Applied Ocean Research 158 (2025) 104552

Contents lists available at ScienceDirect

Applied Ocean Research

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

Research paper

LiDAR target detection and classification for ship situational awareness: A hybrid learning approach

Filippo Ponzini *, Raphael Zaccone, Michele Martelli

Marine, Electrical, Electronics and Telecommunication Engineering and Naval Architecture Department (DITEN), Polytechnic School, University of Genoa, Via Montallegro 1, Genoa, 16145, Italy

ARTICLE INFO

Keywords:
Marine Autonomous Surface Ships
Marine situational awareness
Obstacle detection
Obstacle classification
LiDAR
Navigation system

ABSTRACT

In recent years, LiDARs have been used to enhance situational awareness of autonomous vehicles, including in the marine domain, driven by the need for reliable detections in Marine Autonomous Surface Ships and Unmanned Surface Vehicles. Detecting obstacles and targets within point clouds is generally handled by a fully unsupervised learning framework. While effective and simple, this approach cannot classify targets. This paper presents a combined unsupervised/supervised approach for detecting and classifying marine targets and obstacles. The unsupervised detection framework is maintained by incorporating a lightweight supervised module capable of classifying detection outputs without disrupting the workflow. Rather than training on the entire point cloud, the proposed method focuses on selected target features, reducing model size and information exchange. Specifically, a Random Forest Classifier is trained on features extracted from the point-cloud dataset. The acquisition of an ad-hoc training dataset and its statistical analysis are presented to identify key features. The selection, training, and validation processes are outlined. Finally, the supervised model is integrated into a state-of-the-art unsupervised LiDAR detection pipeline and tested in a real scenario. The results demonstrate the hybrid framework's effectiveness and compliance with real-time constraints.

1. Introduction

The autonomous vehicles research field is growing year by year in all domains. In the marine field, Marine Autonomous Surface Ships are effectively used to carry out a wide range of missions, including autonomous transportation, surveillance and mapping, scientific data acquisition, Search And Rescue (SAR), and Disaster Management (DM) up to strategic and tactical missions. During operations, Autonomous Surface Vehicles require Situational Awareness (SA) of the surrounding environment, detecting and classifying any obstacles and targets; this aspect is crucial to achieving safe navigation and efficient operation. Several sensors are used individually or embedded in a multi-sensor data fusion framework to fulfil this requirement (Thombre et al., 2022). Many applications involve the use of Automatic Identification System (AIS) data for safe navigation (Zhao and Fu, 2021), or mission path planning is handled by representing obstacles as an occupancy grid (Zaccone, 2024). Often, the autonomous vehicle needs to detect and classify obstacles to make decisions according to the detected target type. AIS data solves this problem by providing a location and a target classification (Huang et al., 2024). However, obstacles encountered during the mission may be AIS-unequipped as in the case of natural obstacles, small vessels, or may intentionally keep the AIS transponder switched off as in the case of dark ships phenomenon (Nguyen, 2023). Consequently, a sensing layer capable of detecting and classifying obstacles is required. The typical choice for ensuring this requirement is to use Radio Detection And Ranging (RADAR), Light Detection And Ranging (LiDAR) and optical sensors, or a combination of them. Such sensors detect surrounding obstacles regardless of whether they are equipped with or without AIS, provide a reasonably accurate estimate of their position relative to the observing vehicle and, possibly, a classification of the target.

2. Related work

In this section, the works available in literature regarding detection and classification focused on navigation in the marine environment are analysed. The capabilities of LiDAR sensors are demonstrated by relevant environment tests in both coastal and blue-water scenarios (Faggioni et al., 2022b), using a detection pipeline based entirely on unsupervised learning, producing accurate detection but no target classification. A RADAR detection framework is presented in Ha et al. (2021), constructing an occupation grid without semantic information.

* Corresponding author.
E-mail address: filippo.ponzini@edu.unige.it (F. Ponzini).

https://doi.org/10.1016/j.apor.2025.104552
Received 30 October 2024; Received in revised form 21 February 2025; Accepted 27 March 2025
Available online 9 April 2025

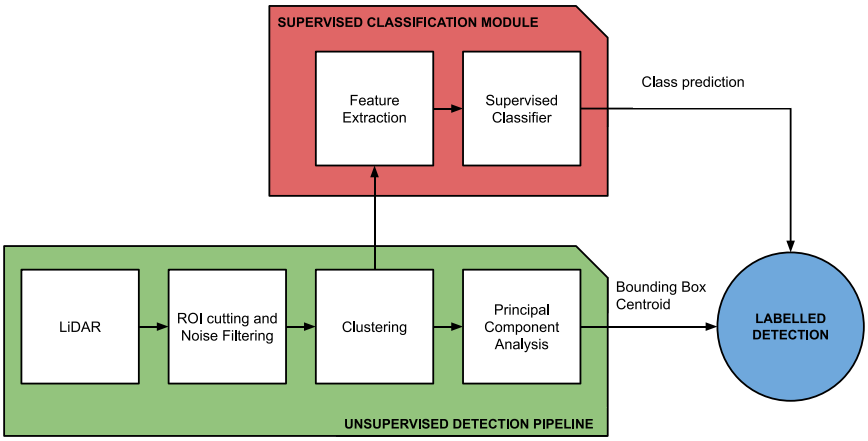


Fig. 1. Supervised classifier module within an unsupervised LiDAR detection pipeline.

A feasibility study on automotive-grade RADAR for marine target localization is discussed in Gennarelli et al. (2022), while Ludeno et al. (2023) explores the use of a lightweight RADAR system to enhance situational awareness through the detection and tracking of marine targets, along with monitoring sea state. The advantages of marine obstacle detection by optical sensors using semantic segmentation are presented in Bovcon and Kristan (2022), providing accurate detection and classification. A detection pipeline based on LiDAR and RGB camera is presented in Faggioni et al. (2022a), classifying the target from the processing of optical sensor data. A monocular Camera, a LiDAR, and a marine RADAR are combined in Clunie et al. (2021) obtaining a marine obstacle detection framework where target classification is still entrusted to optical sensor processing. A multi-sensor tracking system based on RADAR, LiDAR, infrared and electro-optical sensors is presented in Helgesen et al. (2022) with a subsequent open dataset. An attempt to obtain a classification from LiDAR data is presented in Stanislas and Dunbabin (2019), combining the result with the classification obtained from optical sensors. LiDAR plays a detection-only role in the presented applications, or it is subordinate to the classification obtained by optical sensors.

Extending the field of research to the automotive sector, classification applications using LiDARs are more frequent. Deep learning algorithms are capable of analysing LiDAR point clouds, and models such as PointNet (Charles et al., 2017), PointNet++ (Qi et al., 2017), CentrePoint (Yin et al., 2021), Pointpillars (Lang et al., 2019) are successfully employed for this scope. Such models require huge datasets of labelled examples, which are currently absent or inadequate in the marine field, and enormous training resources.

Currently, the LiDAR marine detection approach is entirely based on clustering-based detection or occupancy grid methods, leaving the classification task to the optical sensors. Fig. 1 shows a general LiDAR clustering-based detection pipeline. In particular, after noise filtering and a Region Of Interest (ROI) cutting, the point-cloud is grouped into a set of clusters via a Clustering Algorithm; at this stage, points that constitute an obstacle or a target for the algorithm cluster definition are grouped and separated by the rest of the point-cloud. A typical choice for this purpose is a simple Euclidean distance-based clustering algorithm or a density-based one such as DBSCAN (Schubert et al., 2017) or its hierarchic version HDBSCAN (Campello et al., 2013). After the cluster analysis, a bounding box is fitted on each cluster obtaining its main dimensions and centroid. The advantages of using Principal Component Analysis (PCA) for this scope are presented in Faggioni et al. (2022b).

The pipeline based on clustering is currently one of the most used, allowing for several advantages such as simplicity, generality of the algorithm, no need for training, real-time compliance and robustness. This paper proposes a solution for the LiDAR classification gap of marine obstacles without recourse on heavy detection models and retaining the advantages of the state-of-the-art unsupervised structure. Hence, an independent supervised learning module is proposed to classify the obtained clusters within a classical pipeline. Fig. 1 illustrates the proposed supervised learning module, based on a Random Forest Classifier, within the state-of-the-art detection pipeline. The proposed module takes key features extracted from the point-cloud cluster as input, producing an object classification as output; in this layout, the supervised module operation does not affect the general framework.

3. Methodology

This section details the complete methodology used to develop the LiDAR detection and classification pipeline. It begins with an overview of the approach, followed by a comprehensive explanation of the dataset construction, feature analysis, model training, and model tuning. Each stage is described to highlight the systematic procedures and techniques employed to optimize model performance.

3.1. Overview

The entire work process is summarized in Fig. 2. The construction of a supervised learning module that can be included in clustering-based detection pipeline begins with experimental data collection. The relevant objects in each relevant scenario are extracted and classified to form a labelled dataset. From this set of example objects, A large number of potentially predictive features are extracted. The extracted features are analysed qualitatively and quantitatively through an Exploratory Data Analysis (EDA) to assess their predictive power and subsequently selected. The resulting set of features is used for training and validating a Random Forest Classifier. The model is then fine-tuned by systematically varying its hyperparameters and the number of features through a Recursive Feature Elimination (RFE), producing a grid of potential models characterized by control parameters and obtained KPIs (computational cost and accuracy). This grid allows the selection of the appropriate model for each context based on the KPIs. As in the initial stage, the process loops back to the experimental stage by testing the classifier embedded in the classical detection pipeline in a dynamic real-world scenario.

3.2. Feature selection

The proposed framework aims to generate a light and information-dense set of features for each cluster. Features are selected and extracted from the labelled point clouds. In particular, the focus is placed on statistical and geometrical parameters that characterize the target.

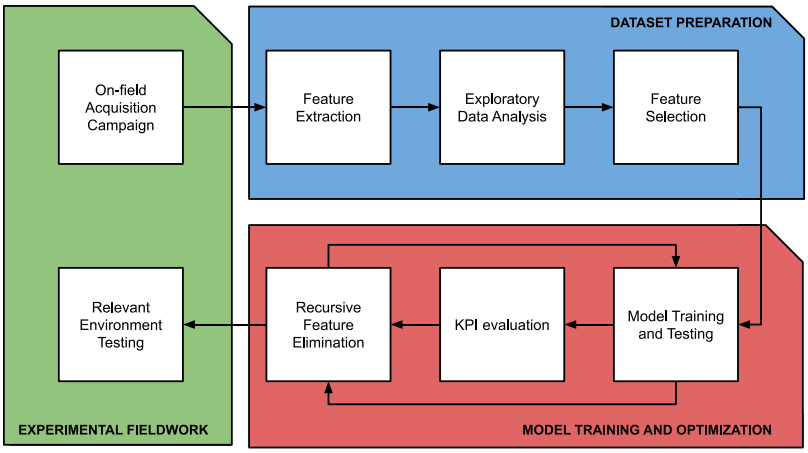


Fig. 2. Work process.

