method was compared to state-of-the-art algorithms such as One-Class SVM with the Euclidean distance. Note that the Euclidean distance requires the ship trajectories to have the same length, which is not always the case in our application. The One-Class SVM method was also used in (Piciarelli et al., 2008) to detect abnormal trajectories when applied to urban roads. Note that the synthetic dataset considered in (Piciarelli et al., 2008) was generated with a ground truth, which is important for performance evaluation. This dataset will be used for performance evaluation in this work. Finally, it is interesting to note that deep learning methods based on variational recurrent neural networks (Nguyen et al., 2018) or transformers (Xie et al., 2024) have been recently considered for detecting abnormal ship trajectories.

1.3. Contributions

The contributions of this paper are summarized below:

- We adapt three standard AD algorithms (One-Class SVM, Isolation Forest, Local Outlier Factor) to make them applicable to ship trajectories having potentially different lengths using appropriate similarity measures. These algorithms do not require any feature extraction step and compare the ship trajectories without preprocessing. This is interesting since determining the best features is generally complicated in many practical applications. The performance of these algorithms is evaluated on a synthetic dataset with known ground truth both in terms of accuracy and computational efficiency.

- The performance of the AD algorithms studied in this paper is determined for detecting anomalies in datasets containing real AIS ship trajectories. The benefits of detecting abnormal ship trajectories using both AIS and radar data, when the AIS is interrupted or spoofed, is also shown qualitatively.

1.4. Paper organization

The organization of the paper is as follows. Section 2 summarizes the principles of One-Class SVM, Isolation Forest, and Local Outlier Factor, which have become reference AD methods. Multiple similarity measures that can be used to compare time series are reviewed in Section 3. Section 4 explains how to combine the methods introduced in Section 2 with these similarity measures to detect abnormal ship trajectories, which is the main contribution of this paper. Performance measures that will be used to evaluate the algorithms are provided in Section 5. Experiments conducted on synthetic and real data are presented and analyzed in Section 6. Conclusions and future work are finally reported in Section 7.

2. Anomaly Detection

This section introduces three reference AD algorithms that have been used intensively in practical applications, i.e., OC-SVM (Schölkopf et al., 2001), Isolation Forest (IF) (Liu et al., 2012) and Local Outlier Factor (LOF) (Breunig et al., 2000). The DBSCAN algorithm (Ester et al., 1996) is also introduced in Section 2.4 since it will be considered in our experiments. As explained in (Chandola et al., 2009), "anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior". From this definition, it is obvious that a fundamental property of anomalies is that they are scarce in the datasets of interest. Thus, AD algorithms have to be trained using datasets containing potential anomalies, with the property of having few anomalies with respect to the number of inliers.

2.1. One-Class SVM

Support vector machines (SVMs) were first used to perform supervised classification by finding a separating hyperplane between two classes in an appropriate space. Inspired by SVMs, the OC-SVM method searches a hyperplane separating the data from the origin in order to detect potential anomalies (Schölkopf et al., 2001). As for SVMs, the kernel trick can be applied to OC-SVM to project the data into a space of higher dimension referred to as reproducing kernel Hilbert space (RKHS). Consider a transformation Φ such that $\Phi(\mathbf{x}_i)$ belongs to this RKHS. The separating hyperplane of OC-SVM defined by the equation $\omega^T\Phi(\mathbf{x}) - \rho = 0$ can be determined by solving the following optimization problem:

$$\begin{aligned} \min_{\omega,\rho,\xi_i} \quad & \frac{1}{2}\|\omega\|^2 - \rho + \frac{1}{\nu_{\mathrm{OC}}N}\sum_{i=1}^{N}\xi_i \\ \text{with} \quad & \langle \omega, \Phi(\mathbf{x}_i)\rangle \geq \rho - \xi_i, \forall i = 1, \ldots, N, \end{aligned} \tag{1}$$

where $\langle ., . \rangle$ is the scalar product defined in the chosen RKHS and ν_{OC} is the maximum proportion of abnormal data in the dataset $\mathcal{X}$. The kernel trick consists of introducing a kernel κ such that $\kappa(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle, \forall(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{X}$. The solution of (1) after applying the transformation Φ to the data vectors $\mathbf{x}_i$ is then defined as $\omega = \sum_{i=1}^{N} \alpha_i \Phi(\mathbf{x}_i)$ with $0 \leq \alpha_i \leq \frac{1}{\nu_{\mathrm{OC}}N}$ and $\sum_{i=1}^{N} \alpha_i = 1$. To determine whether the vector $\mathbf{x}$ is an inlier or an anomaly, the following decision function is considered:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) - \rho\right), \tag{2}$$

where "sgn" is the sign function. The vectors $\mathbf{x}_i$ satisfying $\alpha_i > 0$ are referred to as support vectors. Note that the solution of the OC-SVM problem only depends on the support vectors, which makes the decision function of OC-SVM depend on a limited number of vectors from the database.

2.2. Isolation Forest

This section summarizes the theory behind IF proposed in (Liu et al., 2012) and its extension to time series called functional isolation forest (FIF) (Staerman et al., 2019). Consider a training set $\mathcal{X}_\Phi \subset \mathcal{X}$. An isolation tree is a decision tree created to isolate a given vector $\mathbf{x}_i$. At a given iteration, IF queries one feature j randomly from the vectors of the training set that have not been isolated yet. A threshold τ is chosen randomly between the maximum and the minimum of this feature in $\mathcal{X}_\Phi$. If $x_{k,j} > \tau$, the vector $\mathbf{x}_k$ is included in the right branch of the tree starting from the node. Otherwise, it is directed to the left branch of the tree. An isolation tree is obtained after isolating each vector from $\mathcal{X}_\Phi$. The process is repeated several times to build a forest of isolation trees referred to as isolation forest, the number of trees being chosen by the user. After building the isolation forest, the anomaly scores $s(\mathbf{x}_i, \psi)$ are computed as follows:

$$s(\mathbf{x}_i, \psi) = 2^{-\frac{E[h(\mathbf{x}_i)]}{c(\psi)}}, \tag{3}$$

where $c(\psi) = 2\left[\ln(\psi - 1) + \gamma - \frac{\psi-1}{N}\right]$, $h(\mathbf{x}_i)$ is the depth of $\mathbf{x}_i$ in a given tree, $E[h(\mathbf{x}_i)]$ is the average depth of $\mathbf{x}_i$ over all trees and γ is the Euler constant (see (Liu et al., 2012) for details). Once the anomaly scores have been obtained, a fraction of anomalies ν_{IF} is chosen by the user (corresponding to the highest scores) to detect the anomalies. The number of trees used to build the forest has to be chosen by the user. In the experiments considered in Section 6, 200 trees were sufficient to ensure convergence of the path lengths. An interesting property of IF is that it can handle data with discrete features. This property is interesting because AIS data contains categorical features such as the type of ship.

Functional Isolation Forest (FIF) (Staerman et al., 2019) is an extension of IF for AD in functional multivariate data. The training subset $\mathcal{X}_\Phi$ is compared to a chosen dictionary $\mathcal{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_m]$ using a similarity score s, where m is the number of elements in $\mathcal{D}$. The score between the ith vector of the dataset and the jth element of the dictionary, denoted as $s(\mathbf{x}_i, \mathbf{d}_j)$, is then used to isolate each vector of the database. More specifically, after choosing randomly an atom $\mathbf{d}_j$ from the dictionary, all elements $\mathbf{x}_i$ such that $s(\mathbf{x}_i, \mathbf{d}_j) > \tau$ are placed in the right branch of the child node, where τ is chosen randomly between the minimum and maximum scores obtained with the chosen atom $\mathbf{d}_j$. The remaining vectors are placed in the left branch of the node. For the purpose of AD in ship trajectories, the chosen dictionary is the self dictionary which consists of the data itself, i.e., $\mathcal{D} = \mathcal{X}_\Phi$. In other words, each trajectory is compared to all the other trajectories and the anomaly score is computed using all the anomaly scores resulting from these comparisons (see (Staerman et al., 2019) for other choices).

2.3. Local Outlier Factor

LOF (Breunig et al., 2000) is a state-of-the-art AD algorithm that detects anomalies in a dataset by using the densities of the different vectors belonging to this dataset. Several metrics have to be defined for computing the LOF score of a point $\mathbf{x}_i$ including its k-distance and its local reachability density. The k-distance of $\mathbf{x}_i$, denoted as $\mathrm{kd}(\mathbf{x}_i)$, is the distance between the vector $\mathbf{x}_i$ and its k-th nearest neighbor. The reachability distance between $\mathbf{x}_i$ and a point $\mathbf{x}$ is defined as $\mathrm{rd}_k(\mathbf{x}_i, \mathbf{x}) = \max\{\mathrm{kd}(\mathbf{x}_i), \|\mathbf{x}_i - \mathbf{x}\|\}$, where $\|.\|$ denotes the ℓ_2 norm. With this definition, all points in the neighborhood of $\mathbf{x}_i$ have similar reachability distances contrary to points located far from $\mathbf{x}_i$. The local reachability density of $\mathbf{x}_i$ is defined as

$$\mathrm{lrd}_k(\mathbf{x}_i) = k \left(\sum_{\mathbf{x} \in V_i} \mathrm{rd}_k(\mathbf{x}, \mathbf{x}_i) \right)^{-1}, \tag{4}$$

where V_i is the neighborhood of $\mathbf{x}_i$ containing its k-nearest neighbors. Finally, the LOF score of $\mathbf{x}_i$ is the average ratio of local reachability densities of $\mathbf{x}_i$ with those of its k-nearest neighbors, i.e.,

$$\mathrm{LOF}_k(\mathbf{x}_i) = \sum_{\mathbf{x} \in V_i} \frac{\mathrm{lrd}_k(\mathbf{x})}{\mathrm{lrd}_k(\mathbf{x}_i)}. \tag{5}$$

Inliers have a LOF score close to 1 whereas outliers have a larger score. To convert the scores into predictions, the vectors with the ν_{LOF} highest scores can be declared as anomalies.

