Transportation Research Part E 209 (2026) 104715

Contents lists available at ScienceDirect

Transportation Research Part E

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

PortMiner: Unsupervised data mining for functional areas extraction in port areas

Huimin Qiang [a,b], Wenlong Niu [a], Xiaodong Peng [a,*], Huanhuan Li [c,*], Zaili Yang [d]

[a] The Key Laboratory of Electronics and Information Technology for Space System, National Space Science Center, Chinese Academy of Sciences, 100190, Beijing, China
[b] University of Chinese Academy of Sciences, 100049, Beijing, China
[c] School of Engineering, University of Southampton, SO16 7QF, Southampton, UK
[d] Liverpool Logistics, Offshore and Marine (LOOM) Research Institute, Liverpool John Moores University, L3 3AF, Liverpool, UK

ARTICLE INFO

Keywords:
Functional area extraction
Recognition of vessel behaviors
Data mining framework
AIS data
Intelligent port operation

ABSTRACT

Accurate extraction of functional areas in port waters is essential for enhancing port operational oversight, optimizing vessel scheduling, and supporting maritime safety. However, existing approaches often rely on supervised learning, extensive parameter tuning, and labeled datasets, limiting their scalability, adaptability, and operational efficiency. To address these gaps, this study proposes PortMiner, a novel unsupervised data mining framework that systematically extracts functional areas from raw vessel trajectory data without requiring manual annotations. The framework introduces a Spatio-Temporal Adaptive Sliding Windows (STASW) method to detect stop behaviors dynamically, using self-adaptive parameters derived directly from the data. Trajectories are first encoded into geohash-based sequential grids, enabling efficient detection of stop and port inbound/outbound behaviors. Functional zones such as berths, anchorages, and navigational channels are then delineated through multi-level spatial aggregation and connectivity-based clustering. Experimental results on benchmark datasets show that STASW achieves 98.83% accuracy, outperforming state-of-the-art deep learning methods, while significantly reducing computational time and cost. Validation against official nautical charts confirms PortMiner's high fidelity in identifying port-functional structures. The extracted results are also made publicly accessible via an interactive platform (https://portminer.netlify.app/), offering practical insights for intelligent port operation and maritime logistics planning.

1. Introduction

Maritime transportation plays a pivotal role in international trade, facilitating the global movement of goods and commodities with high efficiency and reliability (UNCTD, 2024; Liu et al., 2025a). In recent years, the sector has witnessed significant structural and operational shifts, driven by the increasing demand for larger vessels, faster turnaround times, and enhanced service reliability (Teixeira and Guedes Soares, 2018; Liu et al., 2023b). These developments have led to a more dynamic and congested maritime environment, particularly in port waters, thereby amplifying the complexity of port management and heightening concerns around maritime safety and operational resilience (Forti et al., 2019; Rong et al., 2022).

* Corresponding authors.
E-mail addresses: Pxd@nssc.ac.cn (X. Peng), huanhuan.li@soton.ac.uk (H. Li).

https://doi.org/10.1016/j.tre.2026.104715
Received 9 August 2025; Received in revised form 21 January 2026; Accepted 22 January 2026
Available online 2 February 2026

To address these challenges, ports are undergoing a digital transformation, increasingly adopting intelligent systems to support safer, greener, and more efficient operations (Song et al., 2024). The shift toward smart ports, exemplified by initiatives at Tianjin Port in China (Liu et al., 2025b), reflects a broader trend toward integrating automation, data analytics, and real-time monitoring to enhance operational efficiency, mitigate congestion, and facilitate intelligent decision-making for sustainable port logistics (Li and Yang, 2023; Niu et al., 2024; Khalifeh and Caliskan, 2025; Chu et al., 2025). A key enabler of this transformation is the capability to accurately and automatically identify functional areas within port waters, such as anchorages, berths, and navigational channels, which are essential for vessel scheduling, traffic supervision, and operational safety (Pallotta et al., 2013; Yang et al., 2019; Xiao et al., 2024).

The delineation of port functional areas offers multifaceted benefits. It enables the detection of vessel boundary violations and illegal anchoring, supports real-time enforcement, and enhances traffic control in high-density zones (Wei et al., 2024; Rong et al., 2024). Furthermore, it provides structured spatial input for downstream applications such as berth planning, vessel arrival prediction, vessel trajectory prediction, and turnaround time optimization (Wijaya and Nakamura, 2024a; Li et al., 2024, 2025; Qiang et al., 2025a; Gong et al., 2025; Jiao et al., 2026). Despite its strategic importance, however, the current state of functional area extraction remains constrained by methodological limitations in robustness, scalability, and real-time applicability (Ma et al., 2024).

Stop behaviors such as anchoring outside the port or berthing within port limits constitute the basis for identifying functional zones. Traditionally, the identification of such behaviors relies on features from Automatic Identification System (AIS) data (Zrnić et al., 1999). However, due to inaccuracies and temporal delays in navigational status fields, which are often manually reported, reliance on AIS metadata poses significant challenges (Mujal-Colilles et al., 2022). Existing recognition techniques can be broadly classified into: (1) threshold-based rule approaches (Qiang et al., 2025b); (2) unsupervised clustering of trajectory points (Li et al., 2022; Bai et al., 2023), and (3) supervised learning methods, including graph neural networks (Xin et al., 2024). While deep learning methods achieve high accuracy, they require large labeled datasets and substantial computational resources. Threshold-based approaches are often sensitive to parameter selection and lack generalization across diverse port environments. Although data-driven, clustering-based methods often misclassify complex manoeuvres in congested waters (e.g., turning or drifting) as stop events, thereby limiting their reliability in operational use.