Information on the target's shape is obtained by performing the Principal Component Analysis (PCA) on the point cloud, extracting the eigenvalue (also referred to as *explained variance*) and eigenvectors (principal directions) of the covariance matrix. To improve the computational efficiency, PCA calculation is based on Singular Value Decomposition (SVD) Eq. (1). A LiDAR point cloud of a n-points cluster is expressed by a set of LiDAR coordinates $\boldsymbol{x_1}$, $\boldsymbol{x_2}$, $\boldsymbol{x_3}$ that can be stacked together to produce the cluster point cloud matrix $\boldsymbol{X}$. The centred point cloud $\boldsymbol{X}_c = \boldsymbol{X} - \bar{\boldsymbol{X}}$ is calculated, where $\bar{\boldsymbol{X}}$ represent the mean vector. The obtained right singular vectors matrix $\boldsymbol{V}$ contains the principal direction and can be used to express the point cloud into the new reference system defined by the principal directions, according to Eq. (2), where $\boldsymbol{Z}$ is the cluster point cloud expressed in the principal directions.

$$\boldsymbol{X}_c = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{1}$$

$$\boldsymbol{Z} = \boldsymbol{X}_c\boldsymbol{V} \tag{2}$$

Being $\boldsymbol{S}$ the diagonal matrix of singular values s_i, the eigenvalues of covariance matrix (*explained variance*) can be computed according to Eq. (3).

$$\lambda_i = \frac{s_i^2}{n-1} \tag{3}$$

Thus, for each cluster, the eigenvalue λ_1, λ_2, λ_3 are computed, obtaining a key feature set actually showing the variance of the point-cloud on the three principal directions. Indeed, the eigenvalues and their ratios can revalue key aspects of the point cloud observed, such as how much a direction predominates the others. Line- and surface-shaped objects are characterized by specific ratios of the eigenvalues that are thus obtained and can be effectively discerned. Thus, the ellipticities $e_{1,2}$, $e_{1,3}$ and $e_{2,3}$ of the clusters in the planes defined by the pairs of principal directions are calculated according to Eq. (4)

$$e_{i,j} = 1 - \frac{\lambda_j}{\lambda_i} \tag{4}$$

If eigenvalues provide information regarding the importance of a direction, then the cluster ellipticities provide information about its shape and deviation from a spherical or isotropic distribution. Moving from the point clouds of the targets expressed in the sensor's reference system to that obtained through PCA is essential to standardize their representation and the consequent extraction of geometric or statistical features, strongly decreasing the dependence on object orientation. Skewness γ and Kurtosis k can be calculated for each principal direction as provided in Eqs. (5) and (6).

$$\gamma = \frac{\sum_{i=1}^{n}(\boldsymbol{Z}_i - \bar{\boldsymbol{Z}})^3}{(n-1)\,\boldsymbol{\sigma}^3} \tag{5}$$

$$k = \frac{\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{Z}_i - \bar{\boldsymbol{Z}})^4}{\boldsymbol{\sigma}^4} \tag{6}$$

To conclude, skewness provides information about the asymmetry of the distribution, while kurtosis provides information about the shape of the peak of the distribution relative to a normal distribution. Both measures can be useful for extracting additional information from the structure and shape of the point cloud.

Considering that the point-cloud provided by the LiDAR also contains the value of intensity as an integer value in the [0,255] range, this could be further processed in order to obtain additional features. The natural step is to calculate parameters related to the central moments. Hence, mean μ_ι, standard deviation σ_ι, Skewness γ_ι and Kurtosis k_ι for the intensity distribution are calculated. Moreover, the statistical entropy (or Shannon Entropy) is computed. Eq. (7) shows the Entropy value H of a cluster's intensity distribution, where p_i represents the probability of finding a certain intensity value within the cluster.

$$H = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{7}$$

Finally, an estimation of spatial auto-correlation is also performed, deriving information about how similar intensity values tend to cluster in regions of space. Therefore, the Moran's I (Moran, 1950), shown in Eq. (8), and Geary's C (Geary, 1954), shown in Eq. (9), are calculated. In particular, the point cloud $\boldsymbol{X}$ is used to compute the matrix of spatial weights $\boldsymbol{W}$ with zeros on the diagonal, where each component w_{ij} is based on the normalized euclidean distance; the intensity value $\boldsymbol{\iota}$ is used as the feature variable.

$$I = \frac{n}{S_0} \cdot \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} \cdot (\iota_i - \bar{\iota})(\iota_j - \bar{\iota})}{\sum_{i=1}^{n}(\iota_i - \bar{\iota})^2} \tag{8}$$

$$C = \frac{(n-1)}{2S_0} \cdot \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(\iota_i - \iota_j)^2}{\sum_{i=1}^{n}(\iota_i - \bar{\iota})^2} \tag{9}$$

where S_0 is the sum of all the weight w_{ij}, i.e. $S_0 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}$.

Accordingly, the labelled dataset is post-processed including the presented features, for a total of 19 scalar statistics of interest. Details of the experimental campaign and information on the dataset obtained can be found in Section 4.1.

3.3. Feature analysis

Subsequently, the focus is placed on understanding which of the extracted parameters can actually be effectively used to predict the class of an object by quantifying its importance. As an Exploratory Data Analysis (EDA), features distribution are reported in a violin plot form, allowing for a comprehensive visualization of the means, the data

spreads and on the shapes of distributions, thanks to a kernel Density Estimation (KDE) (Parzen, 1962). To obtain quantitative information about whether a feature set can be discriminative for object-class association, a one-way analysis of variance, i.e., an ANOVA (Fisher, 1992) test, is often used. However, since the assumptions for a one-factor analysis of variance are not fulfilled, the Kruskal–Wallis H-test (Kruskal and Wallis, 1952), alternatively known as a one-way ANOVA by ranks test, is employed instead. Since the test is based on the null hypothesis of no significant difference among the feature of the groups, the result can be displayed in terms of p-values. If a feature has a p-value less than the chosen value of significance, it can be considered with good probability as a discriminative feature. In other words, such a feature may be a candidate for the set of features useful for predicting the class to which the object belongs. An in-depth discussion of the extracted feature analysis can be found in Section 5.1.

3.4. Model training

Once the features are obtained, it is necessary to select an algorithm that can accept these variables as training data and classify the target with high accuracy. Accordingly, three classifiers available in the literature are compared. Specifically, a Random Forest classifier (Ho, 1995), a Multi-Layer Perceptron (MLP) (Haykin, 1994) and a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) are chosen. The training dataset is split into 70% for training and 30% for testing. A grid of hyperparameters is defined to obtain the best combination using average accuracy from the 5-fold cross-validation test as a metric. The Random Forest classifier is chosen due to the higher accuracy as demonstrated in Section 5.2. Moreover, it allows for an a posteriori evaluation of feature importance through the model itself. Once the model is fit on all features of interest, an estimate of feature importance can be obtained, evaluating how much each feature contributes to the model's decision-making process. In particular, the Mean Decrease of Impurity (MDI) is evaluated among the decision trees of the Random Forest.

3.5. Model tuning

Considering the need to apply the model in an operational environment, the real time constraint becomes crucial. Therefore, features requiring an incompatible extraction time are eliminated from the set. To select incompatible features, the computational cost is extracted from a test on a real LiDAR scan reporting 40 targets of different nature and size and averaged on a set of 100 run.

To further investigate the feature dependency, a Recursive Feature Elimination (RFE) procedure is also performed. Considering the features importance distribution obtained from the MDI values, subtracted from the features incompatible with the real time constraint, one feature at a time is eliminated, in descending order of importance. For each iteration, the model is retrained on the new set of features and tested with the 5 folds cross validation test, calculating a mean accuracy value. Moreover, given that the computational cost in prediction varies indicatively linearly with the number of estimators in the model, for each feature-set iteration, the number of Random trees estimators is made to vary systematically between 64 and 128; the minimum number of features is limited to 2. This setup results in a test grid of size 15×65, from the number of feature-sets and the number of possible estimators respectively, leading to 975 combinations. The other model features are left fixed as obtained from the hyperparameters tuning. The KPIs calculated for each point identified in the test grid are listed below:

- $\boldsymbol{T}_1$: is the feature extraction time matrix, where each component $T_{1,ij}$ indicates the mean time needed to extract the feature set, for the ijth combination of the test grid. It is tested on the 40 cluster test scenario and averaged on 100 runs. Since this parameter is related only to the number of features in the feature-set, a vector would be sufficient to express this KPI. However, a matrix $\boldsymbol{T}_1$ is build by tiling 65 times the computational time vector in order to make it compatible with the size of the test-grid.
- $\boldsymbol{T}_2$: is the classification time matrix, where each component $T_{2,ij}$ indicates the mean time to perform the classification on the test set, averaged on the number of targets, for the ijth combination of the test grid.
- $\boldsymbol{A}$: is the mean accuracy matrix where each component A_{ij} indicates the mean accuracy obtained from the 5 folds cross validation, for the ijth combination of the test grid.

After calculating the feature extraction time matrix $\boldsymbol{T}_1$ and the classification time matrix $\boldsymbol{T}_2$ for each test grid combination, these matrices are summed to obtain the generalized computational cost matrix $\boldsymbol{T}$. The results derived from the test grid are extensively discussed in Section 5.3, offering a broad range of models and feature sets to guide the selection of the most suitable predictive model on a case-by-case basis.

Once a specific combination of parameters and features is selected, the Random Forest Classifier is retrained on the entire dataset and evaluated on two dynamic operational scenarios, the computational cost of the entire procedure, from sensor data parsing to detection and classification, is continuously monitored as a supplementary KPI. Three additional static scenarios are used to evaluate the effectiveness of the clustering algorithm in more complex environments and its impact on classifier performance. Finally, a progressive object truncation test is performed to observe the effect of partial point clouds on accurate classification.

Table 1
LiDAR main specifications.

Spec.	Value
Vertical FOV	31°
Vertical resolution	1°
Horizontal FOV	360°
Horizontal resolution	0.09° to 0.36°
Range	120 m
Accuracy	0.01 m

Table 2
Organization of the dataset acquisition series.

Class	Members
Quay	238
Motor Boat	313
Sailing Boat	221
Ship	148

4. Dataset

This section provides details on the construction of the dataset and its field acquisition.

The training and testing phases of a LiDAR supervised classifier require a large dataset of sample point clouds. In the automotive field, large labelled datasets are available, such as KITTI (Geiger et al., 2013), ApolloScape (Huang et al., 2018) and others. In the marine field, LiDAR dataset availability suitable for classifier training is more limited regarding the number of frames and variety of targets, making the available datasets more oriented towards developing multi-sensor data fusion systems. The open dataset available in Helgesen et al. (2022) provides a limited variety of targets; The MIT Sea Grant Marine Perception Dataset (Autonomous Underwater Vehicles Laboratory, 2022), collected using the RoboWhaler Autonomous Laboratory (DeFilippo et al., 2021) presents beneficial characteristics for multi-sensor data fusion developments, but targets are acquired using a 16-channel mechanical LiDAR; the state-of-the-art LiDAR sensor is characterized by several channels in the 32–128 range. Finally, the Pohang canal

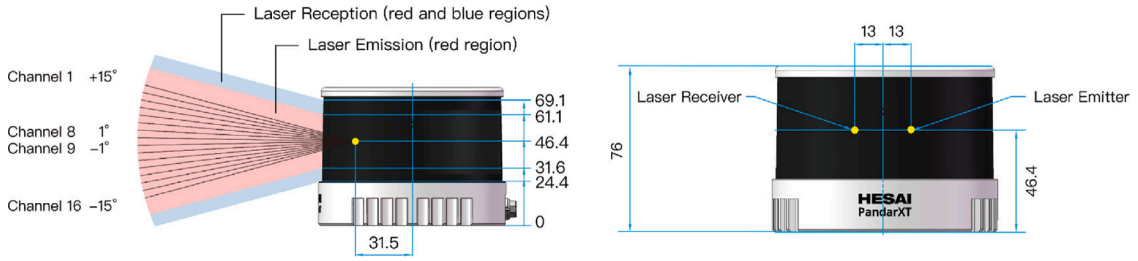


Fig. 3. HESAI Pandar XT-32 LiDAR technical draws.

dataset (Chung et al., 2023) provides a variegated and state-of-the-art sensory dataset related to inland navigation, which is unsuitable for the open seas. For this reason, a new dataset is collected, considering a wide variety of vessels of different sizes, acquired from different Point Of Views (POV). A preliminary post-processing activity is organized to obtain a set of labelled sample targets. The classes chosen for labelling the dataset are:

- *Quay*: point-cloudelonging to docks, piers, concrete constructions and infrastructure.
- *Motor Boat*: point clouds of small motor boats (indicatively less than 24 m in length); motorboats, dinghies, Rigid- hull Inflatable Boats (RIBs) small pilot boats and similar vessels belong to this category.
- *Sailing Boat*: point-clouds of boats characterized by the presence of one or more masts suitable for carrying sails (either unfurled or lowered).
- *Ship*: point clouds of large ships or vessels significantly longer than 24 m.

