Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Advancing Rotation and Scale-Invariant Trajectory Representations for Maritime and Aviation Mobility Data Analysis

Cristiano Landi

cristiano.landi@phd.unipi.it

University of Pisa

Natalia Andrienko

Fraunhofer Institute for Intelligent Analysis and Information Systems

Gennady Andrienko

Fraunhofer Institute for Intelligent Analysis and Information Systems

Riccardo Guidotti

University of Pisa

Research Article

Keywords: Spatio-tempral data, Knowledge representation and reasoning, Human-centered computing

Posted Date: March 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9056857/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Machine Learning on November 14th, 2026. See the published version at https://doi.org/10.1007/s10994-026-07199-4.

Advancing Rotation and Scale-Invariant Trajectory Representations for Maritime and Aviation Mobility Data Analysis

Cristiano Landi[1,2*], Natalia Andrienko[3,4], Gennady Andrienko[3,4], Riccardo Guidotti[1,2]

[1*]Dept. of Computer Science, University of Pisa, Largo Bruno Pontecorvo, 3, Pisa, Italy.
[2]KDD-Lab, ISTI-CNR, Via Giovanni Moruzzi, 1, Pisa, Italy.
[3]IAIS, Fraunhofer Institute, Schloss Birlinghoven 1, Sankt Augustin, Germany.
[3]City St George's, University of London, London, United Kingdom.

*Corresponding author(s). E-mail(s): cristiano.landi@phd.unipi.it;
Contributing authors: natalia.andrienko@iais.fraunhofer.de; gennadiy.andriyenko@iais.fraunhofer.de; riccardo.guidotti@unipi.it;

Abstract

Spatiotemporal data from connected vehicles, sensors, and GPS-enabled devices increasingly support applications ranging from traffic management to environmental analysis. Whereas earlier studies relied on aggregated flows or coarse-grained movement representations, high-frequency data now enable the analysis of fine-grained individual motion and the geometric properties of trajectories. In movement-constrained settings such as road traffic, geometric detail may be of limited value, but in unconstrained environments such as maritime and aviation mobility data, movement geometry can provide important cues about underlying activities. Despite this potential, geometry-based trajectory analysis remains underexplored and is often constrained by rigid decision rules that limit real-world applicability. This paper introduces RoSITa, a spatiotemporal analytics framework that leverages rotation- and scale-invariant shape signatures derived from a relative Hough transform to cluster and structure subtrajectory patterns for intuitive visual exploration. We demonstrate how RoSITa supports visual inspection to identify similar trajectory segments and show that it outperforms existing approaches on a classification benchmark when the task is driven by movement geometry. Finally, we present two real-world case studies illustrating

its ability to capture 2D and 3D motion dynamics, enable interpretable human-in-the-loop analysis, and facilitate the generation of labeled data for future machine learning applications.

Keywords: Spatio-tempral data, Knowledge representation and reasoning, Human-centered computing

1 Introduction

Spatiotemporal data are now generated continuously by GPS-enabled devices, producing massive streams of movement information across domains such as transportation, ecology, and maritime and aviation monitoring [1, 2]. While earlier datasets were typically sampled at coarse temporal or spatial intervals, recent technological advances enable high-frequency tracking enriched with additional sensors. This shift allows analyses that move beyond positions alone and instead focus on the dynamics and geometry of the movement. In this work, we focus on the latter, as the movement's geometry is particularly interesting in unconstrained environments, where it is not restricted by predefined infrastructure. Examples include vessels at sea, animals in the wild, aircraft, or paragliders. In these settings, the shape of a trajectory often reflects underlying behavior. In maritime applications, events such as port entry, anchoring, or trawling are closely linked to characteristic movement geometries, yet remain difficult to formalize in a consistent, machine-readable way [3]. At the same time, labeled datasets in this domain are limited [4], making supervised learning approaches challenging to deploy reliably.

Recent research increasingly relies on deep learning methods for trajectory analysis [5–8]. Although these models can capture complex patterns and achieve strong predictive performance, they typically require substantial volumes of labeled data and operate largely as black boxes. Their limited interpretability hinders validation, reduces trust, and complicates adaptation to new or evolving scenarios. In contrast, human experts can often recognize meaningful events by visually inspecting trajectories and drawing on contextual knowledge. However, purely manual analysis does not scale and is impractical for large datasets. As a result, existing solutions either lack scalability and reproducibility or depend heavily on labeled data and opaque models, leaving a gap for approaches that are both interpretable and data-efficient.

To address this gap, we propose a machine-learning-assisted framework that enables human experts to analyze unconstrained two- and three-dimensional movements at scale. The central idea is to transform trajectories into a rotation- and scale-invariant representation, allowing explicit comparison of movement geometries independent of orientation and size. This approach is motivated by two considerations: *(i)* whether trajectories or subtrajectories exhibiting the same shape, such as a right turn, should be distinguished solely based on heading direction, and *(ii)* whether movement scale is always a relevant attribute. By abstracting from these factors when appropriate, the representation supports robust comparisons across trajectories and enables the clustering of similar shapes into representative movement

pattern groups. These clusters can then be interactively explored with a human-in-the-loop approach, allowing experts to integrate their knowledge in the system by associating shapes with semantically meaningful events. The validated shapes are then used to detect and monitor corresponding patterns in new trajectories. This iterative process not only supports scalable event detection but also facilitates the generation of high-quality labeled data for subsequent machine learning applications. Our results demonstrate that the framework consistently identifies behaviorally meaningful movement geometries while preserving interpretability and reducing dependence on large labeled datasets.

In [9], Vlachos et al. introduce a scale- and rotation-invariant representation for handwritten scratch indexing based on the Hough Transform [10]. Inspired by this idea, in Section 4 we design a ROtation- and Scale-Invariant Trajectory dAta representation (RoSITa) tailored to spatiotemporal trajectories. The proposed representation generalizes the underlying concept to both two- and three-dimensional movement data, enabling consistent comparison of trajectory shapes independent of orientation and scale. In the same section, we demonstrate how RoSITa enables visual analytics through an interpretable geometric transformation, allowing experts to explore and contrast movement patterns, and how it can serve as a preprocessing step to improve downstream machine learning models' ability to capture shape-based patterns. In Section 5, we introduce a human-in-the-loop framework grounded in the RoSITa representation, enabling experts to interactively explore, assess, and validate characteristic trajectory shapes, and demonstrate its practical applicability through a real-world case study in maritime vessel monitoring. Section 6 further extends RoSITa representation to three-dimensional trajectories, highlighting its flexibility beyond planar movement data. We first evaluate RoSITa on a synthetic dataset designed to test its ability to capture complex spatial patterns. Then, we illustrate a practical application in paragliding flight analysis[1], showing how the representation can reveal characteristic flight shapes and support expert-driven exploration of 3D movement behaviors.

In summary, this work makes four main contributions. First, we introduce a novel trajectory data representation, applied for the first time to GPS movement data. Second, we demonstrate the advantages of this representation both for machine learning, improving shape-based pattern recognition, and for visual analytics, enabling interpretable exploration of trajectories. Third, we develop a real-world use case in maritime vessel monitoring, illustrating practical applicability[2]. Finally, we extend the representation to three-dimensional trajectories and showcase its effectiveness in paragliding flight analysis.

[1] Data extracted from t.ly/TBdMp

[2] Early results of the application of RoSITa for vessel monitoring were published as a demo paper in the ACM International Conference on Advances in Geographic Information Systems [11].

2 Related Work

In this section, we introduce the works that most closely relate to RoSITa. As RoSITa transforms trajectory data into vectorized form, we first outline other feature representations used for trajectory data analysis. In the second part, we outline some challenges and strategies adopted for maritime vessel monitoring.

Trajectory Data Representations.

One of the earliest comprehensive overviews of this research in trajectory data mining is provided by Zheng et al. in [12], who survey such methods and categorize the main tasks into trajectory preprocessing, similarity computation, clustering, classification, and pattern discovery. The survey highlights how trajectory representations play a fundamental role in enabling efficient analysis and machine learning on mobility data. Following the taxonomy proposed in [13] for time series, trajectory data representations can be broadly categorized into feature-based, shapelet-based, dictionary-based, and deep-learning approaches.

Feature-based approaches represent trajectories through handcrafted descriptors extracted from the raw movement signals. Typically, point-wise features such as speed, acceleration, heading direction, or turning angle are computed from consecutive observations and subsequently aggregated using statistical functions (e.g., mean, variance, percentiles) to produce fixed-length feature vectors suitable for machine learning models [14]. In mobility data science, these features are commonly divided into point features, which capture local motion dynamics, and interval features that describe global trajectory properties such as straightness, displacement, or sinuosity. Several works rely on combinations of global and local descriptors [15–21], often computed through sliding windows or spatial partitioning strategies, to better capture temporal variability in movement patterns. Although these representations are interpretable and computationally efficient, their performance strongly depends on the choice of handcrafted features and aggregation strategies, which may overlook subtle local patterns and require substantial domain expertise [15, 19, 22, 22–24].

Shapelet-based approaches aim to automatically discover discriminative subsequences within trajectories that characterize specific behaviors or classes. The concept originates from time-series analysis, where shapelets are defined as short subsequences that maximize class separability based on distance measures [25]. In mobility data analysis, Movelets and Geolet represent the earliest attempts to extend this concept to trajectories by identifying discriminative subtrajectories whose presence or similarity is indicative of a specific class [26–29]. These methods transform trajectories into feature vectors that measure the similarity between each trajectory and a set of discriminative subtrajectories. Shapelet-based representations provide interpretable models since classification decisions can be directly linked to specific movement patterns. However, they also pose computational challenges due to the large search space of candidate subsequences and often require labeled data to identify discriminative patterns [30]. Moreover, traditional approaches are geographically bound, as the patterns they discover may depend on the spatial location from which they were extracted.

Dictionary-based approaches transform trajectories into symbolic sequences by discretizing either the spatial domain or the movement dynamics. One common

strategy relies on spatial tessellation systems such as Geohash [31], which encode geographic regions into symbolic identifiers and allow trajectories to be represented as sequences of spatial tokens [32]. Alternatively, discretization techniques originally developed for time series, such as Symbolic Aggregate Approximation [33], can be extended to trajectories by discretizing latitude and longitude signals independently and combining them into symbolic representations. Once trajectories are converted into symbolic sequences, methods inspired by text mining and natural language processing, such as bag-of-words representations, can be applied to extract movement patterns [29]. Dictionary-based representations can enable efficient indexing and large-scale pattern discovery [34] but depend on discretization schemes that may introduce spatial distortions or reduce geometric precision.

Deep learning approaches aim to learn trajectory representations directly from raw spatio-temporal data using neural architectures. Recurrent neural networks, convolutional neural networks, and transformer-based models have been applied to mobility data for tasks such as trajectory classification, clustering, and transportation mode detection [35, 36]. Some approaches convert trajectories into grid-based or image-like representations to leverage convolutional architectures, while others treat trajectories as sequences of spatial tokens processed through sequence models [37].

The work by Kontopoulos et al. [38] is the most closely related to RoSITa. In that study, the authors introduce TraClets, a method that converts trajectories into fixed-size images for processing by a CNN classifier. The approach first normalizes each trajectory within a bounded grid, allowing trajectories with different spatial scales to be mapped to the same image space. Individual trajectory points are then encoded as pixels, with color indicating the object's speed within predefined discrete ranges. To preserve motion continuity, consecutive points are connected using the Bresenham line algorithm [39], where the color of each line encodes the acceleration between the two positions. The final image representation captures trajectory shape together with speed and acceleration information, allowing CNNs to learn discriminative visual patterns of motion. However, this representation has two main limitations that we address with RoSITa. First, the resulting images are highly sparse. Second, when different segments of a trajectory overlap in the image space, the corresponding information can be overwritten, leading to a loss of motion details.

Vessel monitoring and abnormal maritime behavior detection.

Maritime surveillance plays a central role in ensuring safety and security in the maritime domain, particularly in response to threats such as piracy, illegal fishing, narcotics trafficking, human trafficking, and weapons smuggling. Coastal states rely on surveillance systems to monitor vessel activity and assess the risks posed by these threats. Modern surveillance increasingly relies on automated approaches due to the growing scale and complexity of maritime operations and the evolving tactics vessels use to evade detection [40]. Machine learning techniques have therefore become widely used for detecting abnormal vessel behaviors in maritime monitoring systems. Existing research typically analyzes vessel movement data to identify deviations from normal activity patterns and detect potential threats. Supervised learning approaches rely on labeled datasets to train models that detect predefined abnormal behaviors or

intents. These methods have been widely applied to identify relatively simple movement abnormalities and often achieve high detection accuracy when labeled data are available. However, obtaining reliable labeled maritime data is difficult because annotating vessel behavior requires domain expertise, and ground truth is limited. For this reason, many studies adopt unsupervised learning techniques that detect anomalies from unlabeled data by extracting features to identify, for example, deviations from typical vessel movement areas or movement dynamics [41–44].

In this context, trajectory analysis is a fundamental component of maritime monitoring, as abnormal vessel behavior often manifests as unusual movement patterns. RoSITa positions itself within this research direction by focusing on the analysis and comparison of vessel trajectory shapes, without the need to define specific task-related features, which may be suboptimal and damage the reusability of the approach, and without using opaque deep learning approaches. By representing and clustering sub-trajectory geometries, RoSITa enables the discovery of recurrent movement patterns that domain experts can inspect to identify meaningful maritime events.

3 Background

This section introduces the fundamental concepts required to understand the contributions. We begin by defining the notation that establishes a common language and formal framework for the key elements discussed throughout the paper. We then describe the Hough Transform (HT) [10] and the rotation- and scale-invariant representation proposed in [9], which provides the foundation for the present work.