Choosing the number of neighbors k to compute the local reachability densities deserves some attention. A simple guideline given by (Breunig et al., 2000) is to examine at least the 10 nearest neighbors. The SANS algorithm studied in (Zhu et al., 2016; Entezami et al., 2023) allows the number of neighbors to be determined dynamically. In short, k is chosen as the smallest value such so that 1) any normal point $\boldsymbol{x}_i$ contains another point $\boldsymbol{x}_j$ in its k-nearest neighbors and 2) $\boldsymbol{x}_i$ also belongs to the k-nearest neighbors of $\boldsymbol{x}_j$. The two vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are then natural neighbors also called mutual neighbors.

2.4. Density-Based Spatial Clustering of Applications with Noise

DBSCAN (Ester et al., 1996) is a clustering algorithm that is able to detect outliers present in the training dataset after performing unsupervised classification. This algorithm has been applied successfully for the analysis of ship trajectories, e.g. for nominal maritime routes identification (Zhang and Li, 2022) and abnormal behavior detection (Zhang et al., 2023). DBSCAN also computes point densities, as explained in what follows. Consider a radius neighborhood ϵ and a minimum number of points minPts fixed by the user. The dataset is initially divided into three categories containing core points, border points and outliers. A core point has at least minPts other points in its vicinity of radius ϵ, in contrast to border points and outliers. A border point can be reached directly by a core point but does not have minPts other points in its vicinity. An outlier does not have minPts other points in its vicinity and cannot be reached by a core point with a distance less than ϵ. Clusters are finally constructed such that all points of the same cluster are reachable by another core point belonging to that group. Heuristics to find appropriate values for ϵ and minPts are available in the literature (Ester et al., 1996). More details about the choice of these parameters are given in Section 6. Note that the value of ϵ can significantly influence the number of clusters. While adjusting the value of this parameter can be challenging in practical applications, a variant of DBSCAN named Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) uses hierarchical clustering to perform DBSCAN for multiple values of ϵ (Campello et al., 2013).

3. Similarity measures for time series

The methods presented in Section 2 all require the vectors $\mathbf{x}_i$ from the training data to have the same dimensions. This constraint is necessary to compare one chosen feature for IF or to compute the Euclidean norm between two vectors for the other methods. This section first introduces basic similarity measures for time series (Section 3.1) and then explains how to generalize these similarity measures using known operations such as kernels or cosine similarity (Section 3.2).

3.1. Basic similarity measures

This section introduces core similarity measures used to compare multivariate time series based on Dynamic Time Warping (DTW) (Keogh and

Ratanamahatana, 2005), the Time Warp Edit Distance (TWED) (Marteau, 2009) and the Longest Common Subsequence (LCSS) (Zhang and Li, 2022). Since each multivariate time series is described by multiple features at several time steps, we concatenate all these features into matrices. Note that one column of these matrices contains the features associated with a given time instant. Denote as $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ and $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_m]$ two matrices to be compared, where n and m are the number of time instants of these time series and the vectors $\mathbf{u}_i$ and $\mathbf{t}_j$ with $i = 1, ..., n$ and $j = 1, \ldots, m$ have the same dimension D and represent samples at times i and j respectively.

3.1.1. Dynamic Time Warping

DTW is one of the most common measures used to compare time series, that was considered for speech signals in (Keogh and Ratanamahatana, 2005). The DTW cumulated score between the first i columns of $\mathbf{U}$ and the first j columns of $\mathbf{T}$ is defined as:

$$s_{\mathrm{DTW}}(i,j) = d(\mathbf{u}_i, \mathbf{t}_j) + \min \begin{cases} s_{\mathrm{DTW}}(i-1, j-1) \\ s_{\mathrm{DTW}}(i, j-1) \\ s_{\mathrm{DTW}}(i-1, j) \end{cases} \tag{6}$$

where $d(\mathbf{u}_i, \mathbf{t}_j)$ is the Euclidean distance between the vectors $\mathbf{u}_i$ and $\mathbf{t}_j$. The DTW distance between the matrices $\mathbf{U}$ and $\mathbf{T}$ is $\mathrm{DTW}(\mathbf{U}, \mathbf{T}) = s_{\mathrm{DTW}}(n, m)$. Note that DTW does not have all the necessary properties of a distance, e.g., the vectors $[0, 1, 1]$ and $[0, 0, 1]$ have the same DTW similarity equal to 0. However, this metric remains a common measure used to compare time series and its computational complexity is quite limited (see Section 6 for more details about the execution times of the different algorithms).

3.1.2. Time Warp Edit Distance

TWED is another popular similarity measure for analyzing a historical dataset containing trajectories (Marteau, 2009). It is defined recursively as follows:

$$s_{\mathrm{TWED}}(i,j) = \min \begin{cases} s(i-1, j-1) + d(\mathbf{t}_i, \mathbf{u}_j) + d(\mathbf{t}_{i-1}, \mathbf{u}_{j-1}) \\ s(i-1, j) + d(\mathbf{t}_{i-1}, \mathbf{t}_i) + \lambda \\ s(i, j-1) + d(\mathbf{u}_{j-1}, \mathbf{u}_j) + \lambda \end{cases} \tag{7}$$

for the two multivariate time series $\mathbf{U}$ and $\mathbf{T}$. The first term of the minimum corresponds to a match between $\mathbf{U}$ and $\mathbf{T}$, whereas the second and third terms are obtained after deleting a point in $\mathbf{T}$ and $\mathbf{U}$, respectively. The hyper-parameter λ is chosen by the user to penalize deletions. Unlike DTW, TWED is a distance in the mathematical sense.

3.1.3. Longest Common Subsequence

The LCSS was used to compare textual data in (Kiwi et al., 2005). It was considered to compare trajectories in (Zhang and Li, 2022) and is defined as:

$$c_{LCSS}(\mathbf{U}, \mathbf{T}) = \begin{cases} c_{\mathrm{LCSS}}(\mathbf{U}, \mathbf{T}) = 0 \text{ if } n = m = 0 \\ 1 + c_{\mathrm{LCSS}}(\mathrm{Tail}(\mathbf{U}), \mathrm{Tail}(\mathbf{T})) \text{ if } d(\mathbf{t}_i, \mathbf{u}_j) \leq \eta \\ \max(c_{\mathrm{LCSS}}(\mathrm{Tail}(\mathbf{U}), \mathbf{T}), c_{\mathrm{LCSS}}(\mathbf{U}, \mathrm{Tail}(\mathbf{T}))) \text{ otherwise} \end{cases} \tag{8}$$

where $\mathrm{Tail}(\mathbf{U})$ is the list obtained after removing the first element of $\mathbf{U}$ (this implies that $\mathbf{U}$ is not empty), $d(\mathbf{t}_i, \mathbf{u}_j)$ is the Euclidean distance between the vectors $\mathbf{u}_j$ and $\mathbf{t}_i$ and η is a parameter chosen by user to describe how close two points should be to be declared as similar. The authors of (Zhang and Li, 2022) have chosen to normalize the LCSS as follows:

$$\mathrm{LCSS}(\mathbf{U}, \mathbf{T}) = 1 - \frac{c_{\mathrm{LCSS}}(\mathbf{U}, \mathbf{T})}{\min(n, m)}, \tag{9}$$

and have used $\eta = 1$ nm (nautical mile) using monthly traffic flow charts.

3.2. Extending the similarity measures

The similarity measures introduced in the previous section can be used directly to compare time series but can also be generalized using kernels and the cosine similarity. These generalizations are presented in this section.