4.1. On-field dataset acquisition

A wide range of LiDAR sensors are currently available, from the older 16-channel mechanical LiDARs to the new solid-state LiDARs. The most used to provide situational awareness to autonomous vehicles are mechanical LiDARs with 32, 64 or 128 channels, as they are the best compromise between point cloud density, operation range and price. For this reason, a field acquisition campaign was performed using 32-channel, representing one of the most popular LiDAR sizes. Specifically, a HESAI Pandar XT-32 was used whose technical characteristics are shown in Table 1.

A schematic representation of the sensor with the channel distribution is given in Fig. 3.

The acquisition campaign was organized to acquire a wide variety of vessels from different POVs.

The dataset consists of LiDAR scans obtained during different experimental data acquisition campaigns, totalling about 15 min of useful LiDAR video recordings, from which 300 representative scans are extracted. A first set of scans is obtained from fixed locations by placing the sensor on different quays to acquire moored boats of different types and sizes. A second set of scans is acquired from the previously described location but with a focus on vessels navigating in the area surrounding the quay acquisition points. A third set of scans is acquired by placing the sensor on a small boat and navigating between moored boats. These three datasets were collected during the experimental campaign presented in Faggioni et al. (2022b,a) and were acquired in the Gulf of La Spezia in Liguria, Italy. A fourth additional dataset was collected by sailing in blue water during ship trials in the Gulf of Augusta, in Sicily, Italy, as presented in Martelli et al. (2022), collecting LiDAR scans of large vessels. The overall dataset comprising the four sets of acquisitions consists of several and variegated targets differing in POV, distance, type, size, and weather/sea conditions, ensuring coverage of many operational scenarios. As an illustrative example, Figs. 4 and 5 show the 2D projected LiDAR point clouds of two operational

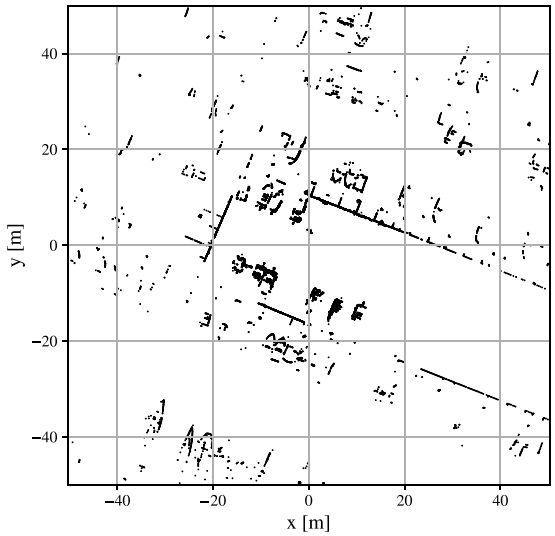


Fig. 4. Sample LiDAR scan of moored boat acquired from a boat sailing in an harbour.

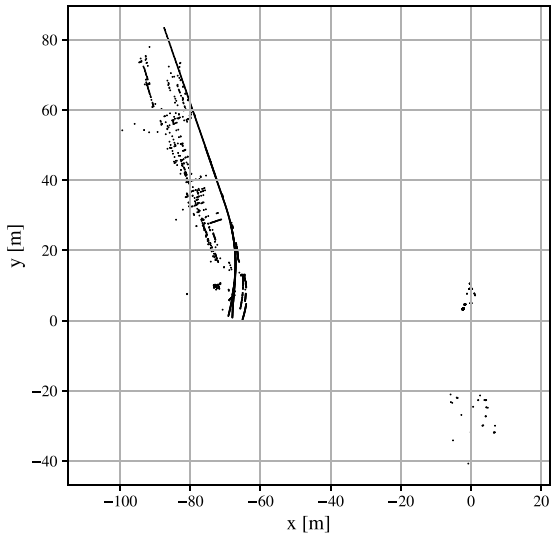
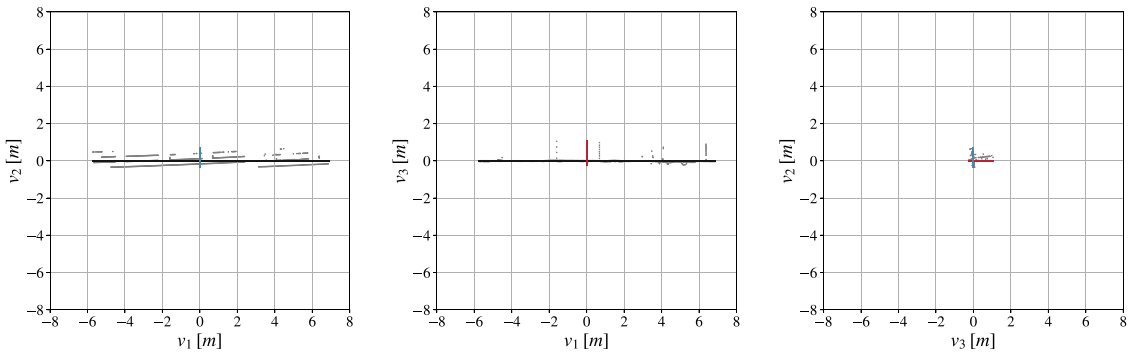


Fig. 5. Sample LiDAR scan of large ship acquired in blue water.

scenario sample scans. Specifically, Fig. 4 shows a harbour scenario in which several moored boats are acquired, whereas Fig. 5 shows the deep-sea acquisition of a large ship.

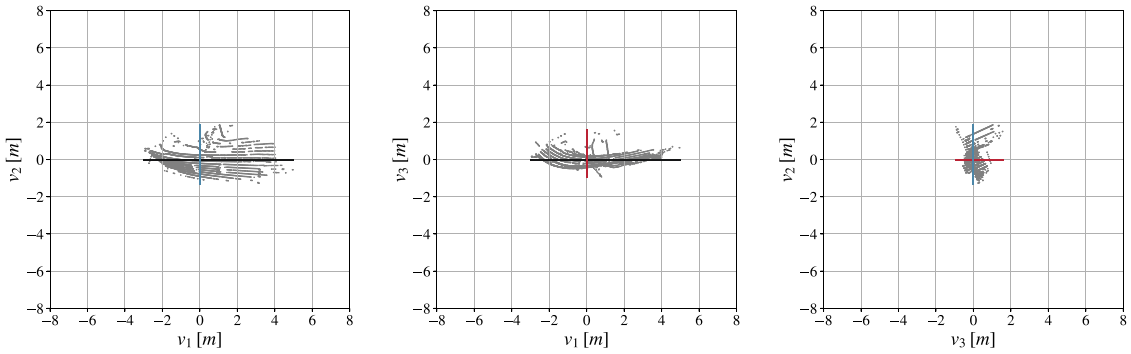
In particular, each suitable target is extracted and labelled from a selection of 300 relevant scans, obtaining a set of 928 examples divided into classes as shown in Table 2.

Figs. 6 7, 8 and 9 present a representative point cloud for each of the 4 identified classes, respectively Quay, Motor Boat, Sailing Boat and



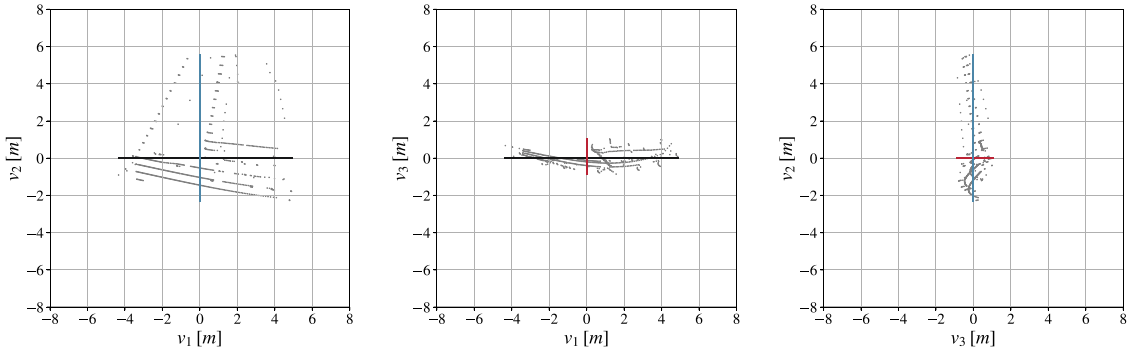
(a) PCA direction 1 and 2. (b) PCA direction 1 and 3. (c) PCA direction 2 and 3.

Fig. 6. Principal Component Analysis on a LiDAR point-cloud of a quay.



(a) PCA direction 1 and 2. (b) PCA direction 1 and 3. (c) PCA direction 2 and 3.

Fig. 7. Principal component analysis on a LiDAR point-cloud of a motorboat.



(a) PCA direction 1 and 2. (b) PCA direction 1 and 3. (c) PCA direction 2 and 3.

Fig. 8. Principal component analysis on a LiDAR point-cloud of a sailing boat.

Ship. The point cloud is expressed in the principal directions $\boldsymbol{v_1}$ $\boldsymbol{v_2}$ $\boldsymbol{v_3}$, according to Section 3.

A sample prototype of the dataset is given in Table 3.

5. Analysis results

This section presents the key findings, including feature analysis, model training, model tuning, and field testing. Quantitative and qualitative analyses are provided to assess the methodology's effectiveness at each stage.

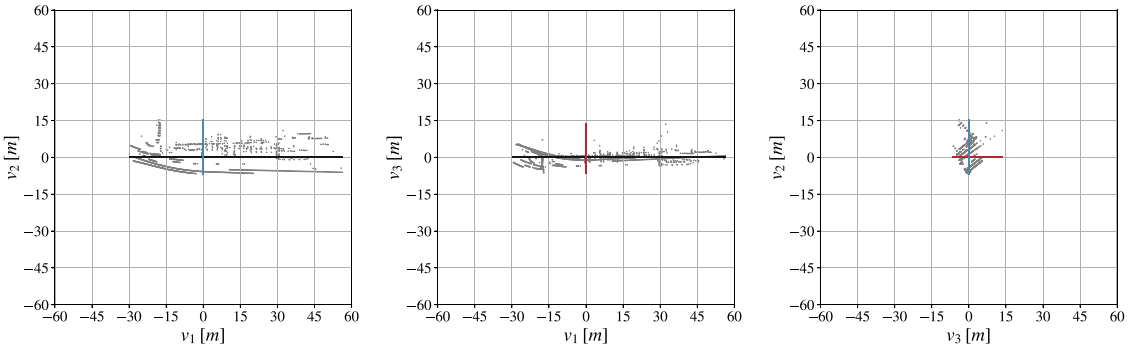
5.1. Dataset features analysis

This section reports the results of the qualitative and quantitative Exploratory Data Analysis of the features extracted from the dataset.

5.1.1. Qualitative analysis

Initially, the feature distributions for each class are illustrated through violin plots. The geometric information of the point cloud is shown through the feature distributions in Figs. 10 and 11. Hence, Fig. 10(a) shows the 3 PCA eigenvalues, i.e., the data variance along the 3 main directions, using a logarithmic scale. In Fig. 10(b) are presented the ellipticities calculated as the ratio of eigenvalues.