Moreover, many existing methods end after stop point recognition and do not advance to the spatial delineation of functional areas. This gap limits their integration into port intelligence systems that require structured and interpretable spatial units for analysis. Some recent works have attempted to define stop areas by constructing polygons around annotated stopping points (Yan et al., 2022). However, such representations entail costly geometric computations for spatial queries and are often ill-suited for real-time deployment.

Similarly, inbound and outbound channel extraction remains a challenging task (Huang et al., 2024). Current methods ranging from vector-based clustering (Santos et al., 2012), grid-based visualizations (Wu et al., 2017), to statistical models like Kernel Density Estimation (KDE) (Filipiak et al., 2020, 2021) suffer from limitations such as sensitivity to grid partitioning, parameter tuning requirements, and low adaptability to varying port traffic conditions. In dense port areas, grid-based approaches often misidentify maneuvering zones as primary navigational channels due to high local vessel density, leading to inaccurate representations.

To address these limitations, a holistic and data-driven unsupervised framework, PortMiner, is proposed for extracting port functional areas directly from raw AIS data. The PortMiner distinguishes itself from existing approaches through two major innovations. First, it proposes a Spatio-Temporal Adaptive Sliding Windows (STASW) algorithm for stop behavior recognition, which eliminates the need for manual parameter tuning and labeled data by adaptively determining behavior patterns from intrinsic trajectory features. Second, PortMiner employs multi-resolution geohash encoding for trajectory representation and area delineation, enabling efficient spatial aggregation, scalable multi-level indexing, and real-time functional area retrieval without relying on computationally expensive polygon operations. The main contributions of this work are summarized as follows:

(1) PortMiner, a novel and end-to-end data-driven unsupervised framework, is developed for vessel behavior recognition and functional area extraction, covering the entire pipeline from raw AIS data preprocessing to spatial delineation.

(2) A data-driven unsupervised STASW method is proposed for stop behavior recognition, which adaptively configures parameters based on spatio-temporal trajectory characteristics without requiring any annotated data.

(3) A geohash-based representation scheme is designed for functional area extraction, supporting multi-level spatial indexing and efficient connectivity analysis via a depth-first search algorithm.

(4) Extensive experiments are conducted on both benchmark datasets and real-world AIS data from the Port of Los Angeles. Comparative evaluations with official nautical charts validate the accuracy and robustness of the proposed framework. The results are also made publicly accessible via an interactive web platform (https://portminer.netlify.app/), promoting transparency and practical uptake.

The remainder of this paper is organized as follows. Section 2 reviews existing methods for vessel behavior recognition and functional area extraction. Section 3 presents the PortMiner framework, including data preprocessing, behavior recognition, and functional area extraction. Section 4 describes the experimental setup and results. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Related work

This section reviews prior research on vessel behavior recognition and functional area extraction, two critical domains closely related to the development of the proposed PortMiner framework. A thorough understanding of existing methods highlights both their advancements and the limitations that motivate this study.

2.1. Vessel behavior recognition

Although AIS data provide detailed positional and navigational attributes of vessels, the reliability of manually input fields, particularly navigational status, is often poor. This has prompted the development of automated behavior recognition techniques, which can be broadly categorized into rule-based analysis, unsupervised clustering, and (semi-) supervised deep learning methods. Statistical and rule-based methods typically rely on predefined thresholds for key features such as vessel speed or course variation. For example, Yan et al. (2022) identified stop behaviors in real time by setting a fixed speed threshold, though with limited accuracy. Wijaya and Nakamura (2024b) improved stop detection performance by smoothing trajectories and applying thresholds to a combination of spatio-temporal and geometric features. Qiang et al. (2025b) introduced a vector-based approach, calculating the cumulative dot product of adjacent trajectory vectors to infer vessel stops.

Unsupervised clustering methods have also gained traction due to their data-driven nature. These approaches often involve preprocessing steps such as trajectory segmentation, dimensionality reduction, or similarity computation before clustering (Li et al., 2022; Bai et al., 2023). For instance, Gao and Shi (2020) utilized T-distributed Stochastic Neighbor Embedding (T-SNE) to reduce data dimensionality, followed by spectral clustering for sub-trajectory segments. Rong et al. (2024) adopted Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to cluster behavior patterns and subsequently employed a random forest classifier to label behavioral patterns. Suo et al. (2025) combined the Smith-Waterman algorithm for behavior pattern recognition with text similarity measures for behavior pattern recognition. Despite their potential, clustering methods are often computationally intensive and scale poorly with large datasets.

Deep learning approaches have recently emerged to address the limitations of traditional methods. Zhang et al. (2022) transformed trajectories into grid representations and utilized Convolutional Neural Networks (CNN) to classify various behavioral types. Ma et al. (2025) proposed a hybrid method combining eXtreme Gradient Boosting (XGBoost) with the Sparrow Search Algorithm (SSA) for behavior classification. These approaches improve recognition accuracy but depend heavily on large volumes of labeled data and extensive training, reducing their generalizability across different ports and vessel types.

In light of the above studies, although substantial progress has been made, most suffer from either rigid assumptions, data inefficiency, or limited adaptability, particularly in recognizing complex stop behaviors in dynamic port environments.