3.2.1. Kernels

Kernels can be used to compute scalar products in an implicit space of higher dimension than the original data space following the kernel trick. The aim is to map the data into a space where the normal data and the anomalies can be separated by a hyperplane, which is not necessarily the case in the original data space. This kernel trick has been used intensively in the frame of SVMs for many practical applications such as geoscience (Honarkhah and Caers, 2010), text categorization (Pradhan et al., 2004) and handwritten character recognition (Decoste and Schölkopf, 2002). One important kernel is the Gaussian kernel (GK), defined by $\mathrm{GK}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$, where

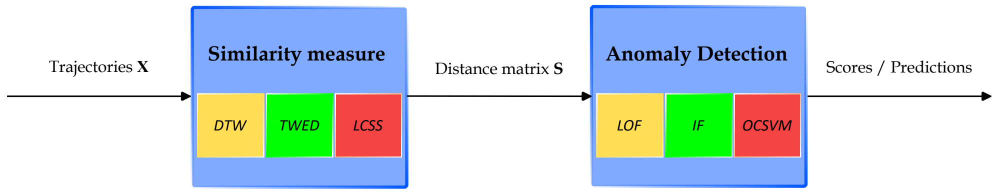


Figure 1: Flowchart of the proposed method.

$\sigma > 0$ is a bandwidth parameter set by the user. It can be shown that this kernel projects the data into a space of infinite dimension, allowing separability between normal data and anomalies in many cases.

When considering time series, the Gaussian kernel needs to be modified to be applied to data having different lengths. Kernels can be adapted to time series using the following formulation investigated in (Badiane and Cunningham, 2021):

$$\kappa_s(\mathbf{U}, \mathbf{T}) = \exp\left[-\frac{s^2(\mathbf{U}, \mathbf{T})}{2\sigma^2}\right], \tag{10}$$

where s is an appropriate similarity measure such as DTW.

3.2.2. Cosine similarity

Another possible extension is to use the cosine similarity. Given a similarity measure s, two metrics sim_s (obtained from the parallelogram law) and $\cos_s$ (cosine similarity) (Sidorov et al., 2014) can be defined as follows:

$$\begin{aligned} \text{sim}_s(\mathbf{U}, \mathbf{T}) &= \frac{s^2(\mathbf{U}, -\mathbf{T}) - s^2(\mathbf{U}, \mathbf{T})}{4}, \\ \cos_s(\mathbf{U}, \mathbf{T}) &= \frac{\text{sim}_s(\mathbf{U}, \mathbf{T})}{||\mathbf{U}|| \times ||\mathbf{T}||}. \end{aligned} \tag{11}$$

Note that these two measures can be quite costly to evaluate since the similarity measure s has to be computed twice in all cases, along with two norms for the cosine similarity.

4. Anomaly detection in ship trajectories

This section studies three new AD methods for detecting abnormal ship trajectories, which is the main contribution of this paper. More precisely, we propose to modify the three state-of-the-art AD algorithms OC-SVM, IF and LOF (summarized in Section 2) to detect abnormal ship behaviour using AIS and potentially radar data. The key point behind these adaptations is to replace the Euclidean norm or the associated scalar product by a similarity measure suited for time series, such as those studied in Section 3. A training set composed of multivariate time series $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_N\}$ is considered, where $\mathbf{X}_i = [\mathbf{x}_{i,t_1}, \dots, \mathbf{x}_{i,t_i}]$ contains all the multivariate measurements of object i from time t_1 to t_i. Note that $\mathbf{x}_{i,t}$ typically contains the position, speed and heading of the ith observed ship at time t. If position and speed are used as features, these methods should detect route deviation, unexpected activity, port arrival and zone entry, as described in (Wolsing et al., 2022). The different steps of the proposed methods are summarized in Fig 1.

4.1. One-Class SVM for Time Series (OCTS)

In order to detect abnormal ship trajectories, we propose to use the OC-SVM algorithm on the observed data matrix $\mathbf{X}$ with a Gaussian kernel κ_s defined as in (10), where s is the DTW similarity measure. The proposed decision function for a trajectory $\mathbf{X}_j$ is defined by:

$$f_s(\mathbf{X}_j) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i \kappa_s(\mathbf{X}_i, \mathbf{X}_j) - \rho + \delta\right). \tag{12}$$

The parameter σ appearing in (10) should be chosen carefully. We propose to follow the strategy adopted in (Trinh et al., 2017), which consists of determining the value of σ maximizing the following cost function

$$J(\sigma^2) = \frac{2}{N} \sum_{i=1}^{N} \left[\exp\left(-\frac{C(\mathbf{X}_i)}{2\sigma^2}\right) - \exp\left(-\frac{F(\mathbf{X}_i)}{2\sigma^2}\right)\right], \tag{13}$$

with the following definitions (for $i = 1, ..., N$):

$$\begin{aligned} F(\mathbf{X}_i) &= \max_{1 \le j \le N} s^2(\mathbf{X}_i, \mathbf{X}_j), \\ C(\mathbf{X}_i) &= \min_{1 \le j \le N, i \ne j} s^2(\mathbf{X}_i, \mathbf{X}_j). \end{aligned} \tag{14}$$

The cost function J can be maximized using an optimization method such as the simplex method, initialized with the median pairwise similarity measure between the data in $\mathcal{X}$, as suggested in (Aggarwal, 2016, p. 83-88).

In order to account for the presence of outliers, we propose to consider only the $(1 - \nu_{\text{OC}})\%$ smaller pairwise similarity measures in order to filter the anomalies. The matrix associated with all the pairwise similarity measures is denoted as $\mathbf{S} = [s_{ij}]_{1\leq i,j\leq N}$ where $s_{ij} = s(\mathbf{X}_i, \mathbf{X}_j)$. In order to reduce the number of false alarms, we propose to include a shifting parameter δ as in (Trinh et al., 2017) to shift the hyperplane away from the normal data. This operation is performed after training the model with a chosen value of ν_{OC} and a kernel κ_s, which leads to (12). If a ground truth is available, an appropriate value of δ can be selected by maximizing the F1-score or with the Precision Recall (PR) curve, as in some of our experiments. Another option for determining an appropriate value of δ (without a ground truth) is the "knee" method (also referred to as "elbow" method) (Satopaa et al., 2011), which determines the point of maximum curvature in the function associated with the anomaly scores of the training set.

4.2. ISolatIon Forest for Time sERies (SIFTER)

The Isolation Forest can be modified to handle ship trajectories with potentially different lengths. We propose to consider a self dictionary, meaning that the pairwise similarity matrix $\mathbf{S}$ is computed using all the ship trajectories. IF is then applied to the columns of $\mathbf{S}$. In order to avoid the curse of dimensionality (Liu et al., 2012), we propose in a second step to apply a principal component analysis (PCA) (Jolliffe, 2002) to the columns of $\mathbf{S}$. PCA computes the eigenvectors $\mathbf{w}_1, \ldots, \mathbf{w}_D$ of $\mathbf{S}$ and keeps the ones with the largest eigenvectors. A chosen ratio $r = \sum_{k=1}^{q} \lambda_k / \sum_{k=1}^{D} \lambda_k$ of the information contained in the pairwise similarity matrix will be kept for dimensionality reduction, where $\lambda_1, \ldots, \lambda_q$ are the q largest eigenvalues of the covariance matrix of $\mathbf{S}$ in decreasing order.

4.3. Local Outlier Factor for Time sERies (LOFTER)

This section presents an adaptation of LOF referred to as LOFTER for detecting anomalies in ship trajectories. LOFTER evaluates the reachability distances rd_k between all trajectories from the dataset, with one of the similarity measures investigated in Section 3, and determines the LOF anomaly scores. LOFTER then compares these scores to a threshold to detect abnormal ship trajectories.

4.4. Ship anomaly detection based on DBSCAN

This section summarizes two state-of-the-art adaptations of DBSCAN for anomaly detection in ship trajectories investigated in (Zhang and Li, 2022) and (Zhang et al., 2023). These two methods will be used to compare the results obtained in Section 6.1. DBSCAN counts the number of samples in the neighborhood of radius ϵ of each element. As such, when handling trajectories instead of vectors, the DBSCAN algorithm remains the same, provided two trajectories can be compared. A trajectory $\mathbf{X}_j$ is in the neighborhood of the trajectory $\mathbf{X}_i$ if $s(\mathbf{X}_j, \mathbf{X}_i) \leq \epsilon$ where s is one of the similarity measures investigated in Section 3. The similarity measures considered in (Zhang and Li, 2022) and (Zhang et al., 2023) are LCSS and DTW leading to two algorithms referred to as DLCSS (DBSCAN with LCSS) and DDTW (DBSCAN with DTW).

5. Performance metrics

Evaluating the performance of an AD algorithm requires specific metrics since anomalies are rare, which implies imbalanced normal and abnormal classes. This section introduces some metrics that will be used to evaluate the proposed AD algorithms.

Accuracy (Acc) is the standard performance metric for classification. However, consider a dataset with 1% abnormal data, the rest being normal. In this scenario, predicting all samples as inliers would yield an accuracy equal to 99%. Therefore, precision, recall and F1-score are commonly used to evaluate the AD performance. They are defined as:

$$\begin{aligned} &\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \ \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ &\text{F1} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right), \end{aligned} \tag{15}$$

where TP, FP, TN, FN are the numbers of true positives, false positives, true negatives and false negatives.

While the F1-score provides a quantitative measure of the detection performance, precision-recall curves are also used to provide more visual results. The advantages of precision-recall (PR) curves with respect to receiver operating characteristics (ROCs) are explained in (Saito and Rehmsmeier, 2015). While ROCs are reference curves for binary classifiers, changes in AD performance are more apparent in the PR curves.