These two graphs show the variance in the principal directions and their ratio; the shown distributions reflect the geometric characteristics of the targets. Bulky-shaped targets such as *Motor Boat* and *Ship* are characterized by similar mean values of the ellipticities involving the second eigenvalue ($e_{1,2}$, $e_{2,3}$), while they are distinguished by the value of the eigenvalues due to the significant difference in the size of the objects. *Quay* class linear objects such as the edges of docks that have one direction very predominant over the others; consequently,

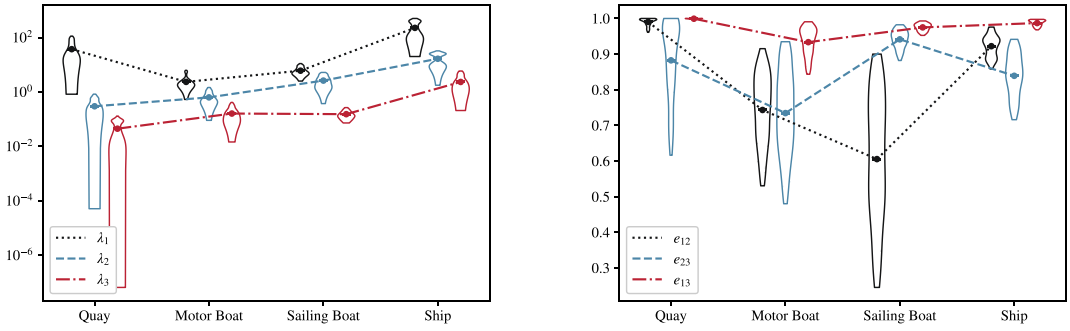


(a) PCA direction 1 and 2. (b) PCA direction 1 and 3. (c) PCA direction 2 and 3.

Fig. 9. Principal component analysis on a LiDAR point-cloud of a ship.

Table 3
Sample prototype of the dataset.

Variable	Type	Description
Class	String	Label describing the object
$\boldsymbol{X}$	n by 3 array	3D point cloud
$\boldsymbol{Z}$	n by 3 array	3D point cloud expressed in principal directions
$\boldsymbol{\iota}$	n by 1 array	Intensity values
λ_1	Scalar	First eigenvalue obtained from PCA
λ_2	Scalar	Second eigenvalue obtained from PCA
λ_3	Scalar	Third eigenvalue obtained from PCA
$e_{1,2}$	Scalar	Point cloud Ellipticity calculated in principal directions 1–2
$e_{1,3}$	Scalar	Point cloud Ellipticity calculated in directions 1–3
$e_{2,3}$	Scalar	Point cloud Ellipticity calculated in principal directions 2–3
γ_1	Scalar	Point cloud Skewness calculated in principal direction 1
γ_2	Scalar	Point cloud Skewness calculated in principal direction 2
γ_3	Scalar	Point cloud Skewness calculated in principal direction 3
k_1	Scalar	Point cloud Kurtosis calculated in principal direction 1
k_2	Scalar	Point cloud Kurtosis calculated in principal direction 2
k_3	Scalar	Point cloud Kurtosis calculated in principal direction 3
μ_ι	Scalar	Mean intensity value
σ_ι	Scalar	Intensity standard deviation
γ_ι	Scalar	Intensity Skewness
k_ι	Scalar	Intensity Kurtosis
H	Scalar	Shannon entropy of the intensity
I	Scalar	Moran's I spatial auto-correlation index
C	Scalar	Geary's C spatial auto-correlation index



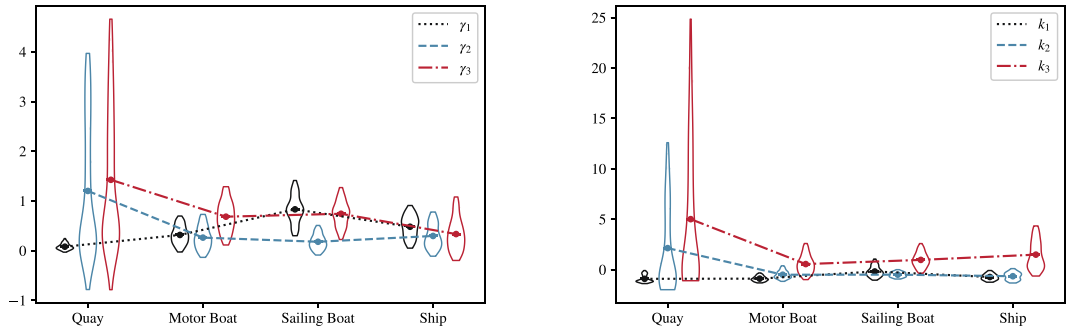
(a) Log-scale point-cloud eigenvalues obtained from the PCA. (b) Point cloud ellipticities calculated in the principal directions.

Fig. 10. Point cloud violin plots: first set.

the average eigenvalue for that direction λ_1 predominates over the other two and the two ellipticities $e_{1,2}$ $e_{1,3}$ in which it is involved are higher than the remaining. Planar targets such as a *Sailing Boat* are characterized by the first two eigenvalues (λ_1, λ_2) having close means and the resulting ellipticity $e_{1,2}$ being much lower than the other two; this is due to the presence of the mast and sails that provide the target with two main directions in which the variance (and hence the correspondence eigenvalue) is comparable.

Bulky-shaped targets such as *Motor Boat* and *Ship* exhibit similar average values for the ellipticities involving the second eigenvalue ($e_{1,2}$, $e_{2,3}$). However, these targets can be distinguished by their eigenvalue mean values, which reflect the significant size differences between the objects.

The *Quay* class identify linear objects like the edges of docks, where one direction is significantly more prominent than the others. Consequently, the eigenvalue (λ_1) corresponding to the predominant



(a) Point cloud Skewness calculated in the principal directions. (b) Point cloud Kurtosis calculated in the principal directions.

Fig. 11. Point cloud violin plots: second set.

direction is much higher than the other two. As a result, the ellipticities $e_{1,2}$ and $e_{1,3}$, which involve the dominant eigenvalue, are higher compared to the remaining.

Planar targets, such as a *Sailing Boat*, are characterized by similar average values for the first two eigenvalues (λ_1, λ_2); the resulting ellipticity $e_{1,2}$ is much lower than the other two. This phenomenon is characterized by the presence of the mast and sails, giving the target two main directions with comparable variance, and thus, similar corresponding eigenvalues.

Figs. 11(a) and 11(b) show the distributions of the successive statistical characteristics of the point-cloud, respectively the point-cloud Skewness and Kurtosis calculated in the principal directions, according to Section 4.1. Considering the average Skewness values for the *Quay* class, a positive skew is found in the second and third main directions. Considering that the second and third main directions in this class roughly correspond to the width and height of the quay (see Fig. 6), the phenomenon is explained by the detection of most points in the first-impact surfaces of the sensor's laser pulse as outlined in Faggioni et al. (2022b), which is also increased by the presence of extremely flat surfaces. For the *Motor Boat* and *Ship* classes, there are no evident variations in Skewness, the differences being mainly due to distance and POV. However, for the *Sailing Boat* class, a lower average γ_2 value than γ_1 and γ_3 can be noticed, being Skewness value in the second principal direction approximately zero or slightly positive. This phenomenon is due to the presence of the mast and rigging that distinguishes this class from the others, making the distribution in that direction quite symmetrical.

Considering the distribution of mean values of Kurtosis over the three main directions, the point-cloud distribution is platykurtic, showing no particular differences between the different classes. An exception is the distinctive trend for the *Quay* class where the k_2 and in particular k_3 values are associated with leptokurtic distributions, confirming the pronounced acquisition of most points towards the first-impact surfaces as discussed for the Skewness distribution.

Intensity related features are reported in Fig. 12. Specifically, Fig. 12(a) shows the intensity distribution mean and the standard deviation for the classes; The mean values do not identify substantial differences between the classes with the exception of the *Quay* class, which is characterized by lower mean values. This is mainly due to the difference in the reflectivity of the material, as the quays are mainly made up of raw, unpainted concrete.

In Fig. 12(b) are reported the intensity of Skewness and Kurtosis without highlighting any distinctive features. The Skewness values are all quite similar and the intensity distributions are generally platykurtic, only the *Ship* class shows a slightly higher Kurtosis mean value, though it does not appear to be a particularly distinctive feature.

The additional features distribution are reported in Fig. 13. The Shannon Entropy, reported in Fig. 13(a), provides useful information for distinguishing between the *Quay* and *Ship* classes. The first is characterized by low statistical entropy, indicating less variable intensity values, whereas the second is characterized by higher entropy, indicating a more significant variability in intensity values. This phenomenon is likely caused by the difference between the two classes in terms of their dimensions and the physical material they are composed of, similar to the case of the mean intensity values. The docks offer a flat surface of the same material with a narrow cross-section, thus providing low variability in intensity values. Large ships, instead, offer larger, heterogeneous surfaces with variable encounter angles and variable materials, resulting in much higher variability in intensity values and, thus, higher statistical entropy.

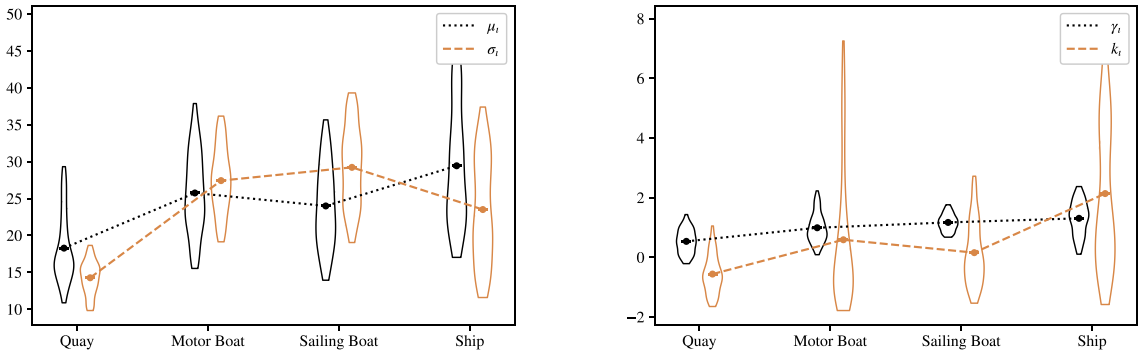
The spatial auto-correlation indices distribution within the classes i.e. Moran's I and Geary's C are reported in Fig. 13(b). No notably different spatial patterns between classes are evident and no remarkably different spatial patterns are evident between the classes. However, the Sailing Boat class shows a slightly lower average value for Moran's I (and therefore a higher value for Geary's C) than the other boats. In other words, points close to each other have less similar intensity values than in the other classes. This is likely due to the mast and rigging, which the LiDAR captures as a more heterogeneous point cloud in terms of intensity.

Fig. 14 provides an overview of the distribution of the mean of the extracted features for each class. Specifically, the mean values for each class feature are normalized by the maximum absolute value that the feature assumes within the entire dataset. The green solid line with circular markers represents the *Quay* class; the blue dashed line with cross markers identifies the*Motor Boat* class, the red dotted line with square markers denotes the *Sailing Boat* class, and the orange dash-dot line with diamond markers represents the *Ship* class. Through careful examination of the proposed features alongside graphical representations, especially focusing on the mean and the spread of the first quartiles, it becomes apparent that a combination of these 19 parameters could potentially be predictive of a class. This suggests constructing predictive models that use these parameters to classify LiDAR-detected objects.

5.1.2. Quantitative analysis

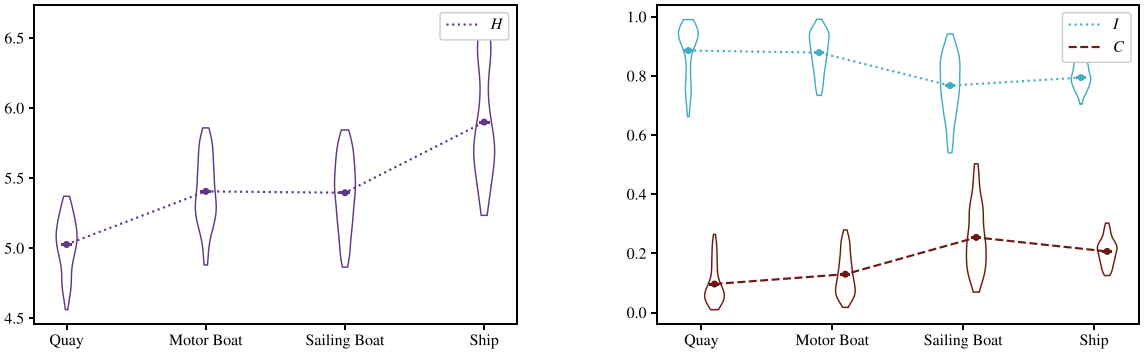
This section focuses on the quantitative analysis, complementing the qualitative observations made. A correlation matrix is presented in Fig. 15, providing information on how the selected features tend to vary together. It is worth noticing that the correlation between eigenvalues and ellipticities is quite high.