2.2. Vessel functional area extraction

Functional area extraction focuses on identifying spatial regions associated with distinct vessel activities, particularly stop areas. These efforts are typically divided into two methodological streams: feature area extraction and channel extraction. Feature area extraction seeks to detect and spatially define zones where vessels exhibit characteristic behaviors. Zhang et al. (2019) developed an autocorrelation-based statistical approach to quantitatively capture the spatio-temporal patterns of maritime traffic, effectively detecting regions with high- and low-speeds in the Singapore Strait. However, such methods often rely on visual inspection of heatmaps and are sensitive to subjective design choices. To address this, polygonal representations have been proposed to encapsulate detected stop points. Yan et al. (2022) and Guo et al. (2024) constructed geometric polygons around stop clusters to represent anchorage and berth areas. Wijaya and Nakamura (2024b) refined this by introducing subdivisions based on vessel type and tonnage. Nonetheless, polygon-based methods are limited by their geometric rigidity: simple polygons may inadequately capture complex spatial boundaries, while high-resolution polygons significantly increase computational costs for spatial queries, especially in real-time applications.

Channel extraction methods can be broadly categorized into three types: cluster-based methods, grid-based methods, and statistics-based methods (Huang et al., 2024). Cluster-based methods use vector-based techniques to identify routes by grouping similar trajectories based on motion patterns. Li et al. (2022) proposed an unsupervised hierarchical methodology called Improved Spectral Clustering with Mapping (ISCM) to extract maritime traffic routes, while Liu et al. (2025a) developed a customized K-Nearest Neighbors block adaptive DBSCAN (CKBA-DBSCAN) for route extraction. Grid-based methods discretize space into uniform cells and aggregate traffic statistics within each grid. Liu et al. (2023a) computed grid-based traffic intensity and used a cumulative importance score to differentiate core traffic zones. These methods, however, are often sensitive to grid resolution and misidentify dense maneuvering zones as main channels. Finally, statistical models such as KDE (Lu et al., 2020) and Gaussian process regression (Rong et al., 2022) aim to derive channel boundaries and central flow lines, but often struggle with computational efficiency and adaptability in heterogeneous port environments.

Overall, existing functional area extraction approaches exhibit limitations in adaptability, scalability, and representation accuracy, particularly in complex and high-density maritime zones such as port approaches.

2.3. Research gaps

Despite meaningful advancements in vessel behavior recognition and functional area extraction, three major limitations remain unresolved:

(1) Most existing approaches address these problems in isolation, lacking an integrated, end-to-end framework that spans from raw AIS data preprocessing to behavior recognition and spatial functional area delineation. This disjointed pipeline limits their operational efficiency and hinders real-world deployment.

(2) Accurate stop behavior recognition remains a persistent challenge due to the diverse characteristics of vessels, varying operational contexts, and environmental noise. Existing methods rely heavily on fixed thresholds or supervised learning, restricting their generalizability across ports without extensive manual tuning or labeled data.

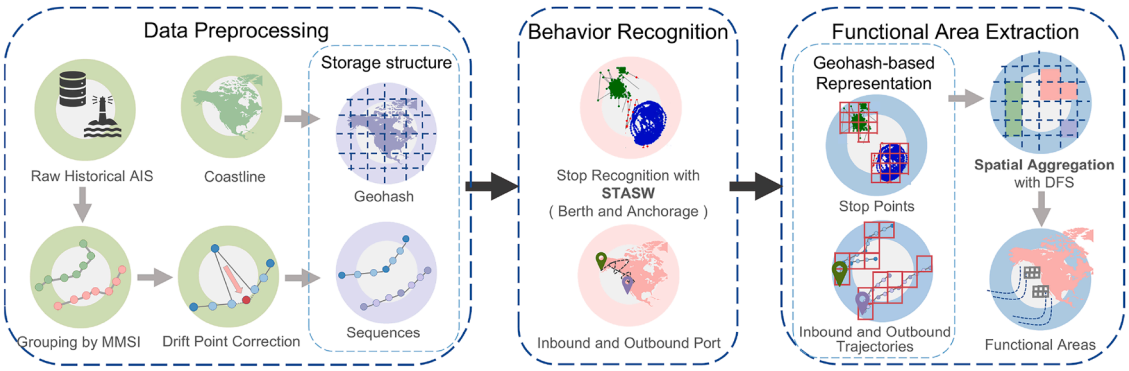


Fig. 1. Overview of the PortMiner framework for unsupervised extraction of port functional areas.

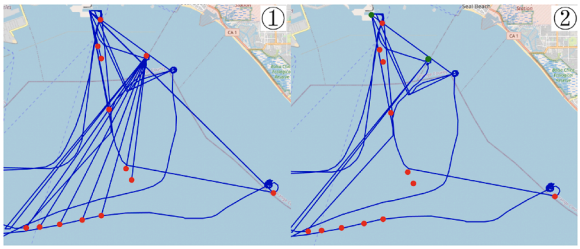


Fig. 2. Comparison of vessel trajectories before and after drift correction. Red points indicate drifted trajectory points, while green points represent the corrected positions. It can be observed that drift events predominantly occur when vessels stop. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(3) Current methods for functional area extraction typically rely on simplistic geometric representations or clustering techniques, which are either computationally expensive or lack the flexibility to represent dynamic and irregular port structures. These constraints reduce their effectiveness for scalable spatial retrieval or real-time monitoring.

3. Methodology

The section presents PortMiner, a unified and data-driven unsupervised framework for mining functional areas in port waters based on the raw AIS data. An overview of the framework is shown in Fig. 1. The framework comprises three key modules: data preprocessing, behavior recognition, and functional area extraction. It integrates coastline and AIS trajectory data, which are structured into geohash grids and trajectory sequences to enable efficient spatial analysis. Leveraging this structured data, PortMiner automatically identifies behaviors such as berth and anchorage, extracts trajectory points associated with port entry and exit events, and segments inbound and outbound trajectories. Finally, using the spatial distribution of these behaviors within geohash-encoded space, PortMiner detects multi-level functional areas and navigational channels.