6. Experiments

This section studies the performance of the proposed AD methods for detecting abnormal ship trajectories in multiple synthetic and real datasets. All methods have been implemented in Python. Execution times were recorded using Python 3.10 and Windows 10 with an Intel i9-10980XE CPU at 3.00 GHz. Section 6.1 considers synthetic data with an available ground truth that are used to compare the proposed methods to the state-of-the-art. Section 6.2 evaluates the performance of the proposed method on real datasets involving AIS and radar trajectories.

6.1. Synthetic data

The first simulations were performed on datasets containing synthetic trajectories initially considered in (Piciarelli et al., 2008). A total of 1000 sets of trajectories were generated with an available ground truth. Each set contains 260 trajectories with 10 anomalies and 250 normal trajectories divided into 5 nominal routes of 50 trajectories. Each trajectory is composed of 16 points described by 2D positions (x_i,y_i) for $i = 1, ..., 16$ that are used to build the matrix $\boldsymbol{X}_i$ of size 2×16 (i.e., $N = 16$). The datasets and the code used for generating the data are publicly available[1]. The algorithms are first tested on these time series referred to as "complete trajectories". In a second step, trajectories of varying lengths are generated to test the robustness of each method to missing data. For trajectories with missing data, a random number of points uniformly distributed in $\{0, \ldots, 5\}$ was removed from each trajectory. This second scenario is referred to as "incomplete trajectories".

The parameters of the proposed algorithms OCTS, SIFTER and LOFTER were chosen as follows:

- OCTS
  - $\nu_{\mathrm{OC}} = 0.12$ has been selected by cross-validation to provide the best detection results,
  - the value of the kernel bandwidth was chosen by maximizing the cost function $J(\sigma^2)$,
  - the performance was computed using 900 sets out of 1000 and the remaining 100 sets were used to set the value of δ. The value of this parameter was determined as the value of δ maximizing the F1-score (see Figs. 3a and 3b).
- SIFTER

[1]https://avires.dimi.uniud.it/papers/trclust/

- $\nu_{\mathrm{IF}} = 0.04 \approx 10/260$, which is the actual proportion of anomalies contained in the synthetic dataset,
- the number of trees has been fixed to 200, which ensures a convergence of the path lengths (a value larger than 100 is recommended in the original paper (Liu et al., 2012)),
- the ratio of information used in PCA was set in order to keep around 10 features. This ratio was chosen by cross validation in order to obtain good AD performance.

- LOFTER
  - $\nu_{\mathrm{LOF}} = 0.04$ as for SIFTER,
  - the 10 nearest neighbors were used to compute densities according to the guidelines of (Breunig et al., 2000),
  - in the experiment DTW-SANS, the number of neighbors was set dynamically for each scenario with the SANS algorithm (Zhu et al., 2016).

The proposed methods are compared to two state-of-the-art algorithms referred to as DDTW and DLCSS (see details in Section 4.4):

- DBSCAN with a DTW similarity measure is used in (Zhang et al., 2023) to detect abnormal trajectories. The hyperparameters used in these experiments are MinPts = 10 (for 5 clusters of 50 trajectories and 10 abnormal trajectories) and $\epsilon = 0.81$ (determined by cross-validation).

- DBSCAN is also used in (Zhang and Li, 2022) with an LCSS similarity measure to detect abnormal trajectories. The hyperparameters used in these experiments are minPts $= 10$, $\epsilon = 0.81$ and $\eta = 0.2$ (to compute the LCSS between two trajectories), which were determined by cross-validation.

Note that methods involving neural networks such as (Nguyen et al., 2018) (Xu et al., 2023) are not evaluated in this work since the volume of training data is too small, e.g. only 1000 sets of 260 trajectories were available for the synthetic data and approximately 600 trajectories for the real data. Note also that the AD detection methods based on feature extraction such as the improved OC-SVM (Sadooghi and Esmaeilzadeh Khadem, 2018) and deep IF

(Xu et al., 2023) are not considered in our comparisons since the proposed AD methods intend to use the time series directly, without any feature extraction step that may be difficult and time consuming.

Examples of predictions and scores obtained for each algorithm on synthetic data are first displayed in Figs. 2a, 2c, 2e and 2b, 2d, 2f. These figures provide qualitative results showing that most outliers (highlighted in red) are detected by the different methods. More quantitative results (allowing the performance of the different methods to be appreciated) are provided in Tables 1 to 8. These results suggest the following comments:

- LOFTER and DBSCAN combined with DTW provide the best predictions with F1-scores close to 95% and the fastest execution times. SIFTER yields an F1-score of approximately 90%, which is close to LOFTER and DBSCAN. Note that OCTS yields an F1-score of 85% with the highest computation time (due to the optimization of J even after finding an optimal value of δ).

- When properly tuned, different AD algorithms often provide similar performance. While this is true for LOFTER and SIFTER, OCTS does not perform well, even after reducing the number of false alarms by increasing the value of δ (using the PR-curve), as observed in Figs. 3a, 3b, 3c, 3d. This may be due to the presence of different clusters, each representing a route. Despite this drawback, OCTS can be used to eliminate redundant data from the training set to speed up computation time. For example, this could be used to build a dictionary of trajectories for SIFTER.

- LOFTER leads to very good detection performance with basic similarity measures, with no clear improvement when using the similarity measures of Section 3.2 (that are not shown here for brevity). When the SANS algorithm is used to set the number of nearest neighbors k, it yields nearly identical performance than optimizing k manually.

- SIFTER performs better when a PCA preprocessing is applied due to the curse of dimensionality. In all cases, wrong classifications persist as certain trajectories lie on the edge of the normal / abnormal frontier making prediction challenging. These wrong detections are clearly seen in Fig. 2c, where the false positives and false negatives are both at the edge of a cluster.

- After comparing the different similarity measures, we can observe that DTW outperforms the other similarity measures for time series (Badiane and Cunningham, 2021). Furthermore, performing AD with DTW only required minutes (see Table 5) while TWED took up to 6 hours depending on the method used.
- Tables 3 and 4 show that LOFTER and DBSCAN perform very similarly. Moreover, the proposed methods remain competitive when compared to the state-of-the-art.
- The conclusions for incomplete data are similar (see Tables 1, 6, 7, 8), with a slight loss in performance when compared to the results obtained with complete data.

Table 1: OCTS Performance for complete and incomplete data.

	Precision	Recall	F1	Acc
κ_{DTW} $\delta = 0$ (%)	29.33	91.77	44.43	91.15
κ_{DTW} $\delta > 0$ (%)	94.82	79.20	85.44	99.01
κ_{DTW} $\delta > 0$ missing data (%)	91.19	76.02	81.92	98.76
κ_{TWED} $\delta > 0$ (%)	95.57	78.06	85.14	99.00

Table 2: SIFTER Performance for complete data.

	Precision	Recall	F1	Acc
DTW (%)	70.28	77.31	73.62	97.87
sim_{DTW} (%)	72.89	80.98	76.36	98.09
$\cos_{\text{DTW}}$ (%)	78.84	86.72	82.59	98.59
GK_{DTW} (%)	62.32	68.55	65.29	97.20
TWED (%)	71.00	78.10	74.38	97.93
sim_{TWED} (%)	63.64	70.00	66.67	97.31
$\cos_{\text{TWED}}$ (%)	75.46	83.01	79.06	98.31
GK_{TWED} (%)	44.95	49.44	47.09	95.73
DTW PCA (%)	**86.41**	**95.04**	**90.52**	**99.23**
sim_{DTW} PCA (%)	81.44	89.58	85.31	98.81
$\cos_{\text{DTW}}$ PCA (%)	81.54	89.69	85.42	98.82
GK_{DTW} PCA (%)	85.67	94.24	89.75	99.17
TWED PCA (%)	80.68	88.75	84.52	98.75
sim_{TWED} PCA (%)	84.66	93.13	88.70	99.09
$\cos_{\text{TWED}}$ PCA (%)	80.55	88.60	84.38	98.74
GK_{TWED} PCA (%)	81.73	89.90	85.62	98.84

Table 3: LOFTER Performance for complete data.

	Precision	Recall	F1	Acc
DTW (%)	96.42	**96.60**	**96.44**	**99.73**
DTW-SANS (%)	**96.88**	96.17	96.42	99.73
TWED (%)	95.82	95.31	95.50	99.66

Table 4: Performance of algorithms from the state-of-the-art (Laxhammar and Falkman, 2014), (Mangé et al., 2023), (Zhang and Li, 2022) and (Zhang et al., 2023) for complete trajectories.

	Precision	Recall	F1	Acc
SHNN-CAD (%)	–	–	–	97.09
OCSVM (%)	58.78	24.63	31.82	96.17
DLCSS (%)	96.20	97.04	96.45	99.72
DDTW (%)	91.01	99.89	94.99	99.65

Table 5: Execution times of AD algorithms with DTW (complete data).