In addition, Fig. 16 presents the p-values of each feature obtained carrying out the Kruskal–Wallis test; being the p-values very small, the bar plot is presented in logarithmic scale. The standard 0.05 p-value is reported as a red dashed line, moreover, the analysis is carried out considering also a random variable. As can be seen, all the selected features largely reject the null hypothesis; note also that the p-value associated with the third direction point-cloud kurtosis k_3, the feature of least importance, is about $4.77e{-}12$ against an average p-value of the random feature of 0.65 per 100 runs.



(a) Intensity mean and standard deviation. (b) Intensity Skewness and Kurtosis.

Fig. 12. Intensity related features violin.



(a) Intensity Shannon Entropy. (b) Spatial auto-correlation indices.

Fig. 13. Additional features violin plots.

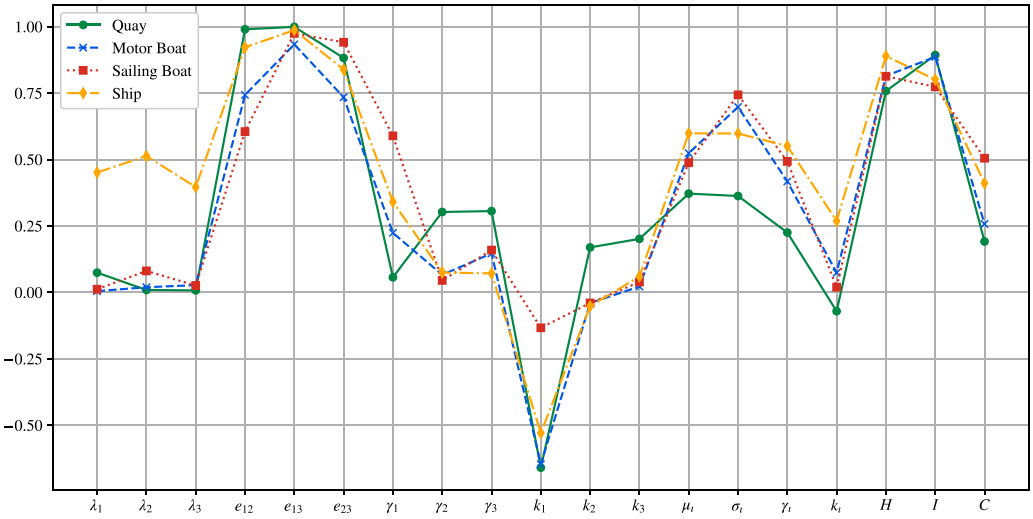


Fig. 14. Global normalized feature distribution.

The quantitative analysis confirms the possibility of using these features, all or a selection, for predictive purposes.

5.2. Supervised model training

This section discusses the training of the supervised model and the tuning of hyperparameters, according to Section 3. In order to select a classification model capable of operating on a feature dataset for comparison, three models were chosen. Specifically, a Random Forest (RF) Classifier, a Multi-Layer Perceptron (MLP), and a Support Vector Machine (SVM) were trained on the same feature dataset. To ensure a fair comparison, the tuning of hyperparameters was performed for all three models. The hyperparameter search grid is shown in Tables 4–6 for the RF, MLP, SVM tuning respectively. A 5-fold cross-validation test was performed on each hyperparameter combination as a metric. The set of hyperparameters that guarantees the best result on the test case is provided in the same table. Although the computational cost of classification was comparable for all the three cases, the RF classifier

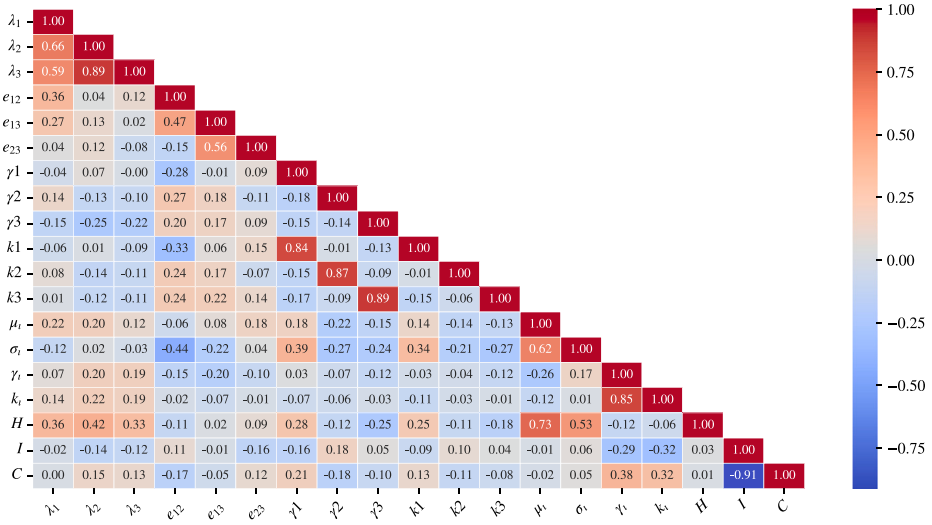


Fig. 15. Correlation matrix within the proposed features.

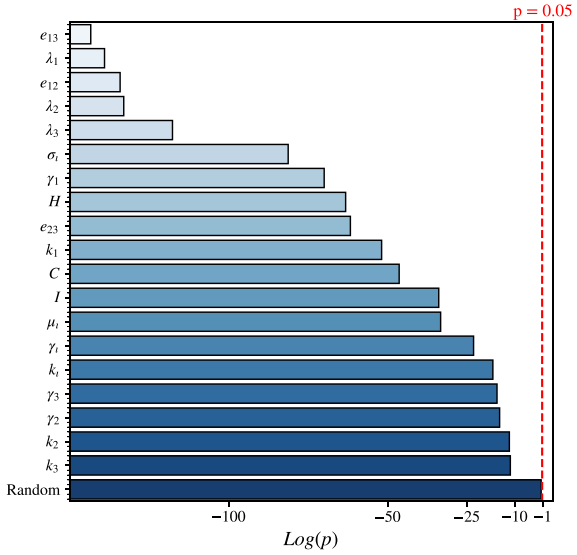


Fig. 16. Kruskal–Wallis p-values bar plot.

Table 4
RF hyperparameters tuning grid.

Hyperparameter	Search range	Best
N° estimators	[64–256]	128
Max depth	[2–50, None]	10
Max features	[$\sqrt{n}$, $\log_2 n$, None]	$\sqrt{n}$
Min samples leaf	[1–5]	2
Min samples split	[2–10]	5

achieved an average accuracy of 94% in the 5-folds cross-validation test, compared to 90% for MLP and 91% for SVM.

It is also worth noting that, in addition to achieving higher accuracy, the Random Forest Classifier inherently provides feature importance rankings, which are essential for the proposed framework. For these reasons, the Random Forest Classifier was chosen as the classification model.

The Random Forest classification report on the whole test set is shown in Table 7, offering an assessment of the predictive ability of the model on the different classes and in a global view. Moreover, the 5-fold cross-validation result is reported in Table 8.

Table 5
MLP hyperparameters tuning grid.

Hyperparameter	Search range	Best
Hydden layer size	[(10–200,)(10–200,10–200)]	(100,20)
Activation	[relu, tanh]	tanh
Solver	[adam, lbfgs]	adam
Alpha	[0.001–0.1]	0.005
Learning rate	[constant, adaptive]	constant
Batch size	[auto, 10–100]	auto

Table 6
SVM hyperparameters tuning grid.

Hyperparameter	Search range	Best
C	[0.1–100]	10
Kernel	[linear, rbf, poly, sigmoid]	rbf
Gamma	[scale, auto]	scale

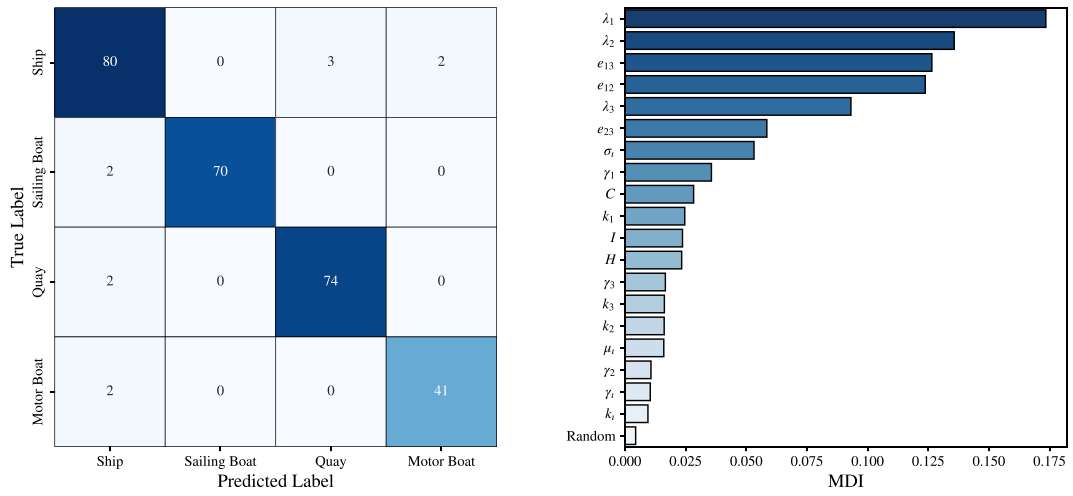
Table 7
Classification report.

Class	Precision	Recall	F1-score	Support
Motor Boat	0.94	0.96	0.95	85
Quay	0.99	0.96	0.97	72
Sailing Boat	0.99	0.99	0.99	76
Ship	0.95	0.95	0.95	43
Macro average	0.97	0.97	0.97	276
Weighted average	0.97	0.97	0.97	276
Accuracy			0.97	276

Table 8
5-folds cross-validation result.

Fold	Accuracy
1	0.90
2	0.98
3	0.99
4	0.97
5	0.87
Mean	0.94

The training results are displayed in Fig. 17. The confusion matrix obtained on the test set is shown in Fig. 17(a). Fig. 17(b) shows the MDI feature importance of the model, including the injection of a random variable in the training set, to verify that all the features provide a MDI greater than the one produced by a random variable. Confirming the Exploratory Data Analysis, the features contribute to



(a) Random Forest confusion matrix.

(b) Random Forest feature importance.

Fig. 17. Random forest training results.

the classification with a similar distribution of importance to that found with Kruskal–Wallis test presented in Section 5.1.

5.3. Model optimization and tuning

The model is trained and produces sufficiently accurate results for the intended application. However, it may be of interest to optimize the selection of features to be used, seeking the suitable trade-off between model lightness and prediction accuracy.

The first consideration to be made, bearing in mind the real-time application of this model, is the computational time required to obtain the feature set from the LiDAR point cloud. Since the two spatial auto-correlation indices (Moran's I and Geary's C) are the most computationally onerous, a test is performed on the computational cost of feature set extraction with and without the two indices. Specifically, the test is performed on a real LiDAR scan reporting 40 targets of different nature and sizes and averaged on a set of 100 runs. Moreover, the computational cost analysis is performed on an Off-the-Shelf laptop whose technical specifications are given in Table 9. The features extraction code is implemented in Python 3.12.