We begin by defining the simplest type of trajectory data, namely the position of an object in 2-dimensional space over time. Formally:

Definition 1 (Trajectory)**.** A *trajectory* X is a sequence of spatio-temporal points $X = \{(lat_0, long_0, t_0), \ldots, (lat_m, long_m, t_m)\} \in \mathbb{R}^{m\times 3}$ where the spatial vectors $\vec{x}_j = (lat_j, long_j)$ are sorted by increasing timestamp t_j, i.e., $\forall j \in [1, m)$ we have $t_j < t_{j+1}$.

This type of data is typically not sampled at a constant rate, as observations are often recorded irregularly due to sensor limitations, energy-saving policies, communication constraints, or event-driven acquisition processes. Additionally, trajectories into a dataset often have a different number of observations, so we denote by $\mathcal{X} \in \mathbb{R}^{n\times m\times 3}$ a set of n trajectories of length *at most* m. As in the time-series domain, many trajectory analysis tasks can be formulated as a two-phase process: first, the data are transformed into a tabular vector representation, then analyzed using classical data analysis methods. For example, consider the definition of the trajectory clustering task:

Definition 2 (Trajectory Clustering)**.** Given a trajectory dataset $\mathcal{X}$, *Trajectory Clustering* is the task of defining a function *tclu* from the space of possible input trajectories $\mathcal{X}$ to an output $c \in \mathcal{C}$ denoting its cluster identity.

We can define the clustering function *tclu* as the composition of a vectorization function v, which maps each trajectory to a fixed-size feature vector, and a function

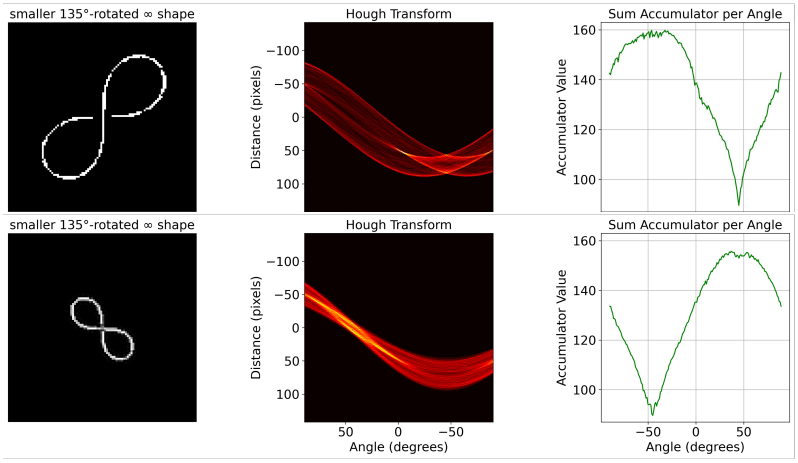


Fig. 1: Overview of shape signature extraction. Left: Example of a trawling vessel trajectory. Center: Hough transform of the subtrajectory, where each sinusoid corresponds to a trajectory point. Right: shape signature obtained by vertically summing the Hough accumulator.

clu, which maps $v(\mathcal{X})$ to one cluster. Thus, given a trajectory X, the output is $tclu = (clu \circ v)(X)$. This formulation can also be applied to other tasks, such as regression or classification, by adjusting *clu* accordingly.

The Hough Transform (HT) [10] is a feature-extraction technique typically used in computer vision to detect geometric shapes by mapping points from Cartesian space into a parameter space. First, given an image, edge detection is applied to extract salient contour points that represent potential geometric structures. For line detection, each contour point in the image, i.e., Cartesian space, votes for all possible lines passing through it, typically parameterized in polar coordinates by (ρ, θ): ρ is the perpendicular distance from the origin to the line, and θ is the angle of the line normal. High values in this accumulator space correspond to dominant linear structures. The voting process is governed by the equation $\rho = x\cos\theta + y\sin\theta$, which maps each point (x, y) into a sinusoidal curve in the (ρ, θ) space. The Hough accumulator is a matrix of size $N_\rho \times N_\theta$, where N_ρ and N_θ are the number of discretized bins for ρ and θ.

In [9], Vlachos' index represents each shape as a vector of N_θ elements, obtained by squaring the Hough accumulator values and summing them by column. Figure 1 illustrates this process: on the left is reported a visual representation of a movement pattern identifying trawling vessels; in the center, the HT is shown, where each sinusoid corresponds to a point of the subtrajectory; on the right, the shape signature is obtained by squaring the accumulator's values and summing them by column. Please note that, while the scale of the trajectory does not affect the shape signature, rotation results in a shift along the θ axis. This vector representation is then used to compute pairwise similarities between shapes through the L_2 norm, while achieving rotation invariance by considering circular shifts along the θ axis and selecting the minimum distance over all possible alignments. To further reduce computational cost,

the signatures are transformed into the Fourier domain and only the magnitudes of the coefficients are retained, yielding a compact, rotation-invariant descriptor that enables efficient comparison and indexing.

Finally, we introduce the function *bestfit*, employed in GEOLET to quantify the similarity between a trajectory X and a subtrajectory S. Conceptually, this operation aligns with the geometric approach of ROSITA: it evaluates how well the shape of S fits within X by considering all possible relative positions. Specifically, *bestfit* computes the minimal distance between S and all subtrajectories of X obtained by sliding a window of size $|S|$ along X, while normalizing each segment to a common reference point. Formally,

$$bestfit(X, S) = \min_{j=0}^{m-|S|} \{d\big(shift(X_{j:j+|S|}), shift(S)\big)\} \tag{1}$$

where $shift(X)$ translates the trajectory so that its starting point is at the origin: $shift(X) = \vec{x}_0 - \vec{x}_0, \ldots, \vec{x}_m - \vec{x}_0$. By focusing on relative geometry rather than absolute positions, this function emphasizes shape-based similarity, consistent with the core principles of ROSITA. Additionally, it has been demonstrated [34] that such a similarity measure captures both geometric and kinematic aspects of trajectories, although it requires quadratic computational complexity.

4 Rotation- and Scale-Invariant Representation

This section is organized into four parts. We begin by formally introducing ROSITA, the rotation- and scale-invariant trajectory representation at the core of our framework. Inspired by [9], it builds upon the original Hough transform formulation while investigating alternative strategies for scaling and compressing the accumulator. In contrast to prior work, our focus is on data mining applications and the analysis of GPS trajectories in both two- and three-dimensional spaces. As detailed in the following sections, our goal is to apply such representation in various mobility data analysis applications ranging from on land traffic to sea vessels and ending to paragliding. Next, we illustrate how the ROSITA representation supports the visual inspection of geometric structures within trajectories. We then conduct a sensitivity analysis on a synthetic dataset designed to examine the impact of ROSITA parameters and provide practical guidance for their selection. Finally, through experiments on both synthetic and real-world data, we show that ROSITA transforms trajectories into a more structured representation that is well-suited for conventional machine learning methods.

As mentioned in the introduction, the proposed representation is motivated by two key considerations. First, certain events, such as harsh stops or sudden accelerations, are significant hazards regardless of the direction of travel. Second, the relevance of movement scale depends on the context: for road-constrained human mobility, scale is critical—fast turns onto a residential street are far more hazardous than similar maneuvers on a highway—whereas for unconstrained motion, such as animal movement or open-field human motion, scale may be less informative and can often be disregarded. To illustrate this, we consider maritime vessel trajectories, where scale and rotation are often not critical. Efficient maritime vessel monitoring is critical for

Algorithm 1: RoSITa: Scale- and Rotation-Invariant Trajectory Representation

Input: Trajectory X, radial bins N_ρ, angular bins N_θ, non-linear transform t, aggregation operator agg

Output: Invariant representation vector $\vec{a}$

1 **Function** `RoSITa`$(X, N_\rho, N_\theta, t, agg)$**:**

2 $nbins \leftarrow N_\rho/\sqrt{2}$ `// Spatial resolution for trajectory rescaling`

3 $\tilde{X} \leftarrow \text{scale}(X, nbins)$ `// Rescale` X `to resolution` $nbins$

4 $A \leftarrow \text{Hough}(\tilde{X}, N_\rho, N_\theta)$ `// Compute Hough accumulator`

5 $m \leftarrow |X|$ `// Number of trajectory points`

6 $\tilde{A} \leftarrow t(A/m)$ `// Normalize and apply non-linear transform`

7 $\vec{a} \leftarrow agg(\tilde{A})$ `// Aggregate along` ρ `to obtain a vector of size` N_θ

8 **return** $\vec{a}$

ensuring navigational safety, protecting marine environments, and supporting regulatory compliance [45]. Events such as entering or exiting a port, anchoring, or engaging in specific activities like trawling are commonly of interest, yet difficult to formally define in a consistent, machine-readable way [3]. Most machine learning approaches require large volumes of labeled data to reliably detect such patterns, yet labeled datasets in the maritime domain remain limited [4].

Algorithm 1 summarizes how, given a trajectory X, RoSITa derives a scale- and rotation-invariant representation. Given the dimensions of the Hough accumulator, N_ρ and N_θ, the first step is to compute the target spatial resolution r used to rescale the trajectory (lines 2–3), ensuring that the latitude and longitude coordinates fit optimally within the accumulator. Although rescaling does not affect the final invariant representation, it allows for a more effective use of the accumulator's resolution. The rescaled trajectory is then used to compute the Hough accumulator A (line 4), capturing the distribution of points in terms of distance and angle. To account for trajectories of varying numbers of observations, the accumulator is normalized by the number of points m. Finally, a non-linear transformation t is applied to the normalized accumulator, and it is aggregated along the radial dimension to produce a compact vector $\vec{a}$ of size N_θ, which encodes the dominant directional patterns of the trajectory. The resulting representation is invariant to scale and, up to a cyclic shift in the vector, to rotation, making it suitable for subsequent comparison, clustering, or classification tasks. Measuring distance between vectors while accounting for cyclic shifts could be done via brute-force rotation, but this is computationally expensive, with complexity quadratic in the vector length. To reduce this overhead, researchers in [9] propose using the magnitude of the (Fast) Fourier Transform (FT). By representing the signal in the frequency domain rather than the spatial domain, the FT effectively removes phase information associated with cyclic shifts, allowing the Euclidean distance to be computed directly on the transformed vectors. The computational complexity of computing the Hough accumulator is $O(mN_\rho)$, the FT requires $O(m \log m)$, and the

Euclidean distance computation is linear in the vector length N_ρ. Under the assumption that $N_\rho = 180$ and $N_\rho \ll m$, the overall asymptotic complexity is dominated by the Fourier transform and thus is $O(m \log m)$.

While in [9] the proposed approach is designed for indexing shapes in a database, $O(m \log m)$ may be too expensive for data-intensive applications designed to run in real time. To this end, we introduced two distance measures that do not use the FT and can be computed in linear time, thereby reducing the computational complexity of RoSITa to linear in the trajectory length m. The first distance function aims to capture the overall magnitude differences (OMD) and, given two vectorized trajectories $\vec{a}$, $\vec{b}$ can be formally defined as follows:

$$\text{OMD}(\vec{a}, \vec{b}) = \left| \sum_{i=0}^{|\vec{a}|} \vec{a}_i - \vec{b}_i \right| \tag{2}$$

The second function circularly shifts both $\vec{a}$ and $\vec{b}$ so that their maximum values are aligned at the first position, and then computes the Euclidean distance. We denote this distance as MED.

At this point, we obtained a scale-invariant representation and a rotation-invariant distance measure. To capture the temporal aspect of trajectory data, a sliding window with a fixed time-based step size is sufficient. The final representation of a trajectory X is the concatenation of all the representations of X over a sliding window with a constant temporal step. This transformation has several notable properties: (i) RoSITa representation can be computed in linear time with respect to the number of observations in X; (ii) the size of the accumulator matrix does not depend on the number of observations in X; (iii) the representation is invariant to shape and scale; and (iv) the resulting shape signature can be visually inspected.

In the remainder of this section, we first present qualitative insights derived from a visual inspection of the proposed representation. We then perform an exploratory analysis of alternative design choices, including non-linear transformations of the Hough accumulator, aggregation strategies, and distance measures for comparing representations. Finally, we evaluate the effectiveness of the proposed representation on multiple supervised mobility datasets.

4.1 RoSITa for Visual Inspection

Figure 2 demonstrates how an infinity-shaped trajectory, typical of vessels performing trawling activities, can be decomposed and visually analyzed through using this representation. The 30-minute trawling path is partitioned into six non-overlapping 5-minute slices, "Slice 1" to "Slice 6", each isolating a distinct phase of motion.