	OCTS	SIFTER	LOFTER	DDTW
Time (s)	1105	761	272	275

Table 6: Metrics from the state-of-the-art (Mangé et al., 2023), (Zhang and Li, 2022) and (Zhang et al., 2023) for incomplete trajectories.

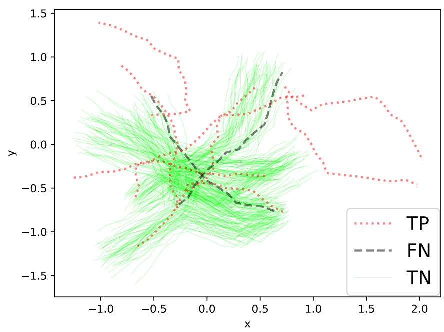
	Precision	Recall	F1	Acc
DLCSS (%)	96.83	94.59	95.49	99.66
DDTW (%)	99.90	89.22	93.95	99.58

Table 7: SIFTER Performance for incomplete data.

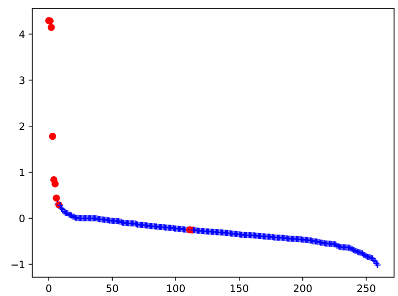
	Precision	Recall	F1	Acc
DTW (%)	60.77	66.85	63.67	97.07
sim_{DTW} (%)	64.18	70.60	67.24	97.35
$\cos_{\text{DTW}}$ (%)	76.04	83.64	79.66	98.36
GK_{DTW} (%)	54.83	60.31	57.44	96.56
TWED (%)	43.06	47.37	45.11	95.57
sim_{TWED} (%)	66.75	73.43	69.93	97.57
$\cos_{\text{TWED}}$ (%)	68.03	74.83	71.27	97.68
GK_{TWED} (%)	31.36	34.50	32.86	94.58
DTW PCA (%)	84.38	92.82	88.40	99.06
sim_{DTW} PCA (%)	67.87	74.66	71.10	97.67
$\cos_{\text{DTW}}$ PCA (%)	74.57	82.03	78.12	98.23
$\mathbf{GK}_{\mathbf{DTW}}$ PCA (%)	**85.65**	**94.21**	**89.73**	**99.17**
TWED PCA (%)	51.91	57.10	54.38	96.32
sim_{TWED} PCA (%)	80.97	89.07	84.83	98.77
$\cos_{\text{TWED}}$ PCA (%)	68.78	75.66	72.06	97.74
GK_{TWED} PCA (%)	61.65	67.81	64.58	97.14

Table 8: LOFTER Performance for incomplete data.

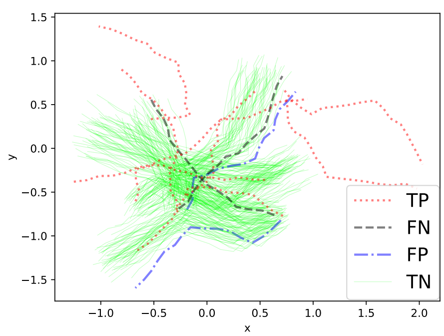
	Precision	Recall	F1	Acc
DTW (%)	95.88	**96.22**	**95.98**	**99.69**
DTW-SANS (%)	**95.95**	95.93	95.87	99.68
TWED (%)	93.81	93.00	93.32	99.49



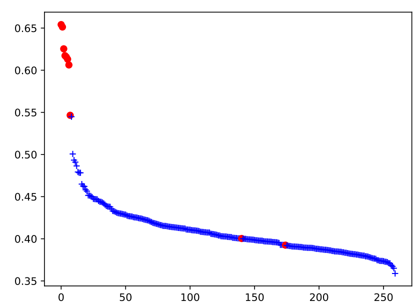
(a) OCTS predictions.



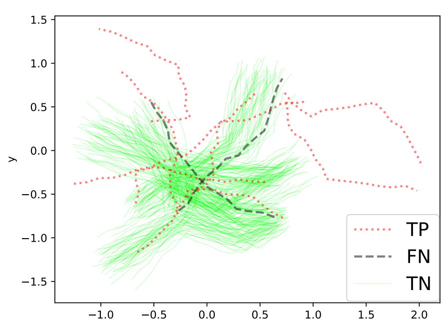
(b) OCTS scores.



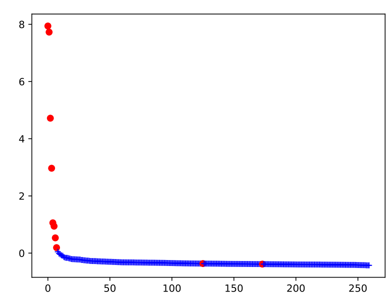
(c) SIFTER predictions.



(d) SIFTER scores.

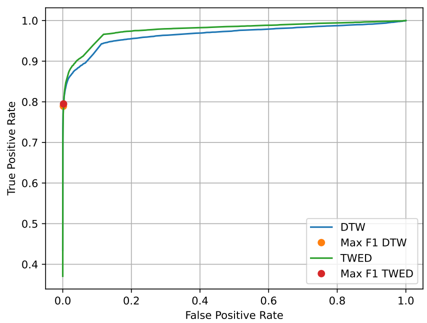


(e) LOFTER predictions.

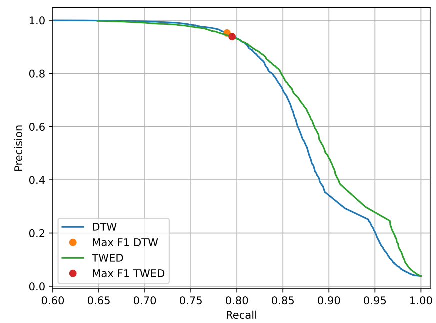


(f) LOFTER scores.

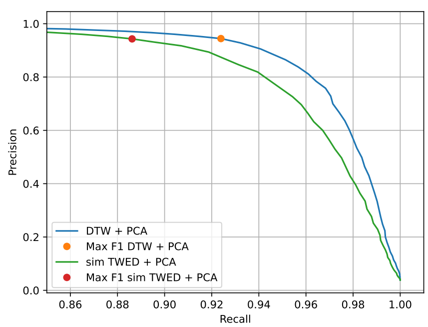
Figure 2: Examples of synthetic trajectories with their predictions and anomaly scores (anomalies are in red).



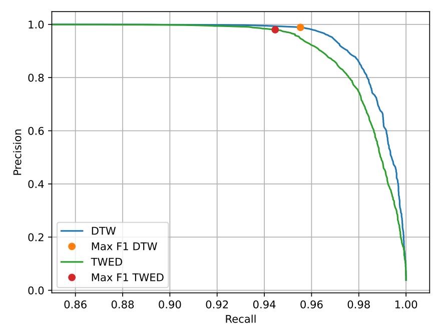
(a) OCTS, ROC curves with 10% of the complete datasets for multiple values of δ.



(b) OCTS, PR curves obtained using 10% of the datasets for multiple values of δ.



(c) SIFTER, PR curves.



(d) LOFTER, PR curves.

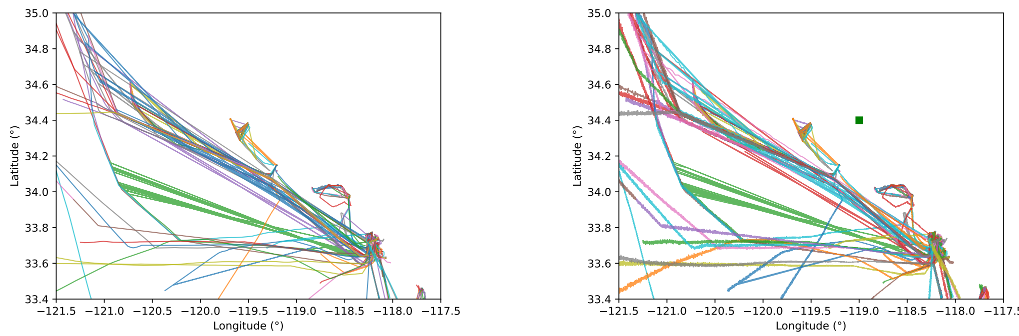
Figure 3: Performance curves for the algorithms applied to the complete synthetic dataset.

6.2. Real data

This section briefly recaps the relevant properties of Automatic Identification System (AIS) data before introducing the real dataset and showing the results obtained with the proposed AD methods.

6.2.1. Automatic Identification System

AIS has been introduced to avoid collision between ships (Silveira et al., 2013). It is a cooperative system, meaning ships themselves choose to broadcast their information. Based on the GPS for positioning, when it is not falsified, this system provides a wealth of both dynamic information such as position, speed, heading and static information like the ship name, identifier, dimensions, type and destination. AIS also benefits from the accuracy of the



(a) AIS dataset from MarineCadastre.

(b) Simulated radar data. The green square is the radar position.

Figure 4: AIS and radar trajectories for the dataset California.