3.1. PortMiner framework

To provide a clear overview of the PortMiner framework and its data organization, a detailed illustration of the methodology and data processing flow is presented. As shown in Fig. 1, the framework comprises a unified data architecture and two primary analytical modules:

- **Geohash-based data structure:** AIS data and coastline information are organized using both raw coordinates and hierarchical geohash grids. This enables efficient spatial encoding, indexing, and retrieval, which supports high-performance behavior recognition and functional area delineation.
- **Vessel behavior recognition:** Two behavioral categories are recognized, stop events (e.g., berthing, anchoring) and inbound/outbound port movements. For stop behavior detection, we introduce a novel STASW method that adaptively determines parameters without manual tuning or labeled data. Inbound and outbound events are inferred based on temporal gaps between trajectory points near port boundaries.
- **Functional area extraction:** Recognized behaviors are aggregated within geohash grids and processed using a connectivity-based clustering algorithm to extract spatially coherent functional zones, including berth areas, anchorages, and navigational channels.

As illustrated in Fig. 1, the preprocessing steps include vessel classification by Maritime Mobile Service Identity (MMSI), drift correction (see Fig. 2), and geohash encoding. These structured datasets support comprehensive performance benchmarking, real-

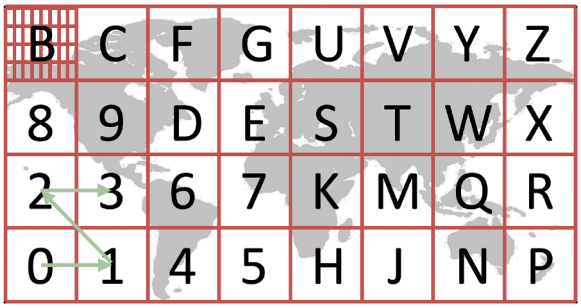


Fig. 3. Illustration of the geohash spatial encoding scheme. The Earth's surface is recursively divided into rectangular cells, each assigned a unique alphanumeric code. Each cell can be further subdivided, with longer codes corresponding to finer spatial resolution.

Table 1
Description of variables.

Variable	Description
t	Timestamp (s)
λ	Longitude
ϕ	Latitude
p	(λ,ϕ)
v	Speed over ground (m/s)
θ	Course over ground (Degree)
v	(v,θ)
δ	Distance between consecutive trajectory points (m)

world validation, and detailed visualization of vessel activity regions. A detailed summary of all parameters, including their statistical derivation procedures and supporting rationale, is provided in Appendix A, Table A.1.

Geohash encoding, shown in Fig. 3, is a spatial indexing method that encodes latitude and longitude into short alphanumeric strings, enabling efficient storage and querying of location data (Ma et al., 2021). It works by recursively dividing the Earth's surface into a grid of rectangular cells, where each cell is assigned a unique code. Longer codes represent smaller, more precise areas. This hierarchical structure supports fast spatial queries, scalable aggregation, and efficient spatial joins. In PortMiner, geohash grids are used to organize vessel trajectory data, enabling effective functional area extraction and real-time spatial matching between AIS trajectories and coastline features.

3.2. Behavior identification

3.2.1. Problem formulation

The task of vessel behavior recognition is formulated as a multi-class classification and prediction problem. The objective is to derive a mapping function $\mathcal{P}$ that infers the vessel's current behavioral category by leveraging both its static characteristics and extracted trajectory-based features. The formulation is defined as:

$$\mathcal{P}(\mathbf{V},\mathbf{T};\mathbb{C}) \rightarrow y \in \{-2,-1,0,1,2\} \tag{1}$$

Here, $\mathcal{P}$ denotes the vessel behavior recognition model (e.g., STASW). $\mathbf{V} = \{\text{type}, \text{length}, \text{width}\}$ is static attributes, $\mathbf{T} = \left\{\mathbf{x}_i^t, \ldots, \mathbf{x}_i^{t+L_x} \mid \mathbf{x}_i^t \in \mathbb{R}^5\right\}$ represents a sequence of trajectory points with $\mathbf{x} = (t, \lambda, \phi, v, \theta)$ (see Table 1), including timestamp, coordinates, speed, and course. i indexes vessels, and L_x is the sequence length. Additionally, the distance δ between consecutive points is computed to enrich the motion features. All parameters $\mathbb{C}$ are fixed constants or adaptively derived from vessel characteristics. y denotes behavior labels: entry (-2), exit (-1), underway (0), berthing (1), and anchoring (2).

3.2.2. Vessel inbound and outbound port recognition

The inbound and outbound recognition method (Algorithm 1) analyzes time intervals between trajectory points located at level 5 geohash-defined boundary zones (Appendix Fig. D.1). In addition to the trajectory sequence, the algorithm also takes as input the indices of trajectory points within the boundary grids, enabling efficient identification of inbound and outbound events. A time threshold of 30 min. is applied to detect significant temporal gaps that indicate vessel movement across the port boundary. This threshold is supported by AIS reporting patterns and further corroborated by the MISM (2020) report, which notes that signal interruptions longer than 30 mins are extremely rare. The threshold selection is further motivated by two considerations: (1) Operationally, vessels navigating near port areas often generate frequent and minor boundary crossings, especially smaller and more maneuverable vessels. To reduce noise and focus on meaningful transitions through main port channels, the method targets larger vessels (length $>$ 50 m), which are less likely to exhibit abrupt directional changes and almost never experience AIS gaps exceeding 30 mins; and (2) Empirically, a statistical analysis of AIS data across vessel types, sizes, and shoreline proximity revealed that for cargo vessels

Algorithm 1 Inbound and outbound recognition.