In Figure 2b, RoSITa is applied to each slice, producing a matrix where rows represent slices and columns represent angles. The intensity of the heatmap corresponds to the values of each shape signature. Slice 2 and Slice 5 both exhibit strong, narrow responses around 130° and 45°, capturing the straight connecting strokes between the two lobes of the infinity path. These directional peaks appear symmetrically on opposite sides of the plot, reflecting a mirrored traversal of the same structure. Similar

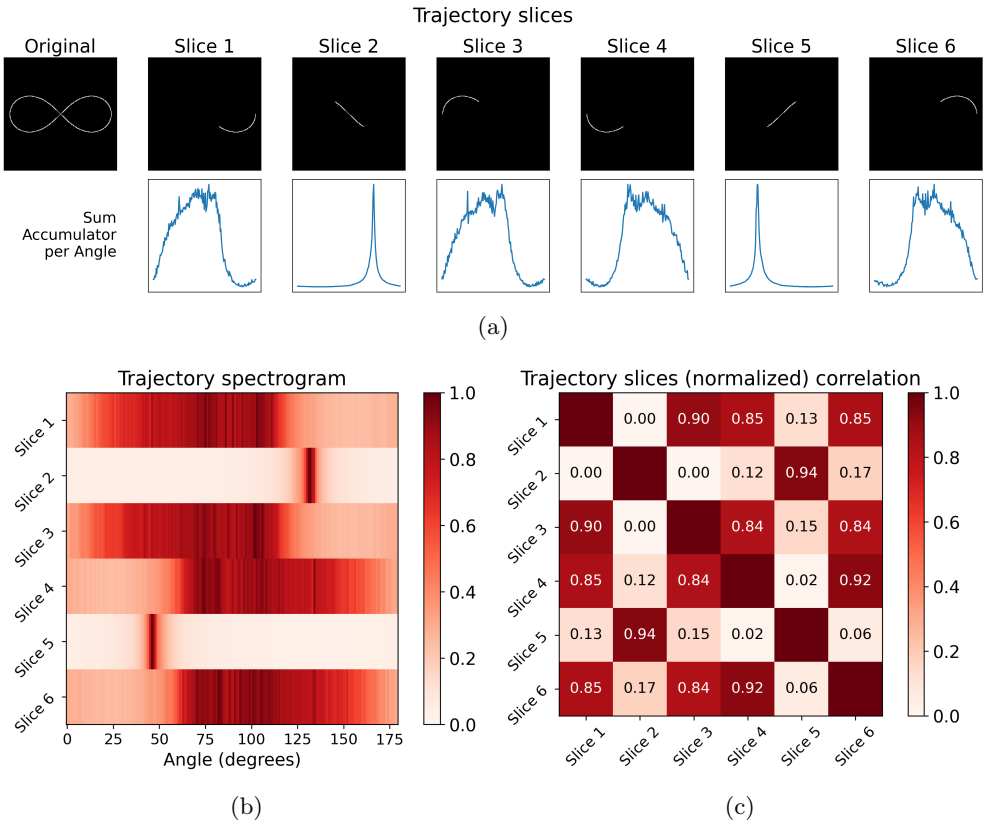


Fig. 2: Example on how RoSITa representation can be used for visual inspection.

correspondences also appear between Slice 1 and Slice 3, and between Slice 4 and Slice 6, indicating that the vessel retraces comparable geometric paths. Although this representation itself does not encode time, its integration with a sliding window enables the detection of temporal recurrence within the trajectory. Consider speed: if the vessel slows down while tracing one lobe of the path and speeds up on the return, each time-based window will capture different spatial extents. A slower segment produces a shorter spatial shape within the same time window, while a faster segment spans a longer one. Consequently, their shape signatures will differ and align only with segments of similar geometry and speed, allowing the method to capture not just spatial repetition but also variation in movement intensity. To better visualize similarities between slices, Figure 2c presents the similarity between all slices. We compute the similarity using MED. Additionally, we scaled the values in the matrix to the range [0, 1], where 1 indicates the maximum similarity (MED is 0). As expected, slices with the same shape exhibit higher similarity regardless of their placement or orientation. The figure also shows that the proposed representation effectively distinguishes between

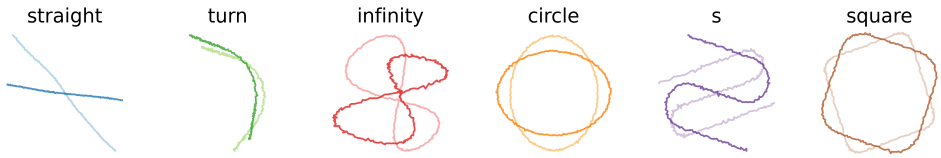


Fig. 3: Synthetic 2-dimensional shapes dataset.

different shapes: while Slice 2 and Slice 5 are highly similar, they show low correlation with the other slices.

This demonstrates that the proposed representation not only captures the geometric structure of the trajectory but also, through a sliding window, reveals temporal patterns and variations in motion, enabling precise identification of repeated maneuvers and their intensity differences. This example further highlights how the RoSITa representation can support visual inspection, showing how trajectory patterns can be intuitively explored within the RoSITa feature space.

4.2 RoSITa: Sensitivity Analysis for 2D Trajectories

As described in the previous section, RoSITa representation is parametrized by the number of radial and angular bins N_ρ and N_θ, a non-linear transformation t, and an aggregation operator agg. Additionally, to ensure complete rotation invariance when comparing two vectorized trajectories, it's necessary to use a distance measure that's insensitive to circular shifts.

To better isolate the effects of these parameter combinations, we designed a synthetic dataset comprising six basic trajectory shapes (Figure 3): *straight*, *turn*, *infinity*, *circle*, *S-shaped*, and *square*. These shapes, inspired by typical vessel movements [46], were selected to span a range of geometric complexities and curvature patterns. Straight and turn trajectories represent common maneuvers, while curvilinear patterns such as infinity, circle, and S-shaped movements mimic looping or oscillatory behaviors often seen in throwing activities. The square-like trajectory is included as a deliberately challenging case to stress-test the representation, as its combination of straight segments and corners can easily be confused with right turns, circles, or S-shaped movements. Because the vertical aggregation of the accumulator's columns discards positional information, trajectories with different spatial arrangements, such as a single straight line versus two parallel straight lines, can produce very similar vectorized representations, although this effect is partially mitigated by the non-linear transformation applied prior to summation. The straight trajectory consists of a primarily linear path, the right-turn trajectory begins with a straight segment followed by a smooth quarter-circle curve, the infinity or figure-eight forms two interconnected loops, circular trajectories are smooth closed loops, S-shaped trajectories combine arcs and short straight segments into a serpentine curve, and the square-like trajectory traces four straight edges connected by small rounded corners, which can be easily confused with right-turn, circle, or S-shaped movements. For each trajectory type, multiple instances were generated by varying parameters such as scale, orientation, curvature, segment lengths, and starting positions, while the Gaussian noise

level remained constant across all instances to simulate natural variability. We created such a dataset by generating 1000 trajectories for each pattern, for a total of 6000 distinct trajectories. Each trajectory is composed of 200±50 points on average. We also generate a smaller dataset comprising only 20 trajectories per pattern. This reduced dataset simulates a data-scarce scenario and allows us to assess whether the representation space remains discriminative even when the variability between training and test samples is substantially higher.

The objective of this sensitivity analysis is to assess how different choices of non-linear scaling functions t, aggregation functions agg, and vector-wise distance functions d affect the performance of a machine learning model built upon the resulting representations. To this end, the machine learning task is formulated as classification, with the target being the prediction of shape type. The performance is evaluated using a k-nearest neighbors (KNN) classifier with $k = 1$. This choice is motivated by the fact that 1NN relies directly on the underlying distance measure, without introducing additional model parameters or learning dynamics, thereby isolating the effects of the proposed design choices in the representation space. As non-linear transformations t, we consider the logarithmic, square, and cubic root functions, as well as the identity mapping, in which the accumulator remains unchanged. For column-wise accumulator aggregation agg, we evaluate several statistical operators: sum, mean, standard deviation, variance, maximum value, and count of non-zero elements. For the distance function d, following [9], we first employ the Fast Fourier Transform to map each shape descriptor into the frequency domain and compute the distance on the magnitude spectrum, which is invariant to circular shifts. We denote such a strategy using FFT. In addition to the novel distance measure proposed, we also experimented with the Earth Mover's Distance (EMD), which quantifies the minimal cost of transforming one distribution into another and provides robustness to shifts and distributional variations. For 1-dimensional vectors, as in our setting, the computational complexity is $\mathcal{O}(N_\rho \log N_\rho)$. Finally, as a baseline, we compute the Euclidean distance ED without any prior normalization.

We evaluated multiple accumulator resolutions, considering $N_\rho \in 36, 90, 180$ and $N_\theta \in 204, 339, 509, 679, 1018, 1527$.[3]. The dataset was partitioned into a 70-30 train-test split. Model selection was performed via 10-fold cross-validation on the training set, choosing the configuration that maximized the macro F1 score. As no substantial performance variations were observed across accumulator sizes, we fixed $N_\rho = 180$ and $N_\theta = 200$ for the subsequent results exploration.

Table 1 reports the macro F1 scores for all combinations of t, agg, and d. For readability, each cell is shaded on a red-to-green scale, where red corresponds to a score of 0 and green to 1. We also computed the average F1 score for each distance function across all combinations of t and agg. Across both datasets, two main trends emerge. First, all configurations employing the OMD distance perform consistently worse than the alternatives. Second, omitting non-linear scaling and aggregating columns by mean or sum yields comparatively lower performance. On the full version of the generated dataset, performance remains similar across most hyperparameter combinations.

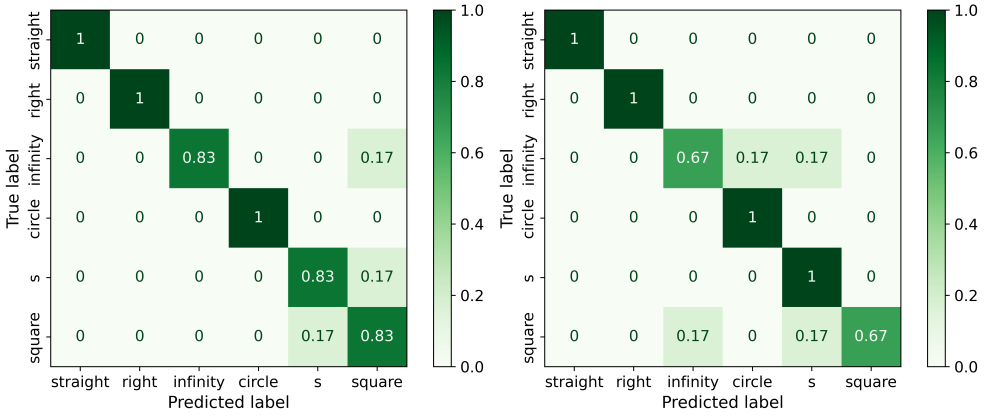
[3]For N_ρ, these values correspond to equal-size bins of approximately 5°, 2°, and 1°, respectively. The values of N_θ are computed as the diagonal (hypotenuse) of square matrices corresponding to common image resolutions, namely $144, 240, 350, 480, 720, 1080$

		large 2D dataset						small 2D dataset					
		log	$\sqrt[3]{}$	id	$\sqrt{}$	^{2}	max by d	log	$\sqrt[3]{}$	id	$\sqrt{}$	^{2}	max by d
FFT	#nonzero	0.936	0.936	0.936	0.936	0.936		0.583	0.583	0.583	0.583	0.583	
	max	0.975	0.955	0.976	0.959	0.989		0.833	0.833	0.833	0.833	0.861	
	mean	0.926	0.979	0.245	0.985	0.998	0.999	0.389	0.750	0.278	0.750	0.833	0.889
	std	0.998	0.962	0.998	0.938	0.999		0.750	0.639	0.778	0.500	0.889	
	sum	0.999	0.963	0.204	0.977	0.999		0.806	0.639	0.250	0.667	0.806	
	var	0.998	0.961	0.998	0.941	0.963		0.833	0.639	0.861	0.528	0.806	
	mean by d	0.919±0.190						0.683±0.172					
OMD	#nonzero	0.268	0.268	0.268	0.268	0.268		0.306	0.306	0.306	0.306	0.306	
	max	0.409	0.382	0.429	0.374	0.421		0.472	0.333	0.389	0.306	0.472	
	mean	0.333	0.303	0.305	0.275	0.327	0.429	0.306	0.333	0.278	0.278	0.361	0.472
	std	0.307	0.211	0.324	0.208	0.379		0.333	0.194	0.361	0.250	0.278	
	sum	0.394	0.297	0.265	0.309	0.411		0.417	0.306	0.306	0.250	0.417	
	var	0.357	0.200	0.353	0.212	0.415		0.306	0.222	0.389	0.250	0.361	
	mean by d	0.318±0.069						0.323±0.066					
ED	#nonzero	0.959	0.959	0.959	0.959	0.959		0.556	0.556	0.556	0.556	0.556	
	max	0.945	0.928	0.944	0.932	0.911		0.556	0.583	0.556	0.583	0.444	
	mean	0.722	0.979	0.253	0.984	0.989	0.994	0.361	0.694	0.250	0.639	0.583	0.639
	std	0.993	0.887	0.993	0.962	0.960		0.611	0.389	0.611	0.444	0.556	
	sum	0.994	0.974	0.309	0.975	0.993		0.639	0.556	0.139	0.528	0.639	
	var	0.989	0.888	0.990	0.963	0.907		0.583	0.389	0.583	0.444	0.500	
	mean by d	0.905±0.178						0.521±0.120					
MED	#nonzero	0.919	0.919	0.919	0.919	0.919		0.500	0.500	0.500	0.500	0.500	
	max	0.978	0.971	0.978	0.972	0.961		0.833	0.778	0.833	0.778	0.694	
	mean	0.756	0.970	0.245	0.977	0.990	0.994	0.389	0.556	0.250	0.556	0.750	0.833
	std	0.992	0.876	0.993	0.938	0.993		0.750	0.417	0.722	0.583	0.750	
	sum	0.986	0.944	0.219	0.950	0.994		0.694	0.583	0.139	0.611	0.750	
	var	0.990	0.874	0.989	0.938	0.939		0.750	0.417	0.750	0.611	0.667	
	mean by d	0.900±0.188						0.604±0.171					
EMD	#nonzero	0.870	0.870	0.870	0.870	0.870		0.500	0.500	0.500	0.500	0.500	
	max	0.943	0.924	0.943	0.928	0.951		0.806	0.833	0.806	0.833	0.861	
	mean	0.916	0.952	0.384	0.962	0.984	0.984	0.306	0.667	0.278	0.722	0.861	0.917
	std	0.982	0.913	0.982	0.863	0.974		0.806	0.694	0.778	0.417	0.861	
	sum	0.982	0.927	0.547	0.943	0.982		0.778	0.583	0.556	0.583	0.778	
	var	0.984	0.912	0.983	0.864	0.952		0.833	0.694	0.833	0.417	0.917	
	mean by d	0.901±0.128						0.667±0.182					

Table 1: Sensitivity analysis varying t, *agg*, and d. Cells show macro F1 scores and are shaded using a red-to-green scale, where red corresponds to a score of 0 and green to a score of 1.