GPS system with a maximum error of 10 meters on the true ship positions. Therefore, depending on the information taken into account, different behaviors in certain contexts can be examined and other anomalies detected. For example, if only positions are taken into account, a ship will behave abnormally if it is not on a sea rail. By adding speed and heading, a ship may behave abnormally on a rail if it is not traveling at the right speed or in the wrong direction.

6.2.2. Dataset description

Many AIS datasets are publicly available[2]. The data used in this section contains trajectories acquired near the coast of California[3]. A first set of real data was collected during the month of January 2021 over the course of 20 days. Fig. 4a shows 600 ship trajectories extracted from the California dataset. One can observe that the traffic is dense near ports and that there are two parallel routes established along the coast. As such, few anomalies should be expected in these areas. Furthermore, there are some discontinuities in a few ship trajectories. These discontinuities are probably due to a biased or turned off AIS and should clearly be considered as an anomaly. No ground truth is available for this real dataset. Thus the AD performance is only evaluated qualitatively in this section.

[2] www.marinetraffic.com/en/ais/home/

[3] https://marinecadastre.gov/accessais/

6.2.3. Performance and discussion

OCTS, SIFTER and LOFTER are first considered to detect abnormal AIS trajectories from the real dataset. The positions, speeds and courses over ground of the different ships from this dataset are used to build the matrices $\boldsymbol{X}_i$, for $i = 1, ..., 600$. Note that the columns of these matrices have generally different lengths since the AIS messages are acquired at different instants and the AIS message may be interrupted in some cases. Note also that the positions, speeds and courses over ground have been normalized since they do not have the same ranges. The values of the hyperparameters used in the AD algorithms for these experiments are specified below:

- OCTS: $\nu_{\text{OC}} = 0.1$ and $\delta = 0$ were adjusted manually to detect 10% of anomalies in the dataset. Based on the results obtained with synthetic data, dynamic time warping was used as the similarity measure defining the kernel. The value of the kernel bandwidth was chosen by maximizing the cost function $J(\sigma^2)$.

- SIFTER: The number of trees used in IF was fixed to 200 (as for synthetic data) and the proportion of anomalies was fixed to $\nu_{\text{IF}} = 0.1$. The algorithm was run with a PCA preprocessing ensuring that all the detected anomalies are confirmed leading to $r = 99.9\%$.

- LOFTER: The proportion of anomalies was fixed to $\nu_{\text{LOF}} = 0.1$ and the number of nearest neighbors was calculated using the SANS algorithm to compute the anomaly scores, as suggested in (Zhu et al., 2016).

The anomalies detected by the different algorithms are displayed in red in Fig. 5 whereas the normal trajectories are shown in green. Our conclusions are summarized below:

- The anomalies detected by OCTS are isolated trajectories located at the bottom of the figure and spoofed trajectories located in the top rail. Spoofed trajectories can be identified by positions varying abruptly from one time instant to another. The limited performance of OCTS for detecting some spatial anomalies (e.g. the green trajectories located in the bottom of the figure) can be explained by its limited capability of detecting outliers. As explained in the scikit-learn webpage dedicated to anomaly detection[4], "One-class SVM is known to be sensitive

[4]https://scikit-learn.org/stable/modules/outlier_detection.html

to outliers and thus does not perform always very well for outlier detection". We have also observed that the variability of the OCTS scores computed for all trajectories is more limited than for SIFTER and LOFTER, i.e., the algorithm hesitates between labelling the trajectories as normal or abnormal (see Fig. 5a).

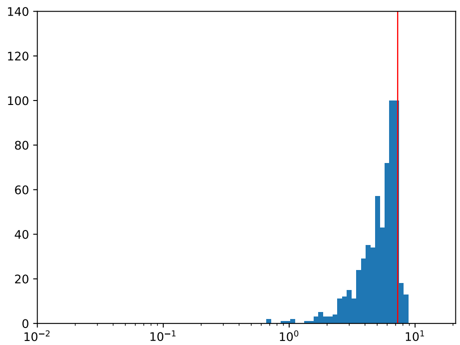
- SIFTER labels many isolated trajectories located at the bottom of the figure as "normal". Moreover, many trajectories located in the rail are detected as anomalies (note that there are only 60 trajectories in this rail and that most of them have been spoofed with abrupt jumps between consecutive locations). Thus, SIFTER seems to concentrate on anomalies associated with spoofed trajectories, when compared to OCTS and LOFTER. For this example, IF tends to define a larger frontier around normal data than the other algorithms, which can also be observed in the datasets investigated in the scikit-learn webpage dedicated to anomaly detection.

- The anomaly scores returned by LOFTER are very high for all isolated trajectories located at the bottom of the figure that are detected as anomalies. The price to pay is that some spoofed trajectories located in the middle of the figure (green lines) are detected as normal. Thus, LOFTER seems to concentrate on spatial anomalies in AIS messages. This is confirmed by the two trajectories containing missing and falsified data displayed in the zoomed in of Fig. 5f that are not detected as anomalies. Note that the SANS algorithm returns around $k = 20$ neighbors for this dataset. Since there are less than 20 trajectories at the bottom of the figure, they are all labeled as anomalies.

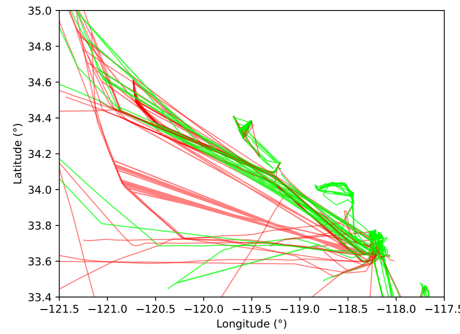
To conclude this discussion, OCTS seems to provide a good compromise for detecting abnormal AIS trajectories. SIFTER and LOFTER complement each other, since SIFTER focuses on spoofed trajectories and LOFTER on isolated trajectories.

6.2.4. AIS and radar

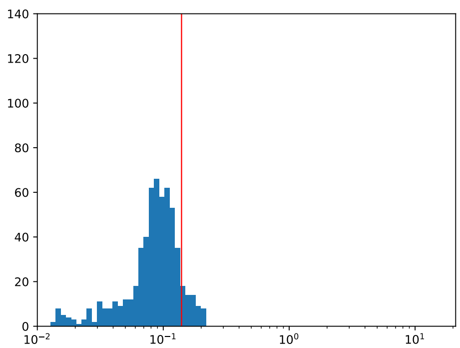
This section studies the utility of the proposed AD algorithms for the detection of abnormal ship behaviors using jointly AIS and radar data. The interest in using radar data (that are less accurate than AIS data) is that all ships in the range of the radar can be detected without being falsified and that the AIS data may not be available on specific time intervals. This explains



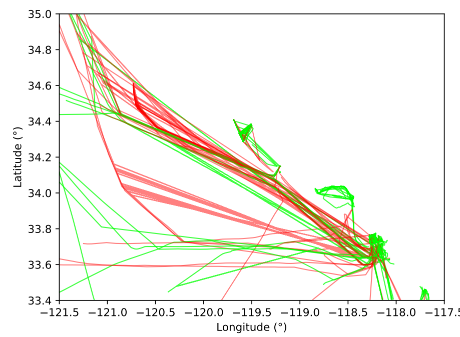
(a) OCTS scores.



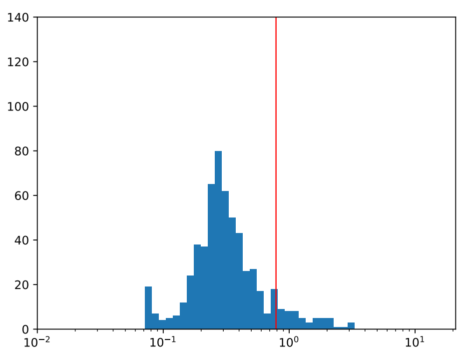
(b) OCTS predictions.



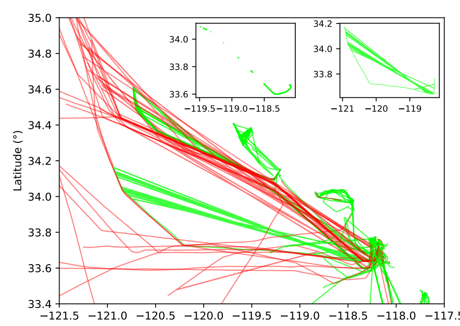
(c) SIFTER scores.



(d) SIFTER predictions.



(e) LOFTER scores.



(f) LOFTER predictions. In the top right corner, an AIS trajectory with missing data (left zoom) and another one with falsified data (right zoom) have been displayed. They have been labeled as normal by LOFTER.

Figure 5: Performance of the algorithms for the AIS dataset. The histograms of scores are displayed in the left figure (log-scale) with the threshold used to detect anomalies in red. The detection results are shown in the right figure with normal trajectories in green and abnormal trajectories in red.

why these two sensors are often used for maritime surveillance (Guerriero et al., 2008). Provided that the association between the two sensors has been performed, the complementarity between AIS and radar data is known to be interesting for monitoring vessels on the sea (Yang et al., 2022).