Input: $\mathbf{x}$, the trajectory points sorted by timestamp; INDEXS, the indices of points at the region boundary; ϵ_t, the time threshold (set to 1800 s).

Output: $\mathbf{x}$ with inbound/outbound state marks for each point.

1: **for** each point $\mathbf{x}_i$ in $\mathbf{x}$ **do**
2: $\quad \mathbf{x}_i.is\_target \leftarrow (\mathbf{x}_i.index \in \text{INDEXS})$
3: **end for**
4: **for** each point $\mathbf{x}_i$ in $\mathbf{x}$ **do**
5: $\quad$ **if** $\mathbf{x}_i.is\_target$ **then**
6: $\quad\quad \mathbf{x}_i.prev\_t \leftarrow \mathbf{x}_{i-1}.t$ (if $i > 1$ else None)
7: $\quad\quad \mathbf{x}_i.next\_t \leftarrow \mathbf{x}_{i+1}.t$ (if $i <$len($\mathbf{x}$) else None)
8: $\quad\quad \mathbf{x}_i.prev\_diff \leftarrow \mathbf{x}_i.t - \mathbf{x}_i.prev\_t$
9: $\quad\quad \mathbf{x}_i.next\_diff \leftarrow \mathbf{x}_i.next\_t - \mathbf{x}_i.t$
10: $\quad\quad$ **if** $(\mathbf{x}_i.prev\_diff > \epsilon_t)$ **or** $(\mathbf{x}_i.prev\_diff$ is None and $\mathbf{x}_i.next\_diff \leq \epsilon_t)$ **then**
11: $\quad\quad\quad \mathbf{x}_i.state \leftarrow -2$ ▷ Inbound event
12: $\quad\quad$ **else if** $(\mathbf{x}_i.next\_diff > \epsilon_t)$ **or** $(\mathbf{x}_i.next\_diff$ is None and $\mathbf{x}_i.prev\_diff \leq \epsilon_t)$ **then**
13: $\quad\quad\quad \mathbf{x}_i.state \leftarrow -1$ ▷ Outbound event
14: $\quad\quad$ **else**
15: $\quad\quad\quad \mathbf{x}_i.state \leftarrow 0$ ▷ No event
16: $\quad\quad$ **end if**
17: $\quad$ **else**
18: $\quad\quad \mathbf{x}_i.state \leftarrow 0$
19: $\quad$ **end if**
20: **end for**

longer than 50 m operating away from the shoreline, the average time gap between consecutive AIS points was approximately 33.97 mins. To ensure robustness and avoid overfitting to specific data characteristics, a conservative 30-minute threshold was adopted. Furthermore, tankers were excluded from the threshold calibration due to their dual roles (in-port service vs. transport), and cargo vessels were used as the reference group to ensure consistency and reliability.

The algorithm begins by identifying all trajectory points located within the boundary grids and marking them as target points (steps 1–3). For each target point, the algorithm computes the time interval between that point and its immediate neighbors. If a target point is located at the beginning or end of the trajectory sequence, the corresponding unavailable time interval is assigned a value of None (steps 6–9). A target point is classified as an entry event if the time gap with its preceding point exceeds the predefined threshold, or if it is the first point in the sequence and the time gap with the subsequent point is within the threshold (steps 10–11). Conversely, a target point is classified as an exit event if the time gap with the subsequent point exceeds the threshold, or if it is the last point in the sequence and the time gap with the preceding point is within the threshold (steps 12–13).

3.2.3. STASW: Spatio-temporal adaptive sliding windows for stop recognition

The stop point recognition algorithm is grounded in the assumption that vessel state transitions are temporally continuous, meaning that vessels do not frequently alternate between moving and stopping states over very short time intervals. As outlined in Section 3.1, it is critical to correct drifted points in advance to mitigate the effects of data anomalies and prevent misclassification. The core principle of the STASW method is to identify trajectory points that remain confined within a localized area over an extended time period, thus requiring joint consideration of both temporal and spatial windows. The method consists of two main components: algorithmic design and adaptive parameter selection.

Algorithm 2 is designed to identify vessel stop points by detecting periods during which a vessel remains within a spatially constrained region. It employs a fixed-size temporal sliding window to scan the trajectory and evaluates the spatial distribution and speed characteristics of points within that window. The input to Algorithm 2 includes a trajectory sequence sorted by timestamp along with relevant parameters. To reduce the computational complexity, the temporal window size ω is fixed, and adaptation focuses on the spatial characteristics of the points within each window.

The process begins by initializing the stop sliding window and defining an initial stop center. Longitude and latitude coordinates are transformed into Universal Transverse Mercator (UTM) coordinates to ensure accurate distance calculation (steps 1–2). As the window slides forward, the mean speed within the window is computed. If it falls below the speed threshold, the algorithm further examines the spatial dispersion of the points to determine whether a stop condition is satisfied. If so, the mean position is updated as the new stop center, and the current stop window is recorded (steps 12–19). Once a stop is detected, subsequent trajectory points are evaluated individually to verify whether they remain within the stop area. If they fall outside, the stop window is closed, and the stop event is finalized (steps 5–10). To address potential misclassification of underway points immediately before or after a stop, a secondary verification process is employed (see Appendix B). This process also categorizes detected stops as berth or anchorage. Two rechecking strategies are introduced: the first is optimized for high-quality AIS data and accurate parameter tuning, while the second offers greater robustness under general data conditions.