Notably, strong results obtained with the standard Euclidean distance, despite its sensitivity to circular shifts, indicate limited rotational variability between training and test samples. This interpretation is supported by the reduced dataset results, where ED ranks among the lowest-performing distances. For the remaining distances, reduced training data and increased variability between training and test sets correspond to lower average scores. Overall, FFT and EMD achieve the highest performance. While FFT exhibits slightly higher average F1 scores and lower variance than EMD, the latter attains the best single configuration when the accumulator values are squared and aggregated using the variance.

To further analyze the limitations of the proposed trajectory representation, Figures 4a and 4b show the confusion matrices of the best- and second-best-performing models on the reduced dataset. The top-performing configuration, achieving a macro



(a) (b)

Fig. 4: Confusion matrices corresponding to the best-performing model (left) and the second-best-performing model (right).

F1 score of 0.917, squares the accumulator values, aggregates them via column-wise variance, and computes distances using EMD. The second-best configuration, with a macro F1 score of 0.889, also applies the squaring transformation, but aggregates the accumulator by column-wise standard deviation and computes distances using FFT. From the confusion matrices, it can be observed that both models achieve perfect classification for straight, right-turn, and circular trajectories. Most misclassifications involve square, s-shaped, and infinity-like trajectories. In the best-performing model, infinity trajectories are occasionally confused with square ones, while square trajectories are sometimes predicted as s-shaped and vice versa. In the second-best configuration, the confusion becomes more pronounced: infinity trajectories are misclassified as circular and s-shaped, and square trajectories are more frequently confused with both infinity and s-shaped classes. These results indicate that geometrically similar and self-intersecting patterns represent the main challenge for the proposed representation, particularly when rotational variability is higher.

4.3 Quantitative Evaluation in Classification Setting

This final part of the section benchmarks the proposed representation against a set of state-of-the-art competitors from the mobility and time series domains. We first introduce the supervised datasets used and the selected competitors. Next, we describe the experimental setup. Finally, we present the classification outcomes.

The datasets are divided into two categories: real-world and synthetic (and simulated) datasets. Synthetic datasets, proposed in [34], are included to address the scarcity of publicly available, fully compatible labeled trajectory datasets. Within the *real-world datasets*, we selected 3 datasets[4] typically used for GPS trajectory classification. The datasets differ in size, semantics, and classification objectives, i.e., `animals`,

[4] `animals` and `vehicles`: t.ly/klP5M, `seabirds`: t.ly/LpMx3, `geolife`: t.ly/FbqVa,

`vehicles`, and `geolife`. Table 2 (top three rows) reports descriptive statistics for the datasets, including the number of trajectories, the total number of observations, the average and standard deviation of observations per trajectory, the average sampling rate, and the number and proportion of target classes. For `animals`, the classification task consists in recognizing three different species. For `vehicles`, the objective is to distinguish between buses and trucks. For `geolife`, we set the classification problem to distinguishing between public and private modes of transport [29].

For *synthetic datasets*, we use an extended version of those introduced in [34]. These datasets, originally designed to test shapelet-based trajectory transformations, aim to isolate cases where the classification task depends either on movement dynamics or on trajectory shape patterns. The authors divide the dataset into synthetic and simulated. Synthetic datasets are deliberately simple to isolate specific properties of the representation under analysis and aim to minimize confounding variables, such as traffic, object interactions, and terrain priorities, that may also affect simulated datasets. The simulated datasets, which are more similar to real-world data but avoid problems such as signal loss or noise, are generated using SUMO [47]. Specifically, the authors used SUMO to simulate cars and bicycles in three global regions with distinct road networks: Rome (Italy), characterized by a curvilinear loop network; New York (USA), featuring a grid-based network; and Athens (Greece), with a combination of curvilinear and grid networks.

The authors partition the datasets into three categories according to the relationship between trajectory characteristics and the target class. The first category, where RoSITa is expected to perform well, includes datasets in which the label depends on trajectory geometry. The synthetic dataset `SR` contains objects moving either along a straight line or performing a 90-degree right turn Each object in this dataset travels at a constant speed for precisely 100 seconds. The dataset `Yao`, inspired by [37], introduces three motion patterns. It includes three distinct movement patterns: straight, circling, and bending. Unlike the `SR` dataset, this dataset introduces additional complexity in the prediction task. Each object travels in a different direction, with a variable sampling rate for a varying number of seconds between 70 and 130. Additionally, four SUMO-based datasets are generated by composing vehicle trajectories in three cities, with the objective of identifying the region in which an object moves: Rome vs. New York (`Ro-NY`), New York vs. Athens (`NY-At`), Athens vs. Rome (`At-Ro`), and Rome vs. New York vs. Athens (`Ro-NY-At`). The second category includes datasets where the predictive task depends exclusively on kinematic properties, a setting in which RoSITa is expected to be less effective due to its geometry-oriented design. The synthetic dataset `FS` contains trajectories that all perform a 90-degree right turn but at two different speeds. Similarly, four simulated datasets are constructed to distinguish between cars and bicycles within individual regions and across all regions: cars and bikes in Rome (`CB-Ro`), New York (`CB-NY`), Athens (`CB-At`), and all three cities combined (`CB`). The third category comprises datasets in which both geometry and kinematics determine the target class, obtained by combining the previous settings. The synthetic dataset `SR-FS` defines four classes: moving straight fast, turning right

		#trj	#obs	#obs per trj	Avg Δtime (s)	Classes
Real-world	`Animals`	102	15k	145 ± 62	4.85 ± 4.85	Cattle 33% Deer 29% Elk 37%
	`Vehicles`	381	178k	467 ± 251	122 ± 160	Bus 28% Truck 72%
	`GeoLife`	5977	5.3M	892 ± 2.5k	265 ± 2.9k	Public T. 43% Private T. 57%
Synthetic	`SR`	200	$20k$	100 ± 0.0	1.0 ± 0.0	Straight 50% Turn 50%
	`FS`	200	$20k$	100 ± 0.0	3.0 ± 2.0	Slow 50% Fast 50%
	`SR-FS`	200	$20k$	100 ± 0.0	3.0 ± 2.0	Straight Fast 25% Straight Slow 25% Turn Fast 25% Turn Slow 25%
	`Yao`	300	$21k$	69 ± 12	1.5 ± 0.7	Straight 33% Circling 33% Bending 33%
SUMO-simulated	`Ro-NY`	2390	932k	390 ± 245	1.0 ± 0.0	Cars in Rome 50% Cars in NY 50%
	`At-Ro`	2032	658k	323 ± 180	1.0 ± 0.0	Cars in Athens 50% Cars in Rome 50%
	`NY-At`	2032	818k	403 ± 248	1.0 ± 0.0	Cars in NY 50% Cars in Athens 50%
	`Ro-NY-At`	3048	1.1M	370 ± 228	1.0 ± 0.0	Cars in Rome 33% Cars in NY 33% Cars in Athens 33%
	`CB-NY`	4622	2.2M	474 ± 288	1.0 ± 0.0	Bikes in NY 50% Cars in NY 50%
	`CB-Ro`	1172	366k	313 ± 177	1.0 ± 0.0	Bikes in Rome 50% Cars in Rome 50%
	`CB-At`	990	326k	289 ± 182	1.0 ± 0.0	Bikes in Athens 50% Cars in Athens 50%
	`CB`	2970	1.1M	376 ± 235	1.0 ± 0.0	Bikes 50% Cars 50%
	Ro-NY-At ×`CB`	6882	2.9M	426 ± 267	1.0 ± 0.0	Rome-NY-Athens ×Bikes & Cars (balanced)

Table 2: Datasets metadata.

fast, moving straight slow, and turning right slow. For the simulated data, an analogous composition yields `Ro-NY-At`×`CB`, which includes two vehicle types across three geographical regions, thus jointly capturing spatial and kinematic variability.

As the primary contribution of this work is the introduction of a novel data representation, we focus our experimental comparison on state-of-the-art representation-based baselines from both the mobility and time-series literature. Whenever possible, we select one representative method for each major family of representations, adhering to the standard time-series taxonomy proposed by Middlehurst et al. [13]. In

particular, our evaluation emphasizes interval-based, shapelet-based, and dictionary-based approaches, ensuring a balanced and principled comparison across distinct representation paradigms. For the interval-based family, we consider TIF [14], which extracts discriminative features from randomly sampled trajectory intervals, and CIF [48, 49], a canonical interval-based time-series classifier that builds ensembles over interval-derived features. For the shapelet-based family, we include GEOLET [29] and MOVELETS [26], which learn geographically meaningful subsequences as discriminative shapelets for trajectories. For the dictionary-based family, to the best of our knowledge, no trajectory-specific competitor is currently available. Therefore, we adopt BORF [50] (Bag-Of-Receptive-Fields) from the time-series domain, a dictionary-based time series transformation that extracts SAX words from time series by applying sliding windows, discretizing, and counting the occurrences of the SAX words in the time series. Finally, we introduce the matrix-based family, where the data representation is represented as a structured matrix. Our proposal belongs to this family, as ROSITA encodes trajectories via a structured matrix, i.e., the Hough accumulator. We also include TRACLETS [38], which creates a matrix representation of each trajectory by plotting its shape as an image and using color to indicate speed and acceleration. Moreover, we include ROCKET [51, 52] (RandOm Convolutional KErnel Transform) to benchmark against current state-of-the-art transformation-based methods for time series. Since most time-series methods require sequences of equal length, we pad trajectories by assuming the object is stationary, i.e., repeating the last observed latitude and longitude until all trajectories match in length. We also experimented with passing the time dimension as an additional signal, though this did not yield significant improvements.

Consistently with the sensitivity analysis, and to isolate the effect of the representation itself, we adopt a k-Nearest Neighbor (KNN) classifier with $k \in 1, 3, 5$. Each dataset is split into training and test sets at a 70–30 ratio. For every representation method, hyperparameters are optimized via 5-fold cross-validation on the training set, selecting the configuration that achieves the highest average macro F1-score. For transformation methods and models supporting it, we enabled parallel computation with up to 8 cores. All experiments were executed in parallel on a 256-core server with 1 TB of RAM. Experiments exceeding 24 hours of runtime or 512 GB of memory usage were automatically terminated. In the reported results, we use the abbreviations OT (Out of Time) and OM (Out of Memory) to indicate experiments that were interrupted due to these limits. Additionally, we indicate other algorithmic errors with the symbol IE (internal error).

For each representation, we tested a wide range of hyperparameters. For TIF, we explored all interval variants—defined by the number of observations, time intervals, or distance-based intervals. The number of intervals ranged from 2 to 20, with different constraints on minimum and maximum lengths. We enabled the use of all features supported by the current method implementation[5], excluding latitude and longitude to avoid making position-based classification tasks trivial [14]. For its time series counterpart, CIF, we used the same number and length of intervals to ensure comparability. For GEOLET, we tested different numbers of geographic shpelets, from 5 to 100, using

[5] TIF implementation is available at the following link: t.ly/7RfmV.

		Geometry	Kinematic	Geom. and Kin.	Real-world	All
kNN	TIF	0.847 ± 0.109	$\underline{0.893 \pm 0.171}$	$\underline{0.984 \pm 0.022}$	$\underline{0.739 \pm 0.171}$	$\underline{0.851 \pm 0.150}$
	CIF	0.816 ± 0.131	0.670 ± 0.206	0.814 ± 0.264	0.451 ± 0.347	0.687 ± 0.255
	GEOLET	$\underline{0.852 \pm 0.115}$	$\mathbf{0.922 \pm 0.138}$	$\mathbf{0.990 \pm 0.014}$	$\mathbf{0.779 \pm 0.233}$	$\mathbf{0.872 \pm 0.155}$
	MOVELETS	0.280 ± 0.445	0.470 ± 0.452	0.500 ± 0.707	0.471 ± 0.545	0.407 ± 0.458
	ROSITA	$\mathbf{0.949 \pm 0.061}$	0.552 ± 0.344	0.794 ± 0.291	0.682 ± 0.100	0.744 ± 0.168
	TRACLETS	0.458 ± 0.140	0.378 ± 0.122	0.320 ± 0.339	0.371 ± 0.327	0.398 ± 0.199
	BORF	0.691 ± 0.230	0.507 ± 0.293	0.289 ± 0.409	0.340 ± 0.398	0.507 ± 0.322
	ROCKET	0.750 ± 0.187	0.606 ± 0.179	0.604 ± 0.189	0.384 ± 0.449	0.604 ± 0.281

Table 3: F1-score aggregated for the analysis tasks: geometry-based, kinematic-based, and both. Results are shown for real-world datasets as well as across all datasets. The representation yielding the best model is in bold, and the runner-up is underlined.

a Geohash-based partitioning with precision levels between 3 and 7. To select the most discriminative shapelets, we experimented with several strategies: computing mutual information scores, using medoids from k-Means clustering, selecting random instances from OPTICS clusters, and randomly sampling all possible subsequences. For the distance computation between trajectories and shapelets, we tested Euclidean, interpolated-time, and interpolated-space distances. For MOVELETS, it is only possible to constrain the minimum and maximum length of the subtrajectories. For smaller datasets, such as `Animals`, `Vehicles`, and all synthetic datasets, we did not impose any constraints. For the remaining datasets, we limited the maximum length of the subtrajectories to the natural logarithm of the trajectory with the largest number of observations in the dataset, as suggested by the authors in a later work [26]. Without this constraint, even relatively small datasets like `Seabirds` and `Vehicles` resulted in OT (Out of Time) or OM (Out of Memory) interruptions. For ROSITA, we introduced a time-based sliding window with durations ranging from 10 seconds to 1 hour. The remaining hyperparameters are evaluated using the same configuration described in Section 4.2. We only tested distance functions with linear complexity. For TRACLETS, the only tunable parameter is the trajectory image size, which we ranged from 50×50 to 1080×1080 pixels. For BORF, we tested window sizes from 3 to 20 symbols and alphabets of 4, 8, or 16 symbols, plus one for stationarity. All other parameters were kept at their defaults, except the *complexity*[6], adjusted from quadratic to linear for computational efficiency on larger datasets. Finally, for ROCKET, we used $10k$ kernels, as suggested by the authors.