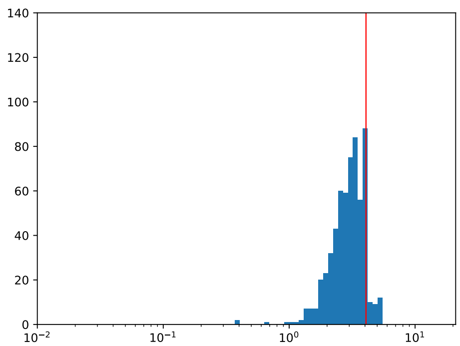
In order to analyze the interest of using AIS and radar data jointly, radar tracks associated with the AIS trajectories described in Section 6.2.2 were simulated as follows: zero-mean Gaussian noise was added to the interpolated AIS tracks in range and azimuth. The noise standard deviations for the range and azimuth noises were fixed to $\sigma_r = 25.51$ meters and $\sigma_\theta = 0.1531$ degrees (°), corresponding to 50 meters and 0.3° with a confidence of 95%. The interpolation was conducted using linear interpolation leading to radar data acquired every 10 seconds. The simulated radar sensor is located at a latitude of 34.4° and a longitude of $-119°$ and is represented as a green square in Fig. 4b. AD was then performed on pairs of AIS and radar features for each ship. The features used in these experiments are the position, speed and heading of each ship given by the AIS and the simulated radar. Note that the AIS / radar association was considered to be known in these experiments.

The joint use of AIS and radar data requires the definition of an appropriate similarity measure to be used in the proposed AD methods. A linear combination of two distances with positive coefficients is a distance. Consequently, $s[(\mathbf{U}_a, \mathbf{U}_r), (\mathbf{T}_a, \mathbf{T}_r)] = (1-\lambda)s_a(\mathbf{U}_a, \mathbf{T}_a) + \lambda s_r(\mathbf{U}_r, \mathbf{T}_r)$ defines a joint similarity measure adapted to AIS and radar data, where $0 \leq \lambda \leq 1$ is the weight of the radar similarity measure and $\mathbf{U}_a$, $\mathbf{T}_a$, $\mathbf{U}_r$ and $\mathbf{T}_r$ are the AIS and radar matrices associated with a given ship. The experiments have been conducted with $\lambda = 0.5$ meaning that the AIS and radar features have the same weight in $s[(\mathbf{U}_a, \mathbf{U}_r), (\mathbf{T}_a, \mathbf{T}_r)]$. It should be noted that DTW may be sensitive to the number of points in the compared trajectories. Since the radar has a higher sampling frequency, s_r tends to be higher than s_a. In order to avoid this problem, the hyperparameter λ might also be adjusted to avoid penalizing long trajectories.

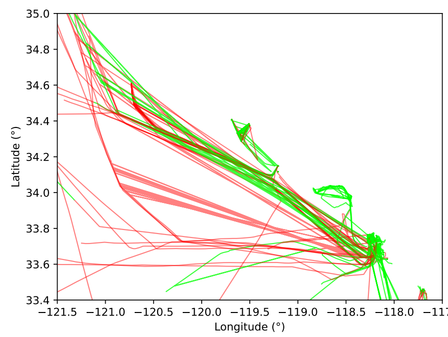
The interest of jointly using AIS and radar data is evaluated qualitatively in Fig. 6. Our conclusions are summarized below:

- The anomalies detected by OCTS have not changed significantly when compared to the AIS only scenario, except for some additional isolated trajectories that have been detected as anomalies in Figs. 6a and 6b. OCTS still performs quite well in this scenario.

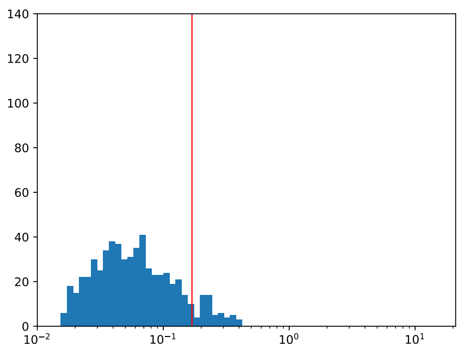
- In the AIS only case, SIFTER seemed to focus on detecting spoofed



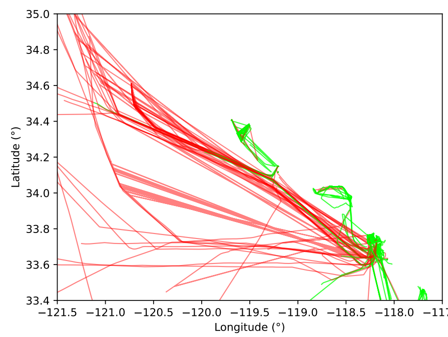
(a) OCTS scores.



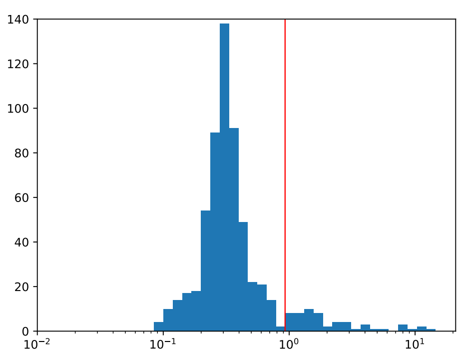
(b) OCTS predictions.



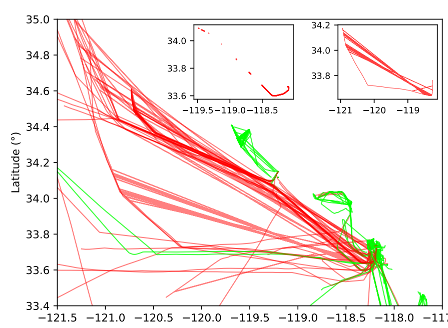
(c) SIFTER scores.



(d) SIFTER predictions.



(e) LOFTER scores.



(f) LOFTER predictions. In the top right corner, an AIS trajectory with missing data (left zoom) and another one with falsified data (right zoom) have been displayed. They have been labeled as abnormal by LOFTER.

Figure 6: Performance of the algorithms for AIS and radar datasets. The histograms of scores are displayed in the left figure (log-scale) with the threshold for anomaly detection in red. The detection results are shown in the right figure with normal trajectories in green and abnormal trajectories in red.

trajectories. When combining AIS and radar data, SIFTER scores have increased even for spoofed trajectories, as displayed in Fig. 6c, and the distinction between normal data and anomalies is clearer (the histogram of anomaly scores shows two modes corresponding to normal and abnormal trajectories). As a result, some additional isolated trajectories have been detected when compared to the AIS only scenario, which is very interesting.

- Adding radar data to LOFTER has helped detecting spoofed AIS while still identifying some ships outside nominal maritime routes, as shown in Fig. 6f. Fig. 6f also shows that the trajectory containing missing data (left) and the abnormal trajectory (right) have been detected as anomalies thanks to the presence of radar data. Finally, it is interesting to note that Fig. 6e shows that the scores associated with normal and abnormal trajectories differ significantly, highlighting the interest of using radar as a complement to AIS.

To conclude, in scenarios where AIS and radar data are available, we recommend the use of OCTS for detecting anomalies in ship trajectories. LOFTER and SIFTER still have some merits for detecting spoofed or shut down trajectories.

7. Conclusion

This paper has investigated new anomaly detection methods for detecting abnormal ship trajectories for maritime surveillance. These new methods were obtained by injecting similarity measures adapted to ship trajectories in state-of-the-art anomaly detection algorithms, namely One-Class SVM, Isolation Forest and Local Outlier Factor. The resulting methods referred to as OCTS, SIFTER and LOFTER are fully unsupervised. They were compared and evaluated with multiple similarity measures using synthetic and real AIS data. The proposed methods can also be applied to a combination of AIS and radar data, which is interesting when the AIS data is spoofed or not available. This interest was confirmed by showing examples of anomalies detected by the combination AIS/radar and not by AIS only.

Our conclusion is that dynamic time warping generally provides very competitive results with respect to the other similarity measures. Regarding the performance of anomaly detection algorithms, LOFTER provided the best results when applied to synthetic datasets with a controlled ground truth and

OCTS performed well for real datasets. SIFTER also has valuable qualities, such as discrete features. Thus we think that the three proposed algorithms have complementary properties and should be considered in practical applications, offering the possibility to detect different abnormal ship behaviors. Note finally that OCTS can be generalized to a semi-supervised scenario, allowing user feedback to be considered (Lesouple et al., 2021).

Future work will be devoted to finding methods allowing the hyperparameters to be determined automatically from the data, and taking into account the ship type, since fishing vessels, cargos or sailing boats have different behaviors depending on their class. It would also be interesting to evaluate the performance of the proposed algorithms to AIS and radar data obtained from realistic maritime conditions, even if it is not easy to obtain ground truth data for these applications. Another research area is performing an accurate association between data resulting from different sensors such as AIS, radar and electronic support measures (ESM). Finally, locating anomalies in specific parts of suspicious trajectories would be valuable, rather than considering the entire trajectories. This could be performed using trajectory segmentation algorithms such as the Ramer-Douglas-Peucker algorithm (Douglas and Peucker, 1973).