Algorithm 2 Stop point recognition.

Input: $\mathbf{x}$, the trajectory points sorted by timestamp; ω, window size; ϵ_v, speed threshold; ϵ_θ, course threshold; ϵ_δ, distance threshold.
Output: $\mathbf{x}$ with stop point marks for each point.

ω_{stop}, $\epsilon_{center} \leftarrow 0$, [0,0] ▷ Parameters ϵ_v, ϵ_θ, and ϵ_δ are adaptively adjusted according to the geographic location of trajectory points; see Table A.1 and related explanations for details.
$[\mathbf{x}_i.x, \mathbf{x}_i.y] \leftarrow$ TransformertoUTM($\mathbf{x}_i.\lambda$, $\mathbf{x}_i.\phi$) ▷ Convert to UTM coordinates for distance calculation.
for i from ω to len($\mathbf{x}$) **do**
  if $\omega_{stop} > 0$ **then**
    if $\mathbf{x}_i.\delta < \epsilon_\delta$ and distance between $\mathbf{x}_i$ and $\epsilon_{center} < \epsilon_\delta$ **then**
      $\omega_{stop} \leftarrow \omega_{stop} + 1$
    else
      recheck($\mathbf{x}_{i-\omega_{stop}:i}$) ▷ Secondary check of stop trajectories; see Appendix B for details.
      $\omega_{stop} \leftarrow 0$
    end if
  else
    if mean($\mathbf{x}_{i-\omega_{stop}:i}.v$) $< \epsilon_v$ **then**
      $x_{dis} \leftarrow$ max($\mathbf{x}_{i-\omega_{stop}:i}.x$) - min($\mathbf{x}_{i-\omega_{stop}:i}.x$)
      $y_{dis} \leftarrow$ max($\mathbf{x}_{i-\omega_{stop}:i}.y$) - min($\mathbf{x}_{i-\omega_{stop}:i}.y$)
      if $\sqrt{(x_{dis})^2 + (y_{dis})^2} < \epsilon_\delta$ **then**
        $\epsilon_{center} \leftarrow$ mean($\mathbf{x}_{i-\omega_{stop}:i}.p$)
        $\omega_{stop} \leftarrow \omega$
      end if
    end if
  end if
end for

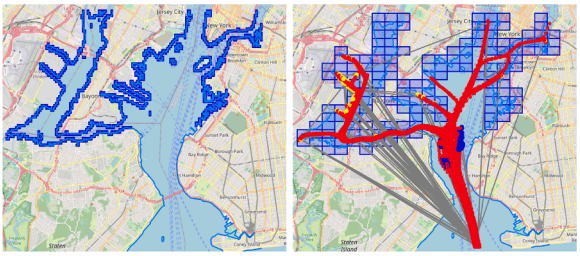


Fig. 4. Geohash encoding of the New York port coastline at level 7 (a) and level 6 (b).

Table A.1 summarizes the definitions and values of parameters used in the algorithm. Two distinct parameter sets are introduced to differentiate between berth and anchorage behavior. Since berthing occurs exclusively near the coastline, the algorithm selects the appropriate parameter set based on whether trajectory points fall within level 6 geohash grids that define the coastline region (Fig. 4 (b)). An adaptive strategy governs parameter selection. For each outer window ω_{out}, the proportion of trajectory points falling within the level 6 coastline geohash is calculated. If this proportion exceeds 50%, the "in-port" parameter set ϵ_v^{in}, $\epsilon_\delta^{\text{in}}$, and $\epsilon_\theta^{\text{in}}$ is applied; otherwise, the "out-of-port" parameter set is used. This proportion is updated in real time after processing each complete ω_{out} window, enabling dynamic adjustment while minimizing unnecessary switching and computational burden. Level 6 geohash cells are adopted as the threshold for parameter switching because, as discussed in Section 4.3.1, level 5 grids are overly coarse and may erroneously encompass inland or offshore regions, whereas level 7 grids are excessively fine and trigger unnecessary parameter adjustments. Consequently, level 6 provides an effective compromise, offering sufficient spatial resolution while maintaining stable and efficient parameter-switching behavior.

Berthing behaviors are typically characterized by low speeds and minimal spatial variation, allowing for detection using a relatively small temporal window ($\omega = 5$ in this study). In contrast, anchoring behaviors involve broader positional fluctuations, necessitating a larger window size for accurate detection. Other parameters are derived through separate statistical analyses of trajectory points, which are first classified into "in-port" and "out-of-port" groups based on level 7 geohash grids along the coastline (Fig. 4 (a)). Level 6 geohash grids are not used due to their inclusion of in-port channels where vessels maintain higher speeds to avoid congestion, potentially leading to misclassification.