As described in the dataset presentation, some datasets were designed so that the classification objective depends on either the movement's geometry or specific kinematic features. We leverage this property and empirically assess how well the proposed and competing representations capture these correlations. Table 3 reports the F1-scores of all selected representations, grouped by task category. For each competitor, we also report the mean performance and standard deviation across all datasets, along with the results from the downstream classifiers. For datasets where the representation technique produced a OM error, the scoring values were set to zero. Across nearly

[6]Refer to BORF documentation at t.ly/fd-lN for details.

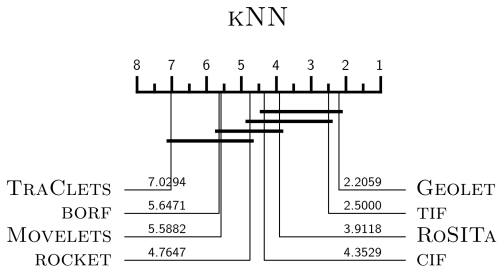


Fig. 5: Comparison of models' rank for the various representations against each other with the Nemenyi test. Representations leading to the downstream model's performance that are not significantly different at 90% confidence are connected. Best ranks on the right.

all dataset groups, the Geolet representation consistently outperforms competing approaches, with tif generally ranking second. On geometry-based datasets, RoSITa achieves particularly strong results, exceeding the runner-up by approximately 10%, which represents the largest margin observed between the top two methods across all categories. Moreover, the relatively low standard deviation indicates greater stability across datasets. These findings suggest that the representation space induced by RoSITa more effectively clusters instances within the same class, whereas alternative methods produce embeddings in which inter-class separation is less clear.

Figure 5 shows the critical difference diagram [53], obtained by ranking each representation's F1 score across datasets. Representations leading to better-performing models are placed to the right. Horizontal lines connect representations whose performance is statistically indistinguishable according to the Nemenyi post-hoc test at the $\alpha = 0.10$ significance level (90% confidence). Although RoSITa ranks third, the diagram indicates that the top four performing representations are statistically indistinguishable. Table 4 shows detailed results for each representation and dataset. The detailed results align with expectations and confirm the discussion on aggregated results: tif, cif, Geolet, and RoSITa emerge as the top-performing representations. However, several notable observations stand out. For instance, RoSITa consistently delivers superior performance on tasks related to the geometry of movement, yet struggles significantly with kinematics-related tasks, such as distinguishing between car and bike trajectories (e.g., in the `CB` dataset). Additionally, Movelets representation faces significant computational limitations, leading to out-of-memory errors for larger datasets such as `GeoLife` and most SUMO-simulated datasets. Despite its high accuracy on smaller datasets such as `Animals` and `Vehicles`, this constraint makes it impractical for large-scale applications. Conversely, the TraClets method consistently underperforms, with poor discriminative power across nearly all benchmarks, suggesting a fundamental inability to capture meaningful trajectory features in the representation space.

Although a performance gap between geometry-based and kinematic-based datasets is expected, given the design of RoSITa, the observed magnitude of this difference indicates potential for further improvement, particularly in better integrating

		TIF	CIF	GEOLET	MOVELETS	ROSITA	TRACLETS	BORF	ROCKET
Real	Animals	0.824	0.730	0.866	0.905	0.573	0.200	0.598	0.678
	Vehicles	0.887	0.718	0.968	0.979	0.770	0.581	0.761	0.858
	GeoLife	0.748	IE	0.842	OM	0.702	0.702	OM	OM
Synthetic	SR	1.000	1.000	1.000	1.000	1.000	0.344	1.000	1.000
	FS	1.000	1.000	1.000	1.000	0.933	0.322	0.510	0.417
	SR-FS	1.000	1.000	1.000	1.000	1.000	0.080	0.578	0.471
	Yao	0.867	0.778	0.978	0.680	0.989	0.641	0.944	0.944
	Ro-NY	0.906	0.870	0.852	OM	0.986	0.575	0.628	0.720
	At-Ro	0.785	0.707	0.739	OM	0.851	0.477	0.567	0.629
	NY-At	0.843	0.897	0.811	OM	0.975	0.449	0.592	0.695
	Ro-NY-At×CB	0.969	0.627	0.980	IE	0.589	0.560	OM	0.738
	CB-NY	0.963	0.632	0.979	0.601	OM	0.577	0.635	0.732
	CB-Ro	0.952	0.641	0.997	0.748	0.583	0.375	0.684	0.747
	CB-At	0.960	0.649	0.956	IE	0.700	0.369	0.704	0.730
	Ro-NY-At	0.680	0.645	0.733	OM	0.895	0.264	0.415	0.513
	CB	0.588	0.428	0.677	OM	0.542	0.247	OM	0.402

Table 4: Summary of experimental results for classification. (macro) F1-score.

kinematic information into the representation. A straightforward strategy, inspired by ensemble approaches in time-series classification, would be to combine RoSITa with complementary representations such as TIF or GEOLET. Alternatively, a hybrid framework could be devised in which RoSITa is first employed to group trajectories with similar geometric patterns, and kinematic-oriented representations are subsequently applied within each shape-consistent cluster.

5 Using RoSITa in Vessel Monitoring

In this section, we propose a human-in-the-loop framework to support experts in the exploration and interpretation of trajectory geometries, and illustrate its application to a maritime monitoring task focused on vessel activity analysis. The framework is motivated by two key observations. First, many maritime activities can be inferred directly from the geometric properties of vessel trajectories [54]. For instance, trawling is typically characterized by slow motion combined with repeated directional changes, producing zigzag, looping, or figure-eight patterns. Second, prior work has shown that shape-based representations can effectively capture both geometric structure and movement dynamics in trajectory data [34]. These insights suggest that trajectory geometry constitutes a meaningful and interpretable signal for event discovery in maritime contexts. Maritime monitoring relies heavily on AIS (Automatic Identification System) data, which provides information such as vessel identity, position, speed, and course, and plays a central role in ensuring navigational safety, environmental protection, and regulatory compliance [45]. Although numerous machine learning techniques have been proposed to analyze AIS data [55, 56], a persistent challenge is the lack of precise and universally accepted definitions of maritime events within such complex and heterogeneous data [3]. Events such as port entry or exit, anchoring, or fishing activities are often of interest but difficult to formalize consistently and in a

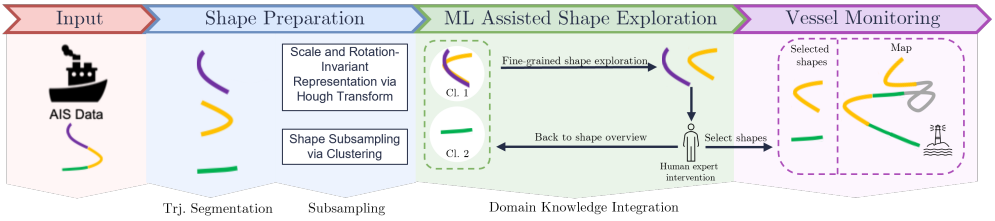


Fig. 6: RoSITa-based Monitoring Framework.

machine-readable manner, and labeled datasets remain limited [4]. While deep learning approaches can model complex patterns [5–8], they often operate as black boxes, limiting interpretability and making validation difficult in low-label settings.

Our framework addresses these challenges by leveraging RoSITa to obtain a robust, scale- and rotation-invariant representation of subtrajectory shapes. Clustering in the resulting representation space groups geometrically similar movement patterns, enabling experts to interactively explore representative shapes and associate them with meaningful maritime events. This human-in-the-loop process combines automated pattern discovery with domain expertise, allowing experts to validate and refine candidate patterns. The validated shapes can then be used to detect and monitor similar events in unseen trajectories, supporting scalable and interpretable maritime analysis [57]. Additionally, iterative expert feedback facilitates the creation of high-quality labeled datasets, further enhancing downstream machine learning applications.

5.1 Framework

Figure 6 illustrates the proposed framework. The framework is structured into three phases: *shape preparation*, *shape exploration*, and *monitoring*. It combines a scale- and rotation-invariant trajectory representation with a distance measure suitable for variable-length comparisons and a set of clustering techniques, enabling the organization of movement patterns into coherent groups that facilitate interactive analysis by domain experts.

During the *shape preparation* phase, the framework partitions input trajectories and extracts a representative subsample of subtrajectories. Trajectories can be segmented using a fixed-length sliding window or spatial tessellation, where splits follow a grid-based partitioning [58]. The choice of method depends on the dataset's characteristics and the expected movement patterns. Each segment is transformed into a rotation- and scale-invariant representation via RoSITa. Similar segments are then grouped using a clustering algorithm adopting one of the distance functions proposed in Section 4. Representative elements are sampled from each cluster[7]. Finally, clusters are sorted to facilitate intuitive user presentation and exploration. In the second phase, *shape Exploration*, domain knowledge is integrated through a human-in-the-loop process. Using the clusters from the previous phase, the expert selects relevant features and interesting shape clusters that align with specific movement patterns. Given the large shape set, the system filters the shapes within the chosen cluster and

[7]For instance, k-means or k-medoids can use the centroid or medoid as a representative subtrajectory.

re-applies clustering based on the selected features, such as speed, acceleration, or trajectory complexity. The expert can iterate between shape-based and feature-based clustering, refining the selection to build a diverse set of shapes representing different vessel behaviors. To label events, the expert links each activity (e.g., trawling, entering port) to one or more representative shapes, creating the event-to-shape map E. During the *shape preparation* phase, trajectory segments are extracted solely based on geometric differences, without considering kinematic attributes. In the subsequent *shape exploration* phase, expert-defined features guide the selection and labeling of representative shapes, forming the event map E. In the *monitoring* phase, E is used to detect events in new trajectories, which requires a similarity function that captures both geometric and kinematic properties. Following [34], we adopt GEOLET's *bestfit* function to match shapes while preserving motion dynamics. To handle invariance to scale and rotation, we define a modified Euclidean distance that minimizes over a scale factor s to optimally align two subtrajectories S_1 and S_2, while the rotation angle θ is inferred from the cyclic shift in the ROSITA representation. The modified Euclidean distance can be defined as follows:

$$d(S_1, S_2) = \min_{\theta \in [-\pi, \pi],\ s>0} \|S_1 - s \cdot R(\theta) \cdot S_2\| \tag{3}$$

where $R(\theta)$ is the 2-dimensional rotation matrix. Please note that the modified Euclidean distance can be avoided if, during the shape exploration phase, the expert selects representative shapes that already cover different scales. Thanks to the framework's structure, the initial clustering based on rotation- and scale-invariant signatures groups such variations together, making it easier for the expert to include them in the event-to-shape map E.

Finally, the framework renders a geographic visualization showing the spatial distribution of detected events, allowing users to observe where specific vessel activities occur. This helps analysts interpret patterns, verify detections, and identify areas of interest or unusual behavior.

5.2 A Case Study for Vessels in the Port of Brittany

This section presents an application of the proposed framework in a case study involving real AIS data from 71 fishing vessels in the ports of Brittany[8]. The first step of the framework involves segmenting the trajectory into subtrajectories of approximately 10 km with a 3km overlap, resulting in a total of 12520 subtrajectories.

For initial clustering of the shape representations, we tested OPTICS[9] [60, 61] and HDBSCAN[10] [62]. OPTICS identified 1232 clusters, while HDBSCAN identified 983. The distance function employed was MED, whose linear computational complexity makes it particularly suitable for efficiently comparing the large number of subtrajectories. To better evaluate the similarity between the results of the two clustering algorithms, we used V-measure [63], which is the harmonic mean of homogeneity and completeness. Homogeneity measures whether each cluster contains only members of a single class,

[8] Dataset available at: https://zenodo.org/records/1167595 [59].
[9] We tested different values of the *eps* and *bic_eps* hyperparameters without significant shifts in the results. We selected the result obtained with default `sklearn` hyperparameters for further analysis.
[10] with a minimum cluster size of 5.

Table 5: Median values per cluster for the three selected features.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
SE (m)	3700	9819	6910	3463	1088	5128
I	0	0	0	0	1	12
Str	0.650	0.997	0.908	0.999	0.420	0.596

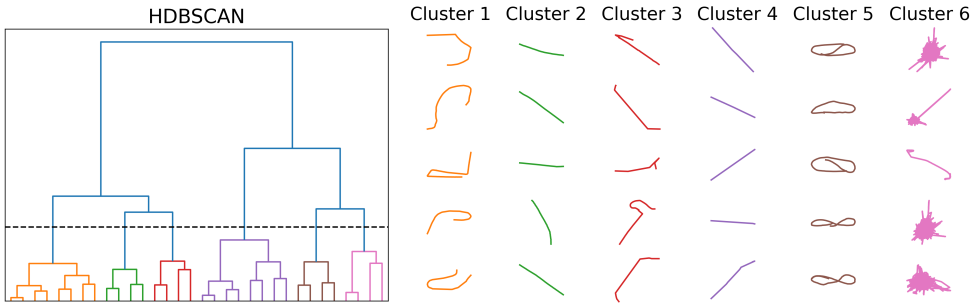


Fig. 7: (left) Dendrogram from hierarchical clustering. (right) Subsample of shapes from each cluster.

while completeness assesses whether all members of a given class are assigned to the same cluster; both measures range from 0 to 1, with higher values indicating better performance. We obtained a V-measure of 0.846, indicating that the OPTICS and HDBSCAN results are largely consistent, with OPTICS effectively refining the HDBSCAN clusters. We selected OPTICS for its data-driven nature, as it requires minimal hyperparameter tuning. This property is particularly advantageous given the variability and complexity of vessel trajectories. The resulting clusters capture distinct movement patterns, providing a structured basis for interactive expert exploration and event labeling.