The average target feature extraction time of the whole set of features is 0.12 s with a standard deviation of 0.28 s. The average target feature extraction time of the whole set of features without the two spatial auto-correlation index is lowered to 0.0035 s with a standard deviation of 0.0005 s. Using the two spatial autocorrelation indices is not a sustainable choice for real-time applications where the LiDAR sensor has an operating frequency between 5 and 20 Hz. Considering that the two indices are associated with a fairly low Mean Decrease of Impurity (MDI) value, eliminating the two indices from the feature set leads to a considerable decrease in computational load at a negligible loss in accuracy. To quantify the loss in accuracy, the Random Forest Classifier is again retrained, excluding Moran's I and Geary's C from the training dataset, thus lowering the number of features from 19 to 17. The 5-fold cross-validation is repeated, and the results are shown in Table 10, outlining an average accuracy of 0.93%.

A Recursive Feature Elimination (RFE) process is conducted to explore feature dependency. Features are removed one by one in order of importance, and the model is retrained and tested at each step using 5-fold cross-validation to calculate mean accuracy. The number of Random Forest estimators is varied between 64 and 128 for each feature set, creating a 15×65 test grid, resulting in 975 combinations. The other model features were left fixed as obtained from the hyperparameters tuning (Table 4). Fig. 18(a) shows the feature extraction mean time T_1 as the number of features varies, in ms. Fig. 18(b) shows the specific feature-set corresponding to each feature number according to the RFE procedure.

Table 9
Personal Computer main specifications.

Spec.	Value
OS	Fedora Linux 40
Kernel	6.9.4
CPU	AMD Ryzen 9 6900HS 4,9 GHz
GPU	AMD Radeon RX 6700S
RAM	16 GB DDR5-SDRAM 4800 MHz

Table 10
5-folds cross-validation result without the spatial auto-correlation indices.

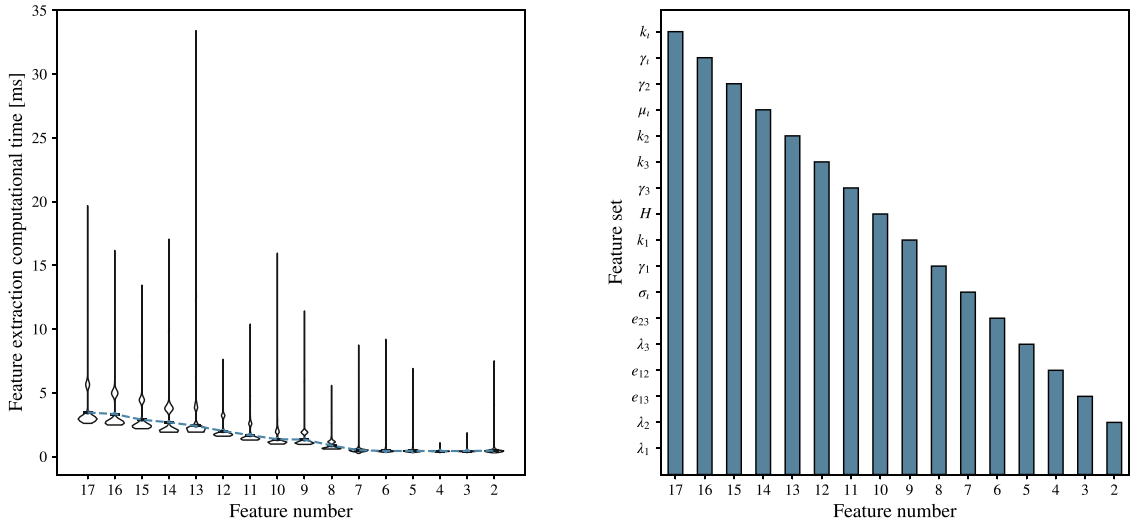
Fold	Accuracy
1	0.875
2	0.95
3	0.99
4	0.96
5	0.85
Mean	0.93

The global computational time T is shown as a heat map in Fig. 19. Note how time T_2 has little influence on the overall computational cost trend T, which is instead strongly dominated by feature extraction time T_1.

The average accuracy A obtained from the 5-fold cross-validation test for each combination of the test grid is shown as a heat map in Fig. 20. As can be seen, the variation in prediction accuracy varies greatly with the number of parameters, while it is not as affected by the number of estimators. As can be observed, the variation in prediction accuracy is dominated by the number of features and varies secondarily with the number of decision trees.

Fig. 21 shows the optimal choice of the number of Decision Trees (in green) for each feature set to maximize accuracy. In red is the achievable accuracy, in blue the computational cost. A minimum number of features to achieve accuracy close to 90% appears to be 8, while in each combination the computational cost is under 5 ms. From the results presented, the perfect combination of number of features and number of decision trees cannot be extrapolated, rather, a performance grid is presented in which the appropriate combination can be chosen on a case-by-case basis.

For the case study, it is chosen to impose a computational cost of less than 3 ms and to maximize accuracy. Therefore, a total of 15 features



(a) Average computational cost for each feature set.

(b) Selected feature set for each feature number.

Fig. 18. Computational cost calculation according to recursive feature elimination.

Table 11
Tests characteristics.

Test	Scan	Frequency	Duration	Average distance	Vessel approx. length	Ground truth class
Scenario 0	150	5 Hz	30 s	49 m	16 m	*Motor Boat*
Scenario 1	150	5 Hz	30 s	67 m	10 m	*Sailing Boat*

Table 12
Scenario 2 clusters' information.

Cluster ID	n° of points	Distance [m]	Reference label	Predicted label
0	722	78.78	Ship	Ship
1	919	21.14	Motor Boat	Motor Boat
2	1430	25.72	Quay	Quay
3	933	15.42	Motor Boat	Motor Boat
4	2091	17.5	Sailing Boat	Sailing Boat
5	1059	18.96	Motor Boat	Motor Boat

Table 13
Scenario 3 clusters' information.

Cluster ID	n° of points	Distance [m]	Reference label	Predicted label
0	305	85.28	Ship	Sailing Boat
1	58	88.08	Quay	Quay
2	322	61.98	Motor Boat	Motor Boat
3	259	80.47	Motor Boat	Motor Boat
4	353	87.5	Motor Boat	Motor Boat

are selected for use (see Fig. 18(b) for specific feature selection), i.e. neglect the skewness γ_i and the kurtosis k_i of the intensity; the number of Decision Trees is set to 72. Thus, the Random Forest Classifier is trained with the chosen parameters on the entire dataset.

5.4. Field verification

This section presents the results of tests conducted on the classification module embedded within the entire LiDAR detection pipeline. Specifically, marine LiDAR scans are read, parsed, and processed in real-time through the pipeline proposed in Section 1. The selected detection-pipeline is based on a DBSCAN clustering algorithm that analyses the LiDAR point cloud after noise filtering. Simultaneously with the detection and subsequent cluster identification, the feature-set discussed in Section 4.1 is extracted and fed into the Random Forest Classifier, according to . The classification obtained, time-by-time, and the overall computational cost of the entire process are saved as KPIs.

Two real-time test scenarios are set. In the first (Scenario 0), 150 frames are acquired at 5 Hz for 30 s total duration of a small motor yacht (Class: *Motor Boat*) sailing close to the acquisition point from right to left. In the second test (Scenario 1), 150 frames at 5 Hz for 30 s total duration are acquired of a sailing vessel (Class: *Sailing Boat*) sailing with the lowered sails from left to right with respect to the acquisition point and at a greater distance than in the first case. Main characteristics of the two experimental tests are summarized in Table 11.

Fig. 22 shows three RGB reference sample images for Scenario 0, obtained at the three key stages of the acquisition (beginning, middle, end); Fig. 23 represents the corresponding LiDAR scans of the same Scenario 0 time series projected in 2D. Out of 150 scans where the ground truth class is *Motor Boat*, the classifier predicted the correct class 140 times, the *Sailing Boat* class 10 times and never the other two, resulting in an accuracy of 93.33%. The average global computational time considering detection, feature extraction and classification is about 12 ms.

Fig. 24 shows three RGB reference sample images for Scenario 1, obtained at the three key stages of the acquisition (beginning, middle, end); Fig. 25 represents the corresponding LiDAR scans of the same Scenario 1 time series projected in 2D. Out of 150 scans where the ground truth class is *Sailing Boat*, the classifier predicted the correct class 148 times, the *Motor Boat* class 2 times and never the other two, resulting in an accuracy of 98.6%. The average global computational time considering detection, feature extraction and classification is about 9 ms.

To evaluate the combined performance of the detection and classification pipeline, three more complex scenarios were selected (Scenarios 2, 3, and 4). Unlike earlier tests conducted in open-water conditions (e.g., Scenarios 0 and 1), these scenarios focus on quay approach operations. Each scenario consists of a single static frame featuring several boats, either sailing or moored, of varying types, along with parts of the quay. Some boats are well-defined in the point cloud, while others appear less distinct, with clusters ranging from over 1000 points to fewer than 100. These static scenarios are designed to test the pipeline's performance against multiple objects of varying sizes, types,

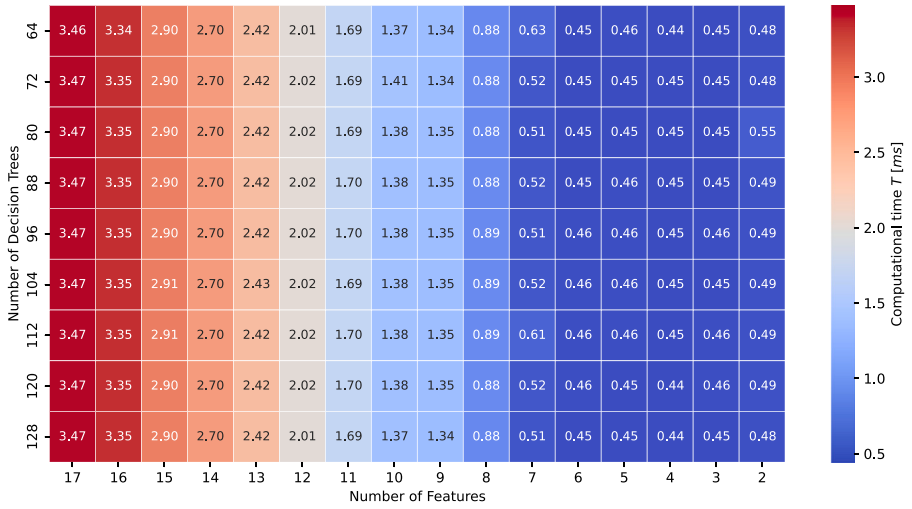


Fig. 19. Global computational time heat map.

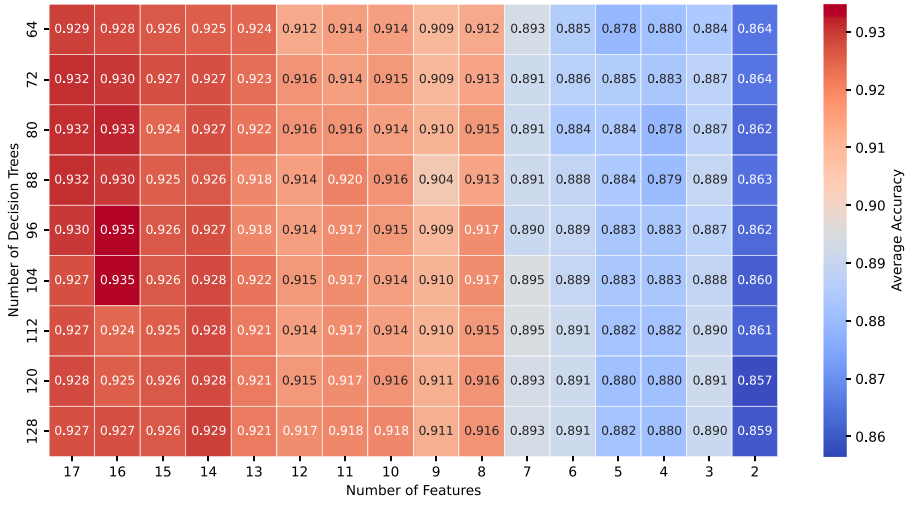


Fig. 20. Average accuracy heat map.

and distances within the same frame, enabling an analysis of partial occlusions, point density, and object distance.