Rather than relying on manually predefined thresholds, PortMiner adaptively determines key parameters from the statistical characteristics of the underlying data distribution. The key thresholds serve complementary functions: the speed threshold ϵ_v and distance threshold ϵ_δ are used in the initial stop detection, while the course threshold ϵ_θ supports the secondary verification process, helping

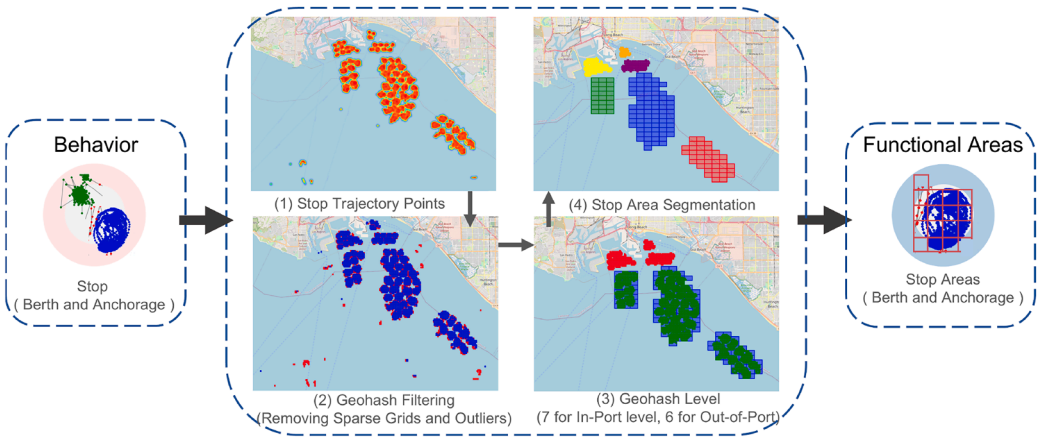


Fig. 5. Workflow for extracting vessel stop functional areas using Geohash filtering and clustering.

to eliminate false positives (i.e., moving states incorrectly identified as stops). These thresholds are selected as follows: ϵ_v is set as the mean speed of trajectory points within the window; ϵ_δ indicates the 99th percentile of pairwise distances between consecutive points; and $\epsilon_\theta^{\text{in}}$ is the 10th percentile of course-over-ground values within in-port segments. While this data-driven calibration reduces human intervention, the selection of statistical measures remains a heuristic choice rather than a purely self-learning process. Nevertheless, this adaptive parameter tuning strategy enhances the robustness and generalizability of the STASW algorithm across diverse operating environments and vessel behaviors.

To conclude the STASW design, a concise summary of its computational complexity is provided. Let N denote the number of trajectory points. With fixed and small window sizes (ω_{in}=5, ω_{out}=10), the core scan in Algorithm 2 processes each point exactly once. Window statistics (such as mean speed and bounding-box extrema) are maintained via $O(1)$ incremental updates, yielding an overall time complexity of $O(N)$ and an auxiliary space requirement of $O(1)$, aside from storage of the trajectory itself. Adaptive coastline switching, implemented using a level-6 geohash majority rule, incurs an additional $O(N)$ set of counter updates. The Type 1 secondary recheck runs linearly within each candidate segment and, when aggregated over all segments, also results in $O(N)$ complexity. The density-based Type 2 recheck achieves $O(N)$ expected time when neighborhood enumeration is performed via geohash bucketing, which bounds candidate expansion. Although a naive all-pairs variant would require $O(N^2)$, such worst-case behavior has not been observed in practice. Overall, STASW has expected linear time and space complexity and scales efficiently to AIS data streams containing millions of points. Full derivations and implementation details are provided in Appendix C.

3.3. Vessel functional area extraction

3.3.1. Vessel stop functional areas

Following the extraction and classification of vessel stop points into berth and anchorage categories (Section 3.2.3), this section focuses on identifying functional stop areas based on these points. Within port areas, berths serve a wide range of vessel types and sizes, each with distinct operational needs. Although a single berth may accommodate various vessel types, there is often a dominant usage pattern. For instance, large cargo vessels require container handling infrastructure such as cranes, while passenger vessels, fishing vessels, and small service vessels typically utilize different facilities. To capture both the functional diversity and dominant usage of berths, the number of stop points for each vessel type is counted within level 7 geohash grids. The vessel type with the highest proportion of stop points in each grid is displayed as the dominant type, although stop points from other vessel types are also retained in the grid data for comprehensive analysis.

In contrast, anchorage behaviors outside the port are more heterogeneous and complex. Vessels may anchor for various reasons, including waiting for berth availability, conducting fishing operations, or yielding to larger vessels. To improve the accuracy of anchorage area delineation and reduce the influence of short term or activity-related stops, only stop points associated with large vessels (length $>$ 50 m) are considered. As shown in Fig. 5, the level 7 geohash grids corresponding to these stop points are extracted, and their frequency is computed. Grids within the lowest 25th percentile by stop point count are treated as low-frequency stop areas and removed to suppress noise. Given the wider spatial distribution of anchorage zones, level 6 geohash grids are employed to reduce computational overhead and enhance spatial generalization. To further enhance the delineation of meaningful stop functional areas and facilitate spatial analysis and port management, a connectivity-based clustering method described in Section 3.3.3 is applied. This step groups adjacent geohash grids into spatially coherent sub-areas, while small isolated clusters, likely caused by drift or data anomalies, are excluded. The resulting stop functional areas are therefore both operationally relevant and robust to data variability.