To organize these shapes for presenting to an expert, we selected three features [14]: the distance between the start and end points of a subtrajectory (SE), straightness (Str), and the number of self-intersections (I). For a given (sub)trajectory X of length m, the features are computed as follows:

$$SE(X) = g(\vec{x}_1, \vec{x}_m) \qquad Str(X) = \frac{g(\vec{x}_{min}, \vec{x}_{max})}{\sum_{i=1}^{m} g(\vec{x}_i, \vec{x}_{i+1})} \qquad I(X) = nbr.\ int.\ lines \quad (4)$$

where $g(\cdot)$ compute the geodesic distance using the *WGS-84* ellipsoid and x_{min} and x_{max} represents the minimum and maximum coordinates of trajectory X. These features capture key geometric and behavioral characteristics relevant to maritime events, such as circling during anchoring or straight-line movement during entry and exit from ports. Hierarchical agglomerative k-means clustering using Ward's linkage groups shapes based on these features, and the dendrogram visualization (Figure 8) offers the user an intuitive way to explore relationships among shape clusters at different levels

of granularity. Figure 7 (left) shows the dendrogram from hierarchical clustering; the black dashed line marks the inter-cluster distance used to cut and obtain 6 clusters. Figure 7 (right) shows a subsample of shapes from each cluster, while Table 5 reports the median value per cluster for the three selected features. Clusters 2, 3, and 4 mainly consist of almost straight, non-intersecting lines and differ in how far the subtrajectory extends. Clusters 2 and 4 include straight trajectories with no return, typically representing transit. The key difference is the distance between the start and end points: cluster 2 has a median SE value of 9819 meters –close to the partitioning threshold–, while cluster 4 has a significantly lower median SE value of 3463 meters. Cluster 3, in contrast, has lower SE and Str values, appearing almost straight (Str) but with more curvature (SE). Clusters 1 and 5 display curvilinear patterns, often linked to trawling. Cluster 1 features large, smooth curves with few or no intersections, suggesting wide-area movements. Cluster 5, in contrast, contains intersecting paths forming tight loops or figure-eight shapes, indicating more localized activity. Cluster 6 includes dense, intersecting trajectories typical of anchored vessels drifting slightly, forming compact, repetitive shapes. Since clusters can reflect overlapping behaviors and no single cluster cleanly maps to a single activity, expert input and iterative, human-guided shape exploration are essential. The proposed user interface supports interactive selection of clusters and individual shapes, enabling domain experts to label movement patterns corresponding to maritime events of interest. This human-in-the-loop approach leverages expert knowledge to build an event-to-shape map, improving detection accuracy for subsequent monitoring. Finally, once shapes are associated with events, the user can select specific vessel trajectories and visualize detected events on an interactive map, providing spatial context to the identified behaviors. Figure 8 illustrates the GUI in action, showing the clustering dendrogram, shape previews, and event mapping.

6 RoSITa for 3-Dimensional Trajectories

In this section, we present a second application of the framework in the context of paragliding flight analysis. Before introducing the use case, and following the approach described in Section 4, we extend RoSITa to handle three-dimensional trajectories. In this setting, a trajectory is defined as a sequence of spatial vectors that include altitude in addition to latitude and longitude, i.e., $\vec{x}_j = (lat_j, long_j, alt_j)$. While the algorithm processes the full three-dimensional representation, we note that the dimensional pairs (lat, alt) and $(long, alt)$ do not carry a direct semantic interpretation for human analysts, as geographic coordinates and altitude belong to different reference dimensions. We then validate the extended algorithm and conduct a sensitivity analysis using a synthetic dataset. Finally, we present the use case on a real-world paragliding dataset comprising 2005 trajectories collected across Italy.

6.1 Extending RoSITa to 3D Trajectories

Extending RoSITa to three-dimensional trajectories can be approached in multiple ways. A first strategy is to apply the original two-dimensional formulation to the pair of dimensions exhibiting the most complex movement, while manually extracting descriptive features from the remaining dimension. In paragliding, most geometric variability

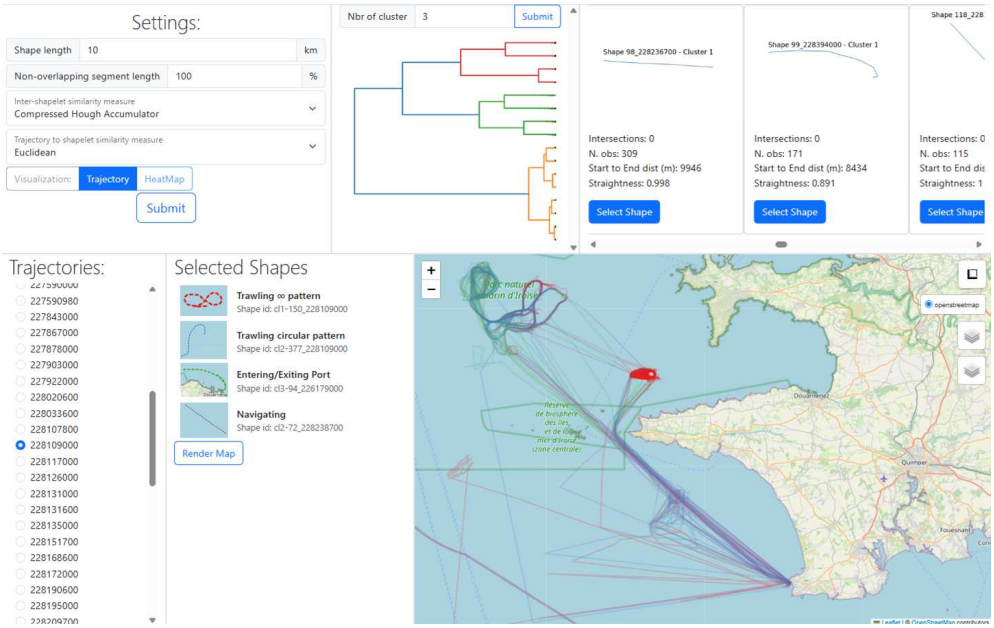


Fig. 8: Vessel Monitoring System

is observed in latitude and longitude, whereas altitude typically descends, with occasional ascents due to thermal lift or wind conditions. Under this assumption, RoSITa can be applied to the top-view projection of the trajectory, while complementary statistics (e.g., mean altitude, variance, rate of descent) are computed separately to capture vertical dynamics. A second approach involves applying the two-dimensional version of RoSITa to different pairs of dimensions, such as latitude–longitude and longitude–altitude, thereby obtaining two distinct vector representations per trajectory. Although this strategy increases flexibility and allows separate modeling of horizontal and vertical structures, ensuring consistent invariance to scale and rotation becomes more complex. In particular, cyclic shifts applied to enforce rotational invariance in one representation may require recomputation or careful alignment of the other, increasing computational overhead. A third approach is to derive a single representation that jointly encodes all three spatial dimensions. By extending the underlying Hough Transform to operate directly in three-dimensional space, this solution captures horizontal and vertical movement patterns simultaneously, preserving geometric coherence across dimensions while maintaining invariance properties. In this section, we focus on the third approach, as it offers a single, self-contained representation. This avoids the need to combine multiple partial representations or manually engineer additional vertical features, resulting in a more streamlined and consistent processing pipeline. Moreover, a unified three-dimensional formulation simplifies both the similarity computation and the subsequent analysis, making it better suited for systematic validation and sensitivity assessment.

To extend the transform to three-dimensional trajectories, we adopt the 3D Hough Transform, which generalizes line detection from the plane to 3D space. In 3D, a line can no longer be uniquely represented by a single pair (ρ, θ); instead, it requires a richer parameterization describing both its orientation and its position in space. A common formulation represents a line through a direction vector, defined by two angles (e.g., azimuth θ and elevation ϕ), and its shortest distance ρ from the origin. Given a point $\vec{x} = (x, y, z)$ in Cartesian space, all possible lines passing through it correspond to a surface in the parameter space (ρ, θ, ϕ). The voting process generalizes the 2D relation by projecting the point onto the normal direction associated with (θ, ϕ), yielding

$$\rho = x \cos\theta \sin\phi + y \sin\theta \sin\phi + z \cos\phi \tag{5}$$

As in the two-dimensional case, points aligned along a common spatial direction yield consistent votes that accumulate in the same region of parameter space. The resulting 3D Hough accumulator is therefore a matrix of size $N_\rho \times N_\theta \times N_\phi$, where N_ρ, N_θ, and N_ϕ denote the number of discretized bins for distance, azimuth, and elevation, respectively.

The RoSITa algorithm (Alg. 1) for three-dimensional trajectories remains largely unchanged. The main modification consists of replacing the 2D Hough accumulator with its 3D counterpart and adapting the aggregation step accordingly. Specifically, after computing the $N_\rho \times N_\theta \times N_\phi$ accumulator, we first aggregate along the ρ dimension and then along either θ or ϕ, obtaining a final vector of size N_ϕ or N_θ, respectively. Depending on the chosen aggregation operator agg, the order of aggregation may influence the resulting representation. For operators such as summation and maximum, the aggregation order does not affect the outcome because they are commutative. In contrast, for operators such as mean, standard deviation, or variance, the order becomes relevant, since the statistics computed along the first aggregated dimension determine the distributional structure passed to the subsequent aggregation step, potentially emphasizing patterns aligned with one plane over the other. We experimentally investigate the impact of this design choice in the following section through a dedicated sensitivity analysis.

6.2 RoSITa: Sensitivity Analysis for 3D trajectories

Similar to the sensitivity analysis conducted on the 2D version of RoSITa, we constructed a synthetic 3D dataset to evaluate the behavior of the extended algorithm under controlled conditions. The dataset consists of six distinct trajectory patterns, each designed to capture a specific type of motion: landing, steady flight, straight movement, planar turning, turning with altitude change, and spiral motion. For each motion category, two representative trajectories are included, providing clear examples of the different movement classes, as illustrated in Figure 9. The trajectories vary systematically in horizontal and vertical dynamics, with parameters controlling aspects such as direction, curvature, pitch, elevation change, and small random deviations to simulate realistic variations. Simple motions, such as takeoff and straight flight, emphasize linear or gently curved paths, whereas more complex motions, such as turning with altitude change or spiraling, introduce coordinated changes across all three

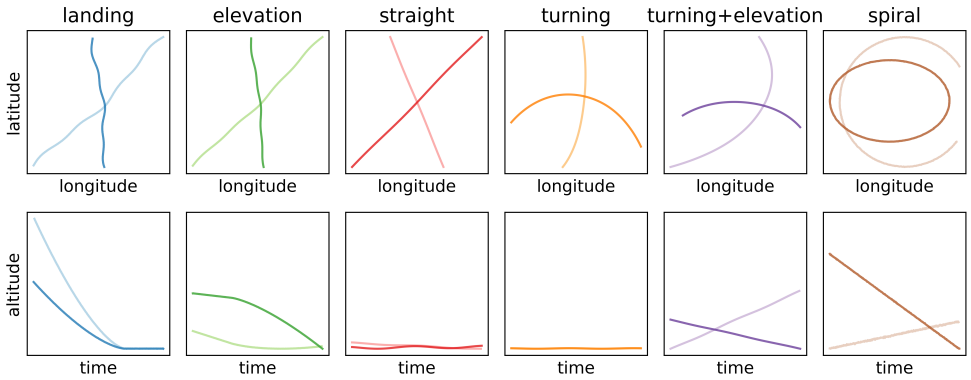


Fig. 9: Synthetic 3-dimensional shapes dataset.

spatial dimensions. As for the `small 2D dataset`, we generated only 20 trajectories per pattern, with an average length of 200±50.

The performance is evaluated using a 1NN downstream classifier using the FFT, OMD, MED, and EMD distances. As non-linear transformations t, we consider the logarithmic, square, and square and cubic root functions. As for the aggregation operator agg, we evaluate several statistical operators: maximum, sum, mean, standard deviation, and variance. Please note that the first two operators are commutative. As for the accumulator resolutions, we tested $N_\rho, N_\phi \in \{36, 90, 180\}$ and $N_\theta \in \{204, 339, 509\}$. The dataset was partitioned into 70-30 train-test set split. Model selection was performed via 3-fold cross-validation on the training set, choosing the configuration that maximizes the macro F1 score.

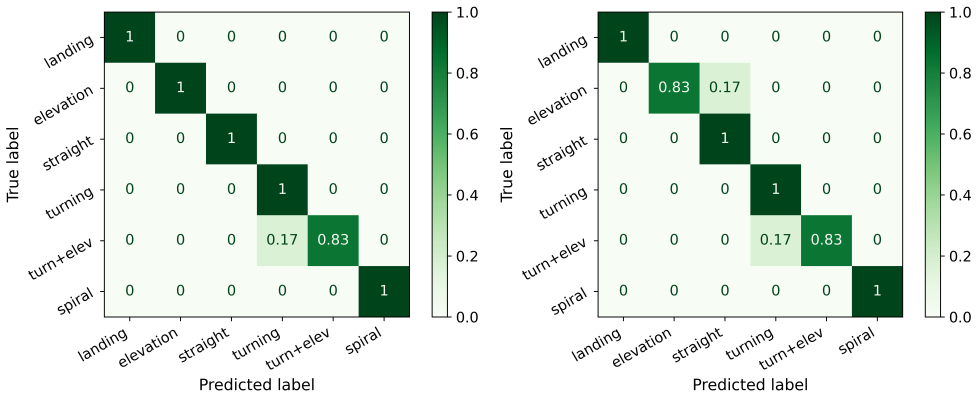
In contrast to the 2D scenario, increasing the Hough accumulator size yields only a slight improvement in performance. This likely reflects the coarser approximation introduced by smaller accumulator sizes, an effect that is amplified by the fact that the aggregation operator is applied twice in the 3D setting, first along ρ and then along either θ or ϕ. For the remaining part of the sensitivity analysis, we fix the parameters at their maximum tested values.