Initially, water noise was filtered by removing points with an intensity below 10 and a height close to sea level (based on the sensor installation). The DBSCAN algorithm was then uniformly applied in all three scans using the radius parameters $\epsilon = 1$ m and a minimum of five points ($minpts = 5$) within the radius. Noise points identified by DBSCAN were excluded and only clusters with more than 50 points were retained for classification. Each valid cluster was then classified using the trained Random Forest Classifier, as previously described. The scenarios present increasing levels of difficulty. Scenario 2 contains a small number of well-defined targets, with only a few noisy points. Scenario 3 has a low number of targets, but some are partially occluded or fragmented; disturbing elements, such as mooring lines, are also present. Scenario 4 features a high number of targets, many of which are fragmented or occluded, along with a significant presence of noisy points. Figs. 26, 27 and 28 visually show the procedure described for Scenarios 2 3 4, respectively. Figs. 26(a), 27(a) and 28(a) show the raw scenario, projected in 2D. Figs. 26(b), 27(b) and 28(b) show the clusters identified by the DBSCAN algorithm and shown in different colours; the points marked by red X's represent the noise cluster. Compared to the raw scenario, the effect of the water noise filter can be observed. Finally, Figs. 26(c), 27(c), 28(c) show the valid clusters numbered, ready to be processed with the classifier. The POV of the sensor is shown with a red circular marker with a black border in all figures. Information and results of the analysis are given in the Tables 12–14 for Scenarios 2 3 4 respectively.

The results achieved in Scenario 2 show that the algorithm performs effectively on well-defined scenes and if the targets are well segmented. It can also be seen that the noise due to the divergent wave created by the hull of cluster 0 is well filtered out by the algorithm. The results obtained in Scenario 3 demonstrate the algorithm's effectiveness in environments with higher noise levels. Observing cluster 2, the algorithm successfully filtered out the mooring lines and accurately predicted the correct class. However, it is important to emphasize that proper segmentation is crucial for effective classification. In particular, observing cluster 0, a classification error corresponding to a segmentation error can be identified. Specifically, due to the shadow cones created by the first-impact obstacles, the stern of cluster 0 is detached from the rest of the hull and classified as noise. Additionally, small occluded objects are not considered valid clusters due to their low point count. Finally, Scenario 4 highlights the impact of occluded and partially acquired

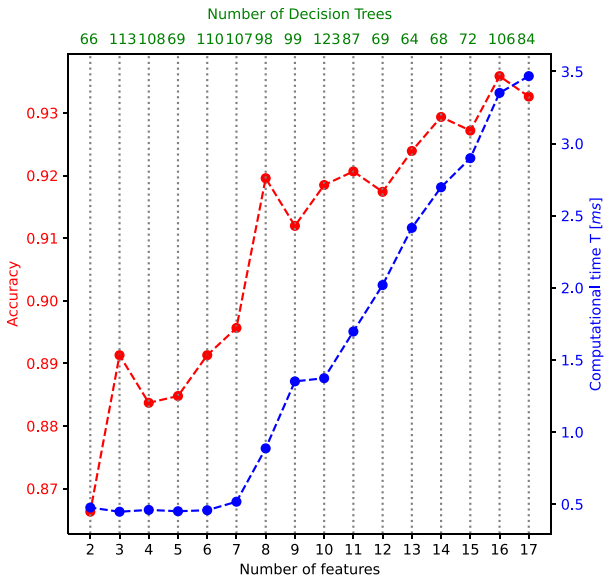


Fig. 21. Best number of decision trees to maximize the accuracy for each feature-set.

Table 14
Scenario 4 clusters' information.

Cluster ID	n° of points	Distance [m]	Reference label	Predicted label
0	564	42.02	Quay	Quay
1	3076	21.45	Quay	Quay
2	188	29.07	Quay	Motor Boat
3	74	35.69	Motor Boat	Motor Boat
4	383	24.74	Motor Boat	Motor Boat
5	126	32.46	Quay	Motor Boat
6	306	28.21	Motor Boat	Motor Boat
7	328	32.0	Sailing Boat	Sailing Boat
8	81	47.32	Quay	Motor Boat
9	75	45.91	Motor Boat	Motor Boat
10	690	18.25	Motor Boat	Motor Boat
11	169	33.23	Sailing Boat	Sailing Boat
12	101	27.65	Motor Boat	Motor Boat
13	122	36.09	Sailing Boat	Sailing Boat
14	87	23.79	Quay	Quay
15	499	41.37	Sailing Boat	Sailing Boat

obstacles on the accuracy of the predicted class. Classes are generally well assigned except in those cases where the object is partially shielded by other objects or poorly segmented (clusters 2, 5, 8). In those cases where the object is acquired without obstruction, segmentation and classification occur correctly and are little affected by distance or number of points. In a global view of the results, it appears that a low number of points or a greater distance to others does not affect the correctness of the classification as long as the proportions and main features of the object are correctly acquired. In contrast, a near object with a medium number of points is incorrectly classified if occluded or not properly acquired in its main shapes.

An additional series of tests was conducted to investigate the dependency of classification on target truncation. The dataset samples were divided by class, and an increasing percentage of each point cloud was progressively removed along the first principal direction. For each level of truncation, classification performance was evaluated, recording the percentage at which the first misclassification occurred, along with the corresponding confusion class. This procedure simulates the effect of partial target acquisition caused by occlusion from another object or the partial grouping of points resulting from a failure in the clustering algorithm. The test results are presented in Table 15, which reports, for each analysed class, the average percentage of occlusion that leads to the first classification failure, along with the most frequently confused class. The results indicate that once the truncation surpasses 40%, the classification becomes unreliable.

Table 15
Progressive truncation test results.

Class	Truncation [%]	Confusion class
Quay	55%	Motor Boat
Motor Boat	47%	Quay
Sailing Boat	40%	Motor Boat
Ship	56%	Motor Boat

6. Conclusions

The framework proposed in the paper demonstrates significant capabilities for the detection and classification of marine targets and obstacles using LiDAR. The integration of a lightweight classification module into the existing clustering-based workflow retains the advantages of effectiveness, simplicity, and generalization of the original framework without compromising real-time performance and adding the ability to predict the class of the target with high accuracy.

An extensive investigation was conducted to select and extract the geometric and statistical characteristics of the point cloud that could indicate the class of the target. A Random Forest Classifier was trained on the obtained feature training dataset and extensively tested on the test dataset, tuning the hyperparameters. The classification was tested by embedding it in the entire LiDAR detection pipeline on completely new real scenarios. The successful application of the combined learning framework in real-world scenarios opens new avenues for enhancing the situational awareness of Marine Autonomous Surface Vehicles, with potential benefits for operational safety and efficiency in the marine domain. The Random Forest Classifier module provides satisfactory results despite having a contained dataset with even non-dense point clouds. Moreover, it guarantees higher accuracy than other classifiers capable of using the same feature set. In addition, classification is subordinate to detection, guaranteeing the ability to detect a target even with a classification failure, not affecting the safety of navigation. However, this is reflected in the inability to adequately classify a target if the clustering-based detection pipeline fails. The results show that classification is effective as long as the clustering-based pipeline provides reliable clusters; in particular, if the cluster is over- or under-segmented, classification may fail. The tests conducted indicate 40% as a possible truncation limit for an object; above this limit, the statistical characteristics extracted from the cluster do not allow effective classification. The applicability domain of the proposed method is navigation in areas compatible with commercial LiDAR ranges. In particular, dynamic scenarios with low to medium distances such as port navigation, navigation in bays and during dock approaches. In these scenarios, the ability to perceive and classify a target that is navigating nearby enables reactive route planning including navigation rules (such as ColRegs). In low distance and low speed scenarios such as inland navigation in marinas, LiDAR can be more effectively exploited in occupancy grid mode as the extreme complication and connection of objects makes clustering-based detection sub-optimal. Despite the advantages and drawbacks demonstrated, the authors emphasize that additional tests and considerations are still necessary. In a sensor fusion perspective, the semantic data obtained from the proposed LiDAR pipeline can be combined with the well-known image-based classification to further enhance and improve the capabilities of the perceptual system; this aspect will be the subject of future developments.

Fig. 22. Reference RGB images of the Scenario 0 acquisition time series.

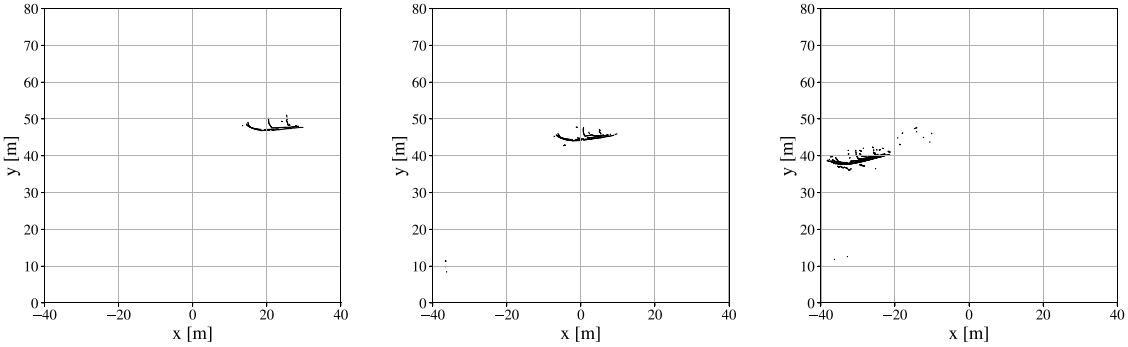


Fig. 23. LiDAR 2D point cloud of the Scenario 0 acquisition time series.

Fig. 24. Reference RGB images of the Scenario 1 acquisition time series.