Credit authorship contribution statement

Valérian Mangé: Conceptualization, Data curation, Methodology, Visualization, Writing – original draft, Writing – review and editing.
Jean-Yves Tourneret: Conceptualization, Funding acquisition, Methodology, Supervision, Validation, Writing – original draft, Writing – review and editing.
François Vincent: Conceptualization, Methodology, Supervision, Validation, Writing – original draft, Writing – review and editing.
Laurent Mirambell: Funding acquisition, Project administration, Resources, Validation.
Fábio Manzoni Vieira: Funding acquisition, Supervision, Resources, Validation

Acknowledgment

Research supported by the French Army Ministry of Defense - Agence de l'innovation de défense.

References

Aggarwal, C.C., 2016. Outlier Analysis. Springer Verlag, New York.

Androjna, A., Perkovic, M., Pavic, I., Miskovic, J., 2021. AIS data vulnerability indicated by a spoofing case-study. Applied Sciences 11, 5015.

Arcudi, A., Frizzo, D., Masiero, C., Susto, G., 2024. Enhancing interpretability and generalizability in extended isolation forests. Engineering Applications of Artificial Intelligence 138, 109409.

Badiane, M., Cunningham, P., 2021. An empirical evaluation of kernels for time series. Artificial Intelligence Review 55, 1803–1820.

Bono, F., Radicioni, L., Cinquemani, S., 2023. A novel approach for quality control of automated production lines working under highly inconsistent conditions. Engineering Applications of Artificial Intelligence 122, 106149.

Breunig, M., Kröger, P., Ng, R., Sander, J., 2000. LOF: Identifying density-based local outliers. ACM Sigmod Record 29, 93–104.

Campello, R., Moulavi, D., Sander, J., 2013. Density-based clustering based on hierarchical density estimates, in: Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining, Gold Coast, Australia. pp. 160–172.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Computing Surveys 41, 1–72.

Decoste, D., Schölkopf, B., 2002. Training invariant support vector machines. Machine Learning 46, 161–190.

Douglas, D., Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization 10, 112–122.

Entezami, A., Sarmadi, H., Behkamal, B., 2023. A novel double-hybrid learning method for modal frequency-based damage assessment of bridge structures under different environmental variation patterns. Mechanical Systems and Signal Processing 201, 110676.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proc. Second Int. Conf. on Knowledge Discovery and Data Mining, Portland, USA. pp. 226–231.

Ghiasi, R., Khan, M., Sorrentino, D., Diaine, C., Malekjafarian, A., 2024. An unsupervised anomaly detection framework for onboard monitoring of railway track geometrical defects using one-class support vector machine. Engineering Applications of Artificial Intelligence 133, 108167.

Guerriero, M., Willett, P., Coraluppi, S., Carthel, C., 2008. Radar/AIS data fusion and SAR tasking for maritime surveillance, in: Proc. Int. Conf. Information Fusion (FUSION), Cologne, Germany. pp. 1–5.

Honarkhah, M., Caers, J.K., 2010. Stochastic simulation of patterns using distance-based pattern modeling. Mathematical Geosciences 42, 487–517.

Jolliffe, I., 2002. Principal Component Analysis. Springer Series in Statistics, Springer.

Keogh, E., Ratanamahatana, C., 2005. Exact indexing of dynamic time warping. Knowl. Inf. Syst. 7, 358–386.

Kiwi, M., Loebl, M., Matousek, J., 2005. Expected length of the longest common subsequence for large alphabets. Advances in Mathematics 197, 480–498.

Laxhammar, R., Falkman, G., 2014. Online learning and sequential anomaly detection in trajectories. IEEE Trans. Pattern Anal. Mach. Intell. 36, 1158–1173.

Lesouple, J., Baudoin, C., Spigai, M., Tourneret, J.Y., 2021. How to introduce expert feedback in one-class support vector machines for anomaly detection? Signal Processing 188, 108197.

Li, H., Li, W., Wang, S., Yang, H., Guan, J., Zhang, Y., 2024. STAD: Ship trajectory anomaly detection in ocean with dynamic pattern clustering. Ocean Engineering 313, 119530.

Liu, F., Ting, K., Zhou, Z., 2012. Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data 6, 1–39.

Mangé, V., Tourneret, J., Vincent, F., Mirambell, L., Manzoni Vieira, F., Pilastre, B., 2023. Détection de comportements anormaux dans des trajectoires de navires avec one-class SVM et dynamic time warping, in: Proc. Conf. GRETSI, Grenoble, France. pp. 1–4.

Marteau, P., 2009. Time warp edit distance with stiffness adjustment for time series matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 306–318.

Nguyen, D., Vadaine, R., Hajduch, G., Garello, R., Fablet, R., 2018. A multi-task deep learning architecture for maritime surveillance using AIS data streams, in: Proc. 2018 IEEE 5th Int. Conf. on Data Science and Advanced Analytics (DSAA), Turin, Italy. pp. 331–340.

Pallotta, G., Vespe, M., Bryan, K., 2013. Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. Entropy 15, 2218–2245.

Pang, G., Shen, C., Cao, L., Hengel, A., 2021. Deep learning for anomaly detection: A review. ACM Computing Surveys 54.

Piciarelli, C., Micheloni, C., Foresti, G.L., 2008. Trajectory-based anomalous event detection. IEEE Trans. Circuits Sys. Video Technol. 18, 1544–1554.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., Jurafsky, D., 2004. Shallow semantic parsing using support vector machines, in: Proc. Human Language Technology Conf. of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, Boston, USA. pp. 233–240.

Puggini, L., McLoone, S., 2018. An enhanced variable selection and isolation forest based methodology for anomaly detection with OES data. Engineering Applications of Artificial Intelligence 67, 126–135.

Qian, C., He, N., He, L., Li, R., Cheng, F., 2024. A weighted Gaussian process regression model based on improved local outlier factor and its application in state of health estimation of lithium-ion battery. Engineering Applications of Artificial Intelligence 138, 109314.

Sadooghi, M., Esmaeilzadeh Khadem, S., 2018. Improving one class support vector machine novelty detection scheme using nonlinear features. Pattern Recognition 83, 14–33.

Saito, T., Rehmsmeier, M., 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS One 10, 1–21.

Sartor, D., Barbariol, T., Susto, G., 2024. Bayesian active learning isolation forest (B-ALIF): A weakly supervised strategy for anomaly detection. Engineering Applications of Artificial Intelligence 130, 107671.

Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B., 2011. Finding a "kneedle" in a haystack: Detecting knee points in system behavior, in: Proc. Int. Conf. on Distributed Computing Systems Workshops, Minneapolis, USA. pp. 166–171.

Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. Neural computation 13, 1443–1471.

Sidorov, G., Gelbukh, A., Gómez Adorno, H., Pinto, D., 2014. Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas 18, 491–504.

Silveira, P., Teixeira, A., Guedes Soares, C., 2013. Use of AIS data to characterise marine traffic patterns and ship collision risk off the coast of Portugal. Journal of Navigation 66, 879–898.

Staerman, G., Mozharovskyi, P., Clémençon, S., d'Alché Buc, F., 2019. Functional isolation forest, in: Proc. Asian Conf. on Machine Learning, Nagoya, Japan. pp. 332–347.

Trinh, V.V., Tran, K.P., Huong, T.T., 2017. Data driven hyperparameter optimization of one-class support vector machines for anomaly detection in wireless sensor networks, in: Proc. Int. Conf. Advanced Tech. for Commun. (ATC), Quy Nhon, Vietnam. pp. 6–10.

Wolsing, K., Roepert, L., Bauer, J., Wehrle, K., 2022. Anomaly detection in maritime AIS tracks: A review of recent approaches. Journal of Marine Science and Engineering 10, 1–19.

Xie, Z., Bai, X., Xu, X., Xiao, Y., 2024. An anomaly detection method based on ship behavior trajectory. Ocean Engineering 293, 116640.

Xu, H., Pang, G., Wang, Y., Wang, Y., 2023. Deep isolation forest for anomaly detection. IEEE Transactions on Knowledge and Data Engineering 35, 12591–12604.

Yang, Y., Yang, F., Sun, L., Xiang, T., Lv, P., 2022. Multi-target association algorithm of AIS-radar tracks using graph matching-based deep neural network. Ocean Engineering 266, 112208.

Zhang, C., Liu, S., Guo, M., Liu, Y., 2023. A novel ship trajectory clustering analysis and anomaly detection method based on AIS data. Ocean Engineering 288, 116082.

Zhang, Y., Li, W., 2022. Dynamic maritime traffic pattern recognition with online cleaning, compression, partition, and clustering of AIS data. Sensors 22, 1–18.

Zhao, Y., Huang, G., Hu, Q., Li, B., 2020. An improved weighted one class support vector machine for turboshaft engine fault detection. Engineering Applications of Artificial Intelligence 94, 103796.

Zhu, Q., Feng, J., Huang, J., 2016. Natural neighbor: A self-adaptive neighborhood method without parameter K. Pattern Recognition Letters 80, 30–36.