Table 6 reports the macro F1 scores for all combinations of t, agg, and d, with cells shaded on a red-to-green scale (red = 0, green = 1). Consistent with the 2D sensitivity analysis, OMD produces the lowest performance among all distance functions. For the remaining distances, no significant differences are observed in either average or maximum scores. Additionally, applying a square-root transformation to the accumulator values generally leads to lower performance across all aggregation operators, except for max, while the cubic root tends to produce the second-lowest results, particularly when combined with summation or mean aggregation. Notably, regardless of the scaling function or distance measure, using the maximum value for aggregation consistently yields the highest F1 scores. The best overall performance, with a macro F1 score of 0.972, is obtained by configurations that use max aggregation and compute distances with EMD. The second-best results are achieved by the same configurations when distances are computed using FFT.

		log	$\sqrt[3]{}$	$\sqrt{}$	^{2}	max by d
FFT	max	0.944	0.944	0.944	0.917	
	mean	0.333	0.472	0.389	0.722	
	std	0.778	0.806	0.444	0.917	0.944
	sum	0.694	0.556	0.500	0.667	
	var	0.722	0.806	0.417	0.722	
	mean by b		0.685±0.204			
OMD	max	0.639	0.722	0.722	0.556	
	mean	0.167	0.222	0.167	0.389	
	std	0.472	0.528	0.278	0.583	0.722
	sum	0.500	0.222	0.333	0.444	
	var	0.361	0.611	0.361	0.444	
	mean by b		0.436±0.173			
MED	max	0.889	0.861	0.889	0.917	
	mean	0.278	0.528	0.361	0.833	
	std	0.833	0.806	0.472	0.944	0.944
	sum	0.778	0.389	0.444	0.833	
	var	0.722	0.778	0.389	0.806	
	mean by b		0.688±0.220			
EMD	max	0.972	0.972	0.972	0.972	
	mean	0.250	0.444	0.417	0.583	
	std	0.750	0.750	0.528	0.861	0.972
	sum	0.694	0.500	0.500	0.722	
	var	0.778	0.750	0.444	0.694	
	mean by b		0.678±0.213			

Table 6: Sensitivity analysis varying t, agg, and d on the proposed 3D trajectory dataset. Cells show macro F1 scores and are shaded using a red-to-green scale, where red corresponds to a score of 0 and green to a score of 1.

Figures 10a and 10b show the confusion matrices of the best and second-best classifiers, both based on logarithmic scaling of the accumulator. As visible from the matrices, almost all classes are predicted perfectly. In both models, the only noticeable degradation occurs for trajectories that combine turning and elevation change, which are occasionally confused with closely related motion patterns. The second-best classifier shows a specific weakness: about 17% of trajectories involving an elevation change without turning are misclassified as straight trajectories. Overall, these errors suggest that the misclassifications occur between motion patterns that are intrinsically very similar. For instance, when the variation in altitude is small, or when different maneuvers share overlapping geometric characteristics. In such cases, the classes may not be clearly distinguishable even with human inspection. Interestingly, landing trajectories, characterized by an initial change in altitude followed by a short horizontal segment parallel to the ground, are never confused with pure elevation-change trajectories nor with straight motion at constant altitude. This further supports the interpretation that misclassifications arise only when the geometric differences between classes are subtle, whereas well-structured composite patterns such as landings remain clearly separable.



(a) (b)

Fig. 10: Confusion matrices corresponding to the best-performing model (left) and the second-best-performing model (right) on the synthetic 3D dataset.

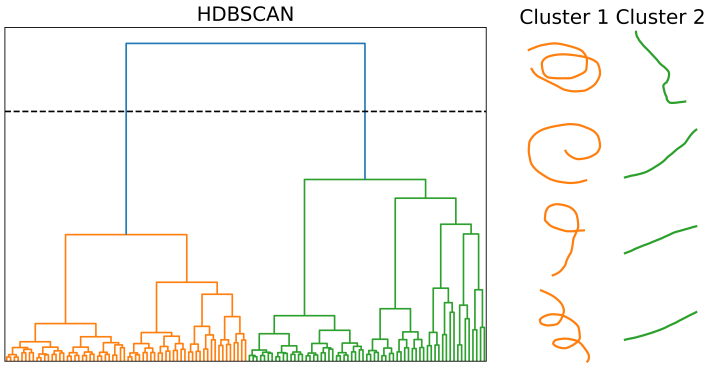


Fig. 11: (left) Dendrogram from hierarchical clustering. (right) Subsample of shapes from each cluster, top-view.

6.3 Case Study for Paragliding Trajectories Analysis

This section presents a real-world application on paragliding trajectories collected in the Italian Alps from the Paragliding Forum[11]. We selected 100 trajectories, yielding approximately 230k recorded points. Following the framework described in Section 5, each trajectory was segmented into non-overlapping subtrajectories of 300 meters, yielding about 7.7k segments. Since some segments had low or irregular sampling rates, we resampled all subtrajectories to a uniform 5-meter spacing, yielding roughly 0.5M points overall. The 3D RoSITa representation was then computed for each subtrajectory. In line with the vessel monitoring use case, we first evaluated density-based clustering through OPTICS and HDBSCAN. However, independently of the chosen

[11] https://paraglidingforum.com/leonardo

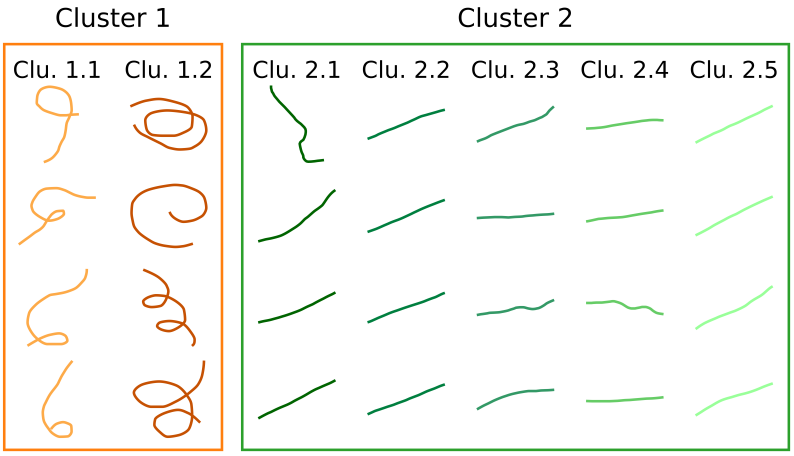


Fig. 12: Subsample of shapes from each final cluster, top-view.

hyperparameters and distance measures, both methods failed to produce meaningful partitions: they either labeled almost all subtrajectories as noise or collapsed the data into a single dominant cluster containing more than 90% of the segments. Given the limitations of density-based approaches in this setting, we turned to distance-based clustering. As the true number of clusters was unknown, we applied Hierarchical Agglomerative Clustering using MED as distance measure. The resulting dendrogram (Figure 11, left) reveals a clear separation into two markedly distinct groups. Visual inspection of representative subtrajectories from each group (Figure 11, right) shows that one cluster predominantly captures nearly straight segments, while the other is mainly composed of curvilinear motion patterns.

Similar to the vessel monitoring use case, clustering can be refined using user-defined features. Instead, we focus on exploring the dendrogram itself to emphasize the capabilities of the RoSITa representation in an exploratory setting, without prior knowledge of task labels. Examining the dendrogram in Figure 11 (left), we analyze the subtrees corresponding to the two main clusters independently. For the first cluster (orange), composed mostly of curvilinear trajectories, the largest gap suggests splitting into two subclusters. For the second cluster (green), the dendrogram is divided further into five subclusters. Figure 12 presents representative subtrajectories from the seven final clusters, illustrating the diversity of paragliding flight patterns captured. Clusters 1.1 and 1.2 primarily differ in the size and number of loops: cluster 1.1 contains shapes with mostly straight segments and small loops, whereas cluster 1.2 includes either larger curvilinear patterns or multiple loops spanning a wider area. The second group of clusters (2.1–2.5) exhibits subtler differences. Cluster 2.1 captures slightly curvilinear trajectories, while clusters 2.2–2.5 are less distinguishable from a top-down perspective. To further characterize these clusters, we compute the proportion of ascending trajectories, which highlights vertical dynamics not immediately apparent in the 2D projection. Table 7 reports the number of ascending and descending trajectories for each cluster, the proportion of ascending flights, and the

Cluster	#Ascending	#Descending	Prop Ascending	Avg Ascend (m)	Avg Descend (m)
2.1	340	1247	0.21	18.7	23.6
2.2	7	418	0.02	48.3	38.0
2.3	12	352	0.03	53.6	40.8
2.4	13	360	0.03	47.0	43.4
2.5	10	376	0.03	32.9	38.3

Table 7: Summary of paragliding flight clusters showing the number of ascending and descending trajectories, the proportion of ascending flights, and the average altitude change in meters for each category.

average altitude change in meters during ascents and descents. These metrics were computed by fitting a linear regression to each flight's altitude profile and classifying flights as ascending or descending based on the slope of the regression line. Cluster 2.1 is dominated by descending trajectories, with only 21% of flights ascending, though the altitude changes are relatively modest (avg. 18.7 meters up, 23.6 meters down). Subclusters 2.2–2.5 show an even stronger prevalence of descending flights, with ascending proportions below 5%. Notably, these subclusters exhibit larger altitude changes during both ascent and descent, reaching up to 53.6m on average during ascents and over 43m during descents. These results highlight a spectrum from frequent, small descents in 2.1 to rare but more pronounced ascents in the other clusters.

We can leverage these clusters to analyze new paragliding trajectories. Since our pipeline relies solely on the RoSITa representation and distance-based clustering, new trajectories can be evaluated using only RoSITa with a similarity match. Each trajectory is segmented into 300-meter portions and converted into a scale- and rotation-invariant RoSITa representation. A 1NN classifier with MED distance assigns each segment to its closest cluster. Figure 13 shows the resulting segments colored according to the map in Figure 12. The trajectory consists of four phases: the first phase (top-left of the figures) features curvilinear and looping patterns; the second phase involves a straight descending flight over the mountainside toward the Tagliamento riverbed. In the third phase, the pilot approaches the landing zone, alternating curvilinear and straight patterns. Finally, in the last phase, only curvilinear maneuvers are performed to reduce speed before landing. This example demonstrates RoSITa's capabilities and high accuracy in analyzing 3D trajectory data.

7 Conclusion

In this work, we introduced RoSITa, a novel rotation- and scale-invariant trajectory representation designed to support the analysis of movement data in unconstrained environments such as maritime and aviation domains. The proposed representation is derived from a transformation of trajectory points through a relative Hough transform, producing compact shape signatures that encode the geometric structure of movement independently of orientation and scale. This design addresses a key limitation of many existing trajectory analysis approaches: the difficulty of comparing movement

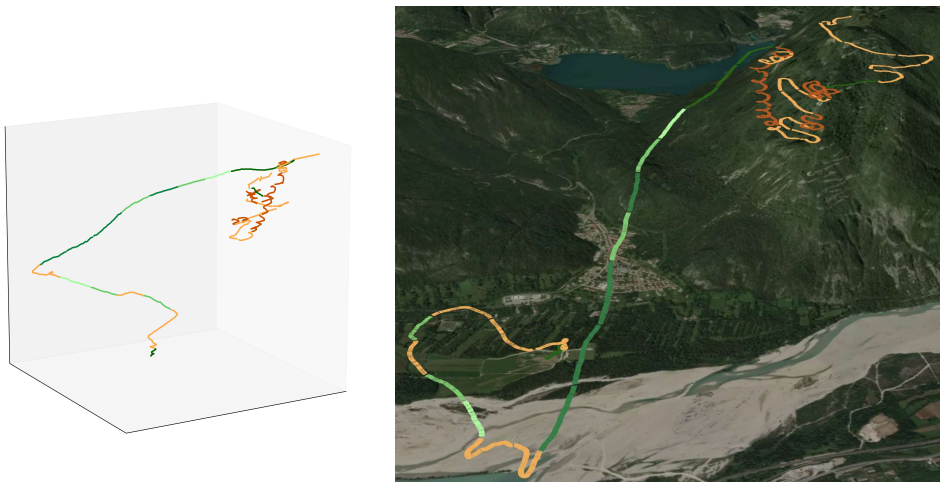


Fig. 13: Visualization of a paragliding trajectory segmented and classified using RoSITa

patterns when trajectories differ in direction, size, or sampling density. By abstracting from these factors, RoSITa enables robust comparisons across trajectories and facilitates the discovery of recurring geometric patterns. Moreover, the representation remains interpretable and computationally efficient, making it suitable both for machine learning pipelines and for visual analytics in human-in-the-loop settings.

We evaluated the effectiveness of RoSITa through experiments on both synthetic and real-world datasets. The sensitivity analysis investigated multiple configurations of the representation, including different nonlinear transformations of the Hough accumulator, aggregation strategies, and distance measures in 2- and 3-dimensional settings. The results showed that the proposed representation produces a discriminative feature space capable of separating complex trajectory shapes, even with limited training data. Furthermore, in classification benchmarks against existing trajectory representations from the mobility and time-series domains, RoSITa demonstrated strong performance, particularly in tasks where movement geometry is a key discriminative factor. These experiments confirm that transforming trajectories into a rotation- and scale-invariant shape space can significantly improve the ability of classical machine learning models to capture geometric patterns.

We also demonstrated the practical utility of the approach through two real-world case studies. The first focused on maritime vessel monitoring, where RoSITa enabled the identification and exploration of characteristic movement patterns associated with vessel behaviors such as trawling or port-related activities. By clustering similar trajectory segments and enabling visual inspection, the framework allowed experts to validate meaningful shapes and associate them with semantic events, facilitating the creation of labeled datasets for downstream learning tasks. The second case study extended the representation to 3-dimensional trajectories in paragliding flight analysis, showing that the approach can capture complex 3D motion dynamics and

reveal characteristic flight maneuvers. Together, these use cases highlight the flexibility of RoSITa across domains and demonstrate how the representation supports both exploratory analysis and knowledge generation from trajectory data.