3.3.2. Vessel inbound and outbound port channels

Based on the identification of vessel entry and exit events in Section 3.2.2, this section examines vessel traffic patterns to delineate dominant inbound and outbound port channels. Four spatial regions exhibiting a clear imbalance between entry and exit events were

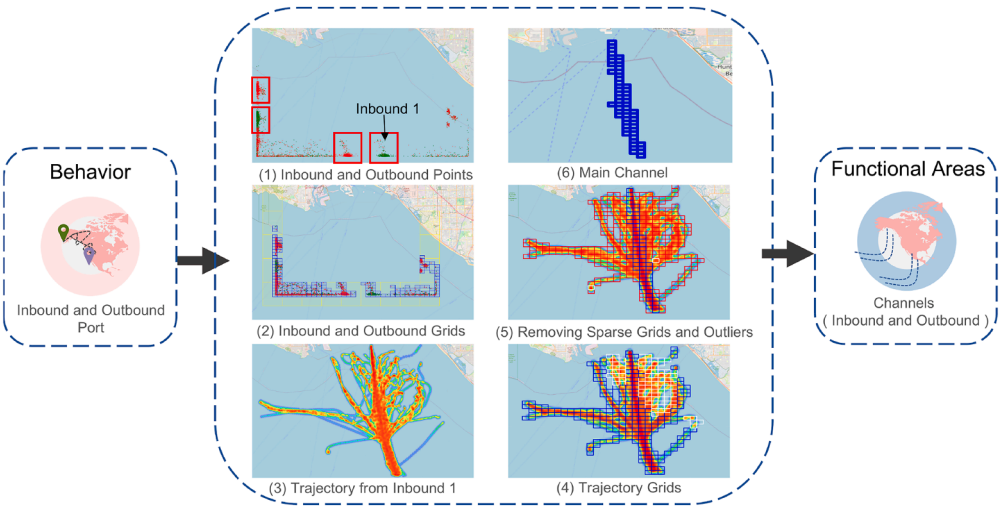


Fig. 6. Workflow for extracting vessel inbound and outbound port channels.

identified, suggesting the existence of two principal inbound and two outbound navigational channels (Fig. 6, step (1)). To support scalable spatial representation, particularly in the broader offshore context, level 6 geohash grids are used to encode the identified regions.

Channel extraction is conducted using boundary-crossing segments of large vessels (length > 50 m) after removing segments affected by drift. This design choice is motivated by three considerations. First, large vessels typically follow stable deep-water approaches with limited maneuvering, offering high-fidelity representations of primary navigational channels. Second, their boundary-crossing behavior is operationally unambiguous: once a large vessel enters or exits a port, it seldom re-crosses the boundary within a short interval. As a result, the extracted channel structure is robust to moderate variations in the time-gap threshold ϵ_t. Third, excluding towing, pleasure craft, and other small or service vessels suppresses the near-shore zigzagging commonly present in their movements, which otherwise introduces artificial fragmentation in the gridded representation and biases subsequent density-based filtering.

The methodological pipeline, illustrated using the Inbound 1 region as an example, involves extracting the trajectory segment from the vessel's entry point to its first stop within the port area. Many of these trajectories pass through anchorage zones before proceeding toward a berth, as visualized through heatmaps (see Step (4) in Fig. 6). These anchorage areas correspond to the regions identified in Section 3.3.1 and are highlighted by white grids in the figure. Notably, a subset of vessels enters the region but subsequently exits without proceeding further into the port, suggesting operational filtering or re-routing. To extract the main traffic channels, the number of trajectory points within each geohash grid is computed. Grids with point counts below the overall mean or those spatially isolated from high-density areas are filtered out to minimize noise. The remaining high-density grids are aggregated to delineate the primary inbound channel for the Inbound 1 region. The same procedure is applied to the remaining inbound and outbound zones, ensuring consistent and data-driven extraction of major navigational pathways. This approach produces channel representations that accurately reflect vessel flow patterns and remain resilient to sparse or outlier data.

3.3.3. Clustering based on geohash connectivity

To extract spatially coherent functional regions and eliminate scattered noise, Algorithm 3 implements a geohash-based connectivity clustering approach. The input to the algorithm is the set G of geohash codes corresponding to functional trajectory points, including berth, anchorage, and inbound/outbound locations. In the algorithm, each geohash code is treated as a node in a spatial graph, with edges defined by four-connectivity (i.e., adjacency through shared grid edges).

The algorithm employs a depth-first search (DFS) strategy to iteratively traverse and group connected geohash grids into sub-areas. For each unvisited geohash code g in G, a new sub-area is initialized, and DFS is recursively applied to identify all four-connected neighboring geohash codes that are also in G and have not yet been visited (steps 11–19). During the traversal, each visited geohash code is marked to avoid redundancy (step 12). Once all reachable nodes from a given starting point are explored, the sub-area is finalized and appended to the output list. This process continues until all geohash codes in G are assigned to a sub-area.

The output is a list of spatially connected sub-areas that reflect operational regions such as berthing zones, anchorage fields, and traffic corridors. Small isolated clusters, often resulting from data noise or drift, are filtered out post-clustering, ensuring that the final delineation remains both functionally meaningful and topologically robust.

4. Experimental results and analysis

To comprehensively evaluate the effectiveness of each component of the proposed PortMiner framework and to derive practical insights applicable to real-world maritime operations, a series of experiments were conducted using both annotated benchmark

Algorithm 3 Connected sub-area extraction.

Input: G, the set of geohash codes representing grids of interest.
Output: *sub_areas*, a list of all connected sub-areas (each is a group of spatially connected geohash codes).

```
visited ← ∅                                          ▷ Set of visited geohash codes
sub_areas ← [ ]                                      ▷ List to store all connected sub-areas
for each g ∈ G do
    if g ∉ visited then
        current_area ← [ ]
        DFS(g, current_area)
        sub_areas.append(current_area)
    end if
end for
function DFS(g, current_area)
    visited.add(g)
    current_area.append(g)
    neighbors ← GETNEIGHBORS(g)                      ▷ Obtain four-connected neighboring geohash codes
    for each n ∈ neighbors do
        if n ∈ G and n ∉ visited then
            DFS(n, current_area)
        end if
    end for
end function
```

Table 2
Summary of datasets utilized in this