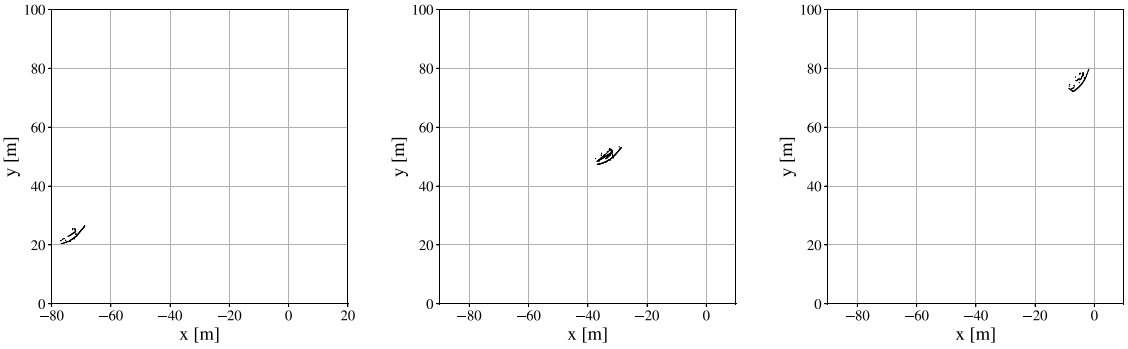
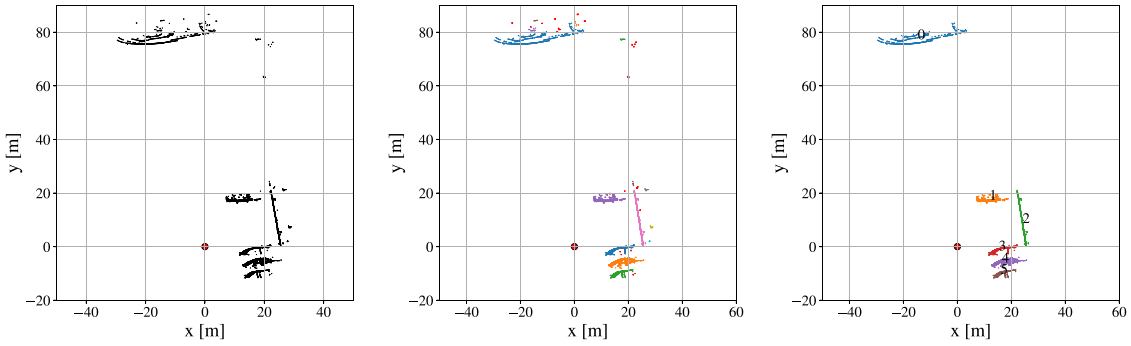
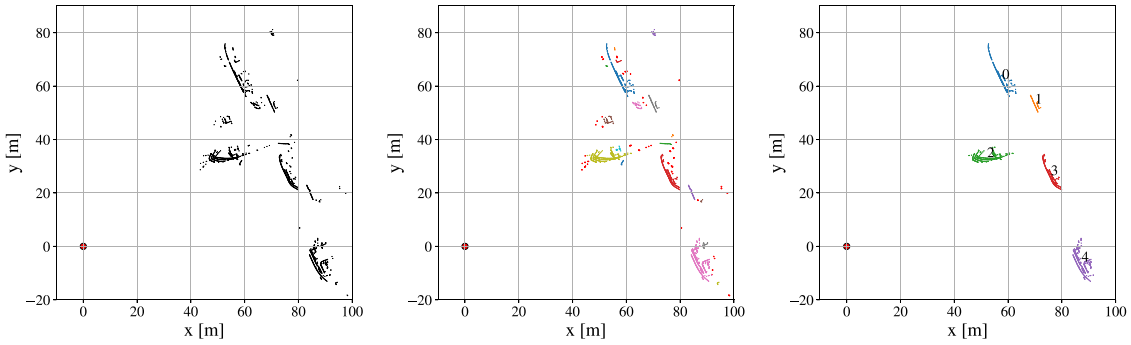


Fig. 25. LiDAR 2D point cloud of the Scenario 1 acquisition time series.



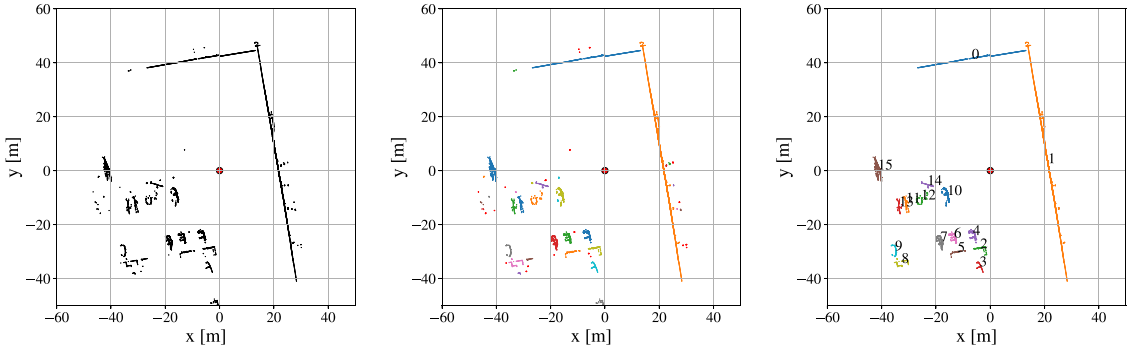
(a) Scenario 2. (b) Clustering on Scenario 2. (c) Cluster filtering on Scenario 2.

Fig. 26. Detection and classification pipeline tested on Scenario 2.



(a) Scenario 3. (b) Clustering on Scenario 3. (c) Cluster filtering on Scenario 3.

Fig. 27. Detection and classification pipeline tested on Scenario 3.



(a) Scenario 4 point cloud. (b) Clustering on Scenario 4. (c) Cluster filtering on Scenario 4.

Fig. 28. Detection and classification pipeline tested on Scenario 4.

CRediT authorship contribution statement

Filippo Ponzini: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Raphael Zaccone:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Michele Martelli:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was partially funded by European Union – NextGenerationEU. Piano Nazionale di Ripresa e Resilienza, Missione 4 Componente 2 Investimento 1.4 “Potenziamento strutture di ricerca e creazione di “campioni nazionali di R&S” su alcune Key Enabling Technologies”. Code CN00000023 – Title: “Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile – CNMS)”. However, views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible.

References

Autonomous Underwater Vehicles Laboratory, MIT sea grant marine perception dataset. URL: https://seagrant.mit.edu/auvlab-datasetsmarine-perception-1/.

[public dataset].
Bovcon, B., Kristan, M., 2022. Wasr—a water segmentation and refinement maritime obstacle detection network. IEEE Trans. Cybern. 52, 12661–12674. http://dx.doi.org/10.1109/TCYB.2021.3085856.
Campello, R.J.G.B., Moulavi, D., Sander, J., 2013. Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (Eds.), Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 160–172.
Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 77–85. http://dx.doi.org/10.1109/CVPR.2017.16.
Chung, D., Kim, J., Lee, C., Kim, J., 2023. Pohang canal dataset: A multimodal maritime dataset for autonomous navigation in restricted waters. Int. J. Robot. Res. 42, 1104–1114. http://dx.doi.org/10.1177/02783649231191145.
Clunie, T., DeFilippo, M., Sacarny, M., Robinette, P., 2021. Development of a perception system for an autonomous surface vehicle using monocular camera, lidar, and marine radar. In: 2021 IEEE International Conference on Robotics and Automation. ICRA, pp. 14112–14119. http://dx.doi.org/10.1109/ICRA48506.2021.9561275.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297. http://dx.doi.org/10.1007/BF00994018.
DeFilippo, M., Sacarny, M., Robinette, P., 2021. Robowhaler: A robotic vessel for marine autonomy and dataset collection. In: OCEANS 2021: San Diego – Porto. pp. 1–7. http://dx.doi.org/10.23919/OCEANS44145.2021.9705871.
Faggioni, N., Leonardi, N., Ponzini, F., Sebastiani, L., Martelli, M., 2022a. Obstacle detection in real and synthetic harbour scenarios. In: Modelling and Simulation for Autonomous Systems. Springer International Publishing, Cham, pp. 26–38. http://dx.doi.org/10.1007/978-3-030-98260-7_2.
Faggioni, N., Ponzini, F., Martelli, M., 2022b. Multi-obstacle detection and tracking algorithms for the marine environment based on unsupervised learning. Ocean Eng. 266, 113034. http://dx.doi.org/10.1016/j.oceaneng.2022.113034.
Fisher, R.A., 1992. Statistical Methods for Research Workers. Springer New York, New York, NY, pp. 66–70. http://dx.doi.org/10.1007/978-1-4612-4380-9_6.
Geary, R.C., 1954. The contiguity ratio and statistical mapping. Inc. Stat. 5, 115–146. http://dx.doi.org/10.2307/2986645, URL: http://www.jstor.org/stable/2986645.
Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The kitti dataset. Int. J. Robot. Res. 32, 1231–1237. http://dx.doi.org/10.1177/0278364913491297.
Gennarelli, G., Noviello, C., Ludeno, G., Esposito, G., Soldovieri, F., Catapano, I., 2022. 24 ghz fmcw mimo radar for marine target localization: A feasibility study. IEEE Access 10, 68240–68256. http://dx.doi.org/10.1109/ACCESS.2022.3186052.
Ha, J.S., Im, S.R., Lee, W.K., Kim, D.H., Ryu, J.K., 2021. Radar based obstacle detection system for autonomous unmanned surface vehicles. In: 2021 21st International Conference on Control, Automation and Systems. ICCAS, pp. 863–867. http://dx.doi.org/10.23919/ICCAS52745.2021.9649770.
Haykin, S., 1994. Neural Networks: A Comprehensive Foundation, first ed. Prentice Hall PTR, USA.
Helgesen, Øystein Kaarstad, Vasstein, K., Brekke, E.F., Stahl, A., 2022. Heterogeneous multi-sensor tracking for an autonomous surface vehicle in a littoral environment. Ocean Eng. 252, 111168. http://dx.doi.org/10.1016/j.oceaneng.2022.111168, URL: https://www.sciencedirect.com/science/article/pii/S0029801822005753.
Ho, T.K., 1995. Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition. Vol. 1, pp. 278–282. http://dx.doi.org/10.1109/ICDAR.1995.598994.
Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., Yang, R., 2018. The apolloscape dataset for autonomous driving. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 1067–10676. http://dx.doi.org/10.1109/CVPRW.2018.00141.
Huang, I.L., Lee, M.C., Nieh, C.Y., Huang, J.C., 2024. Ship classification based on ais data and machine learning methods. Electronics 13, http://dx.doi.org/10.3390/electronics13010098.
Kruskal, W.H., Wallis, W.A., 1952. Use of ranks in one-criterion variance analysis. J. Amer. Statist. Assoc. 47, 583–621. http://dx.doi.org/10.1080/01621459.1952.10483441.
Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 12697–12705.
Ludeno, G., Esposito, G., Catapano, I., Soldovieri, F., Gennarelli, G., 2023. Short range k-band radar for maritime security. In: 2023 IEEE Conference on Antenna Measurements and Applications. CAMA, pp. 30–33. http://dx.doi.org/10.1109/CAMA57522.2023.10352813.
Martelli, M., Faggioni, N., Ponzini, F., 2022. Detecting and tracking multi-object in real marine environment. In: Proceedings of the International Ship Control Systems Symposium. http://dx.doi.org/10.24868/10707.
Moran, P.A.P., 1950. Notes on continuous stochastic phenomena. Biometrika 37, 17–23. http://dx.doi.org/10.2307/2332142, URL: http://www.jstor.org/stable/2332142.
Nguyen, T., 2023. The challenges of dark ships to the safety and security of commercial shipping and the way forward. Asia- Pac. J. Ocean. Law Policy 8, 310–328, URL: https://brill.com/view/journals/apoc/8/2/article-p310_007.xml.
Parzen, E., 1962. On estimation of a probability density function and mode. Ann. Math. Stat. 33, 1065–1076, URL: http://www.jstor.org/stable/2237880.
Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.
Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X., 2017. Dbscan revisited, revisited: Why and how you should (still) use dbscan. ACM Trans. Database Syst. 42, http://dx.doi.org/10.1145/3068335.
Stanislas, L., Dunbabin, M., 2019. Multimodal sensor fusion for robust obstacle detection and classification in the maritime robotx challenge. IEEE J. Ocean. Eng. 44, 343–351. http://dx.doi.org/10.1109/JOE.2018.2868488.
Thombre, S., Zhao, Z., Ramm-Schmidt, H., Vallet García, J.M., Malkamäki, T., Nikolskiy, S., Hammarberg, T., Nuortie, H., H. Bhuiyan, M.Z., Särkkä, S., Lehtola, V.V., 2022. Sensors and ai techniques for situational awareness in autonomous ships: A review. IEEE Trans. Intell. Transp. Syst. 23, 64–83. http://dx.doi.org/10.1109/TITS.2020.3023957.
Yin, T., Zhou, X., Krahenbuhl, P., 2021. Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11784–11793.
Zaccone, R., 2024. A dynamic programming approach to the collision avoidance of autonomous ships. Mathematics 12, http://dx.doi.org/10.3390/math12101546, URL: https://www.mdpi.com/2227-7390/12/10/1546.
Zhao, L., Fu, X., 2021. A novel index for real-time ship collision risk assessment based on velocity obstacle considering dimension data from ais. Ocean Eng. 240, 109913. http://dx.doi.org/10.1016/j.oceaneng.2021.109913, URL: https://www.sciencedirect.com/science/article/pii/S0029801821012609.