Several directions for future work emerge from this study. First, the additive nature of the RoSITa representation enables integration into algorithms that segment trajectory data-drivenly, allowing the discovery of meaningful temporal partitions without predefined windows. Second, the framework could be extended to additional application domains involving unconstrained motion, including aircraft trajectory analysis, wildlife monitoring across terrestrial, marine, and aerial environments, and other mobility scenarios where geometry plays a crucial role in understanding behavior. Future research may explore real-time implementations, multi-scale trajectory representations, and the incorporation of contextual attributes (e.g., environmental conditions or sensor data) to further enhance the framework's analytical power. Finally, a further research direction involves integrating RoSITa within a hybrid framework in which the method is first employed to identify trajectory segments with similar geometric shapes, and subsequently complemented by a kinematic-based analysis aimed at extracting motion characteristics within each group of geometrically consistent trajectories.

Declarations

Funding. This work has been partially supported by the Italian Project Fondo Italiano per la Scienza FIS00001966 "MIMOSA", and by the "TANGO" project that has received funding from the European Union's Horizon Europe research and innovation program under G.A. 101120763, and by Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the *Lamarr Institute for Machine Learning and Artificial Intelligence* (Lamarr22B).

Conflict of interest/Competing interests. The authors declare that they have no conflicts of interest.

Ethics approval and consent to participate. Not applicable.

Consent for publication. The authors declare that they all provide consent for publication.

Data availability. The open-source datasets adopted in this work are available at: `animals` and VEHICLES, `GeoLife`, syntetic and SUMO-Simulated datasets, `Brittany vessels`. For `paragliding`, there is no public release due to privacy concerns.

Code availability. The code will be available publicly upon acceptance.

Author contribution. Cristiano Landi: Conceptualization, Methodology, Software, Validation, Investigation, Resources, Writing – Original Draft, Writing – Review & Editing, Visualization, Project administration. Natalia Andrienko: Conceptualization, Methodology, Writing – Review & Editing, Supervision, Project administration. Gennady Andrienko: Conceptualization, Methodology, Writing – Review & Editing, Supervision, Project administration. Riccardo Guidotti: Writing – Review & Editing, Supervision.

References

[1] Lee, J., Han, J., Li, X., Gonzalez, H.: *TraClass*: trajectory classification using hierarchical region-based and trajectory-based clustering. Proc. VLDB Endow. **1**(1), 1081–1094 (2008)

[2] Naini, F.M., Unnikrishnan, J., Thiran, P., Vetterli, M.: Where you are is who you are: User identification by matching statistics. IEEE Transactions on Information Forensics and Security **11**(2), 358–372 (2016) https://doi.org/10.1109/TIFS.2015.2498131

[3] Andrienko, N.V., Andrienko, G.L., Artikis, A., Mantenoglou, P., Rinzivillo, S.: Human-in-the-loop: Visual analytics for building models recognizing behavioral patterns in time series. IEEE Computer Graphics and Applications **44**(3), 14–29 (2024)

[4] Wang, S., Bao, Z., Culpepper, J.S., Cong, G.: A survey on trajectory data management, analytics, and learning. ACM Comput. Surv. **54**(2), 39–13936 (2022)

[5] Han, F., Liu, Y., Tian, H., Li, J., Tian, Y.: A comprehensive framework incorporating deep learning for analyzing fishing vessel activity using automatic identification system data. ICES Journal of Marine Science **82**(2), 166 (2024) https://doi.org/10.1093/icesjms/fsae166

[6] Shahir, A.Y., Charalampous, T., Tayebi, M.A., Glässer, U., Wehn, H.: Triptracker: Unsupervised learning of fishing vessel routine activity patterns. In: IEEE BigData, pp. 1928–1939 (2021)

[7] Duan, H., Ma, F., Miao, L., Zhang, C.: A semi-supervised deep learning approach for vessel trajectory classification based on ais data. Ocean & Coastal Management **218**, 106015 (2022) https://doi.org/10.1016/j.ocecoaman.2021.106015

[8] Petry, L.M., Júnior, A.S., Bogorny, V., Brandoli, B., Matwin, S.: Challenges in vessel behavior and anomaly detection: From classical machine learning to deep learning. In: Canadian AI. Lecture Notes in Computer Science, vol. 12109, pp. 401–407 (2020)

[9] Vlachos, M., Vagena, Z., Yu, P.S., Athitsos, V.: Rotation invariant indexing of shapes and line drawings. In: CIKM, pp. 131–138 (2005)

[10] Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognit. **13**(2), 111–122 (1981)

[11] Landi, C., Andrienko, N., Andrienko, G.: Rotation- and scale-invariant shape extraction from vessel trajectories for human-in-the-loop monitoring. In: SIGSPATIAL/GIS (2025)

[12] Zheng, Y.: Trajectory data mining: An overview. ACM Trans. Intell. Syst. Technol. **6**(3), 29–12941 (2015)

[13] Middlehurst, M., Schäfer, P., Bagnall, A.J.: Bake off redux: a review and experimental evaluation of recent time series classification algorithms. Data Min. Knowl. Discov. **38**(4), 1958–2031 (2024)

[14] Landi, C., Guidotti, R., Nanni, M., Monreale, A.: The trajectory interval forest classifier for trajectory classification. In: SIGSPATIAL/GIS, pp. 67–1674 (2023)

[15] Bolbol, A., Cheng, T., Tsapakis, I., Haworth, J.: Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. Comput. Environ. Urban Syst. **36**(6), 526–537 (2012)

[16] Zheng, Y., *et al.*: Understanding mobility based on GPS data. In: UbiComp. ACM International Conference Proceeding Series, vol. 344, pp. 312–321 (2008)

[17] Zheng, Y., *et al.*: Learning transportation mode from raw gps data for geographic applications on the web. In: WWW, pp. 247–256 (2008)

[18] Dodge, S., Weibel, R., Forootan, E.: Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. Comput. Environ. Urban Syst. **33**(6), 419–434 (2009)

[19] Etemad, M., Júnior, A.S., Matwin, S.: Predicting transportation modes of GPS trajectories using feature engineering and noise removal. In: Canadian Conference on AI. Lecture Notes in Computer Science, vol. 10832, pp. 259–264 (2018)

[20] Laube, P., Purves, R.S.: An approach to evaluating motion pattern detection techniques in spatio-temporal data. Comput. Environ. Urban Syst. **30**(3), 347–374 (2006)

[21] Reddy, S., *et al.*: Determining transportation mode on mobile phones. In: ISWC, pp. 25–28 (2008)

[22] Xiao, Z., Wang, Y., Fu, K., Wu, F.: Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. IJGI **6**(2), 57 (2017)

[23] Soleymani, A., Cachat, J., Robinson, K., Dodge, S., Kalueff, A., Weibel, R.: Integrating cross-scale analysis in the spatial and temporal domains for classification of behavioral movement. J. Spatial Inf. Sci. **8**(1), 1–25 (2014)

[24] Silva, C.L., Petry, L.M., Bogorny, V.: A survey and comparison of trajectory classification methods. In: BRACIS, pp. 788–793 (2019)

[25] Grabocka, J., Schilling, N., Wistuba, M., Schmidt-Thieme, L.: Learning time-series shapelets. In: ACM KDD, pp. 392–401 (2014)

[26] Ferrero, C.A., Petry, L.M., Alvares, L.O., Silva, C.L., Zalewski, W., Bogorny, V.: Mastermovelets: discovering heterogeneous movelets for multiple aspect trajectory classification. Data Min. Knowl. Discov. **34**(3), 652–680 (2020)

[27] Ferrero, C.A., Alvares, L.O., Zalewski, W., Bogorny, V.: MOVELETS: exploring relevant subtrajectories for robust trajectory classification. In: SAC, pp. 849–856 (2018)

[28] Portela, T.T., Carvalho, J.T., Bogorny, V.: Hipermovelets: high-performance movelet extraction for trajectory classification. Int. J. Geogr. Inf. Sci. **36**(5), 1012–1036 (2022)

[29] Landi, C., Spinnato, F., Guidotti, R., Monreale, A., Nanni, M.: Geolet: An interpretable model for trajectory classification. In: IDA. Lecture Notes in Computer Science, vol. 13876, pp. 236–248 (2023)

[30] Landi, C., Guidotti, R.: Shape-based methods in mobility data analysis: effectiveness and limitations. GeoInformatica (2025) https://doi.org/10.1007/s10707-025-00555-x

[31] Huang, J., Wang, H., Sun, Y., Shi, Y., Huang, Z., Zhuo, A., Feng, S.: Ernie-geol: A geography-and-language pre-trained model and its applications in baidu maps. In: KDD, pp. 3029–3039 (2022)

[32] Suwardi, I.S., Dharma, D., Satya, D.P., Lestari, D.P.: Geohash index based spatial data model for corporate. In: ICEEI 2015, pp. 478–483 (2015). IEEE

[33] Ye, L., Keogh, E.J.: Time series shapelets: a new primitive for data mining. In: KDD, pp. 947–956 (2009)

[34] Landi, C., Guidotti, R.: Shape-based methods in mobility data analysis: effectiveness and limitations. GeoInformatica **29**(4), 999–1032 (2025)

[35] Musleh, M., Mokbel, M.F.: Let's speak trajectories: A vision to use NLP models for trajectory analysis tasks. ACM Trans. Spatial Algorithms Syst. (2024)

[36] Musleh, M., Mokbel, M.F.: KAMEL: A scalable bert-based system for trajectory imputation. Proc. VLDB Endow. **17**(3), 525–538 (2023)

[37] Yao, D., Zhang, C., Zhu, Z., Huang, J., Bi, J.: Trajectory clustering via deep representation learning. In: IEEE IJCNN, pp. 3880–3887 (2017)

[38] Kontopoulos, I., Makris, A., Tserpes, K.: Traclets: A trajectory representation and classification library. SoftwareX **21**, 101306 (2023)

[39] Gaol, F.L.: Bresenham algorithm: Implementation and analysis in raster shape. J. Comput. **8**(1), 69–78 (2013)

[40] Gamage, C., Dinalankara, R., Samarabandu, J., Subasinghe, A.: A comprehensive survey on the applications of machine learning techniques on maritime surveillance to detect abnormal maritime vessel behaviors. WMU Journal of Maritime Affairs **22**(4), 447–477 (2023)

[41] Ford, J.H., Peel, D., Hardesty, B.D., Rosebrock, U., Wilcox, C.: Loitering with intent—catching the outlier vessels at sea. Plos one **13**(7), 0200189 (2018)

[42] Sfyridis, A., Cheng, T., Vespe, M.: Detecting vessels carrying migrants using machine learning. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences **4**, 53–60 (2017)

[43] Zhang, X., Fu, X., Xiao, Z., Xu, H., Qin, Z.: Vessel trajectory prediction in maritime transportation: Current approaches and beyond. IEEE Transactions on Intelligent Transportation Systems **23**(11), 19980–19998 (2022)

[44] Zhang, Y., Zheng, Z., Xiang, C.: Analysis of vessel abnormal behavior detection. In: Proceedings of the 2020 2nd International Conference on Robotics Systems and Vehicle Technology, pp. 45–53 (2020)

[45] Pitsikalis, M., Artikis, A., Dreo, R., Ray, C., Camossi, E., Jousselme, A.: Composite event recognition for maritime monitoring. In: DEBS, pp. 163–174 (2019)

[46] Andrienko, N.V., Andrienko, G.L.: Visual analytics of vessel movement. In: Guide to Maritime Informatics, pp. 149–170 (2021)

[47] Krajzewicz, D.: Traffic simulation with sumo–simulation of urban mobility. Fundamentals of traffic simulation, 269–293 (2010)

[48] Middlehurst, M., Large, J., Bagnall, A.J.: The canonical interval forest (CIF) classifier for time series classification. In: IEEE BigData, pp. 188–195 (2020)

[49] Lubba, C.H., *et al.*: catch22: Canonical time-series characteristics - selected through comparative time-series analysis. DAMI **33**(6), 1821 (2019)

[50] Spinnato, F., Guidotti, R., Monreale, A., Nanni, M.: Fast, interpretable, and deterministic time series classification with a bag-of-receptive-fields. IEEE Access **12**, 137893–137912 (2024)

[51] Bagnall, A.J., Bostrom, A., Large, J., Lines, J.: The great time series classification bake off: An experimental evaluation of recently proposed algorithms. extended version. CoRR **abs/1602.01711** (2016)

[52] Faouzi, J.: Time series classification: A review of algorithms and implementations. Machine Learning (Emerging Trends and Applications) (2022)

[53] Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach.

Learn. Res. **7**, 1–30 (2006)

[54] Kontopoulos, I., Chatzikokolakis, K., Tserpes, K., Zissis, D.: Classification of vessel activity in streaming data. In: DEBS, pp. 153–164 (2020)

[55] Yang, Y., Liu, Y., Li, G., Zhang, Z., Liu, Y.: Harnessing the power of machine learning for ais data-driven maritime research: A comprehensive review. Transportation research part E: logistics and transportation review **183**, 103426 (2024)

[56] Liu, H., Chen, X., Wang, Y., Zhang, B., Chen, Y., Zhao, Y., Zhou, F.: Visualization and visual analysis of vessel trajectory data: A survey. Vis. Informatics **5**(4), 1–10 (2021)

[57] Ahmed, S.A., *et al.*: Trajectory-based surveillance analysis: A survey. IEEE Trans. Circuits Syst. Video Technol. **29**(7), 1985–1997 (2019)

[58] Boots, B., Tiefelsdorf, M.: Global and local spatial autocorrelation in bounded regular tessellations. J. Geogr. Syst. **2**(4), 319–348 (2000)

[59] RAY, C., DRÉO, R., CAMOSSI, E., JOUSSELME, A.-L.: Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance, and Reconnaissance (2018). https://doi.org/10.5281/zenodo.1167595

[60] Ankerst, M., Breunig, M.M., Kriegel, H., Sander, J.: OPTICS: ordering points to identify the clustering structure. In: SIGMOD Conference, pp. 49–60 (1999)

[61] Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)

[62] McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. J. Open Source Softw. **2**(11), 205 (2017)

[63] Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: EMNLP-CoNLL, pp. 410–420 (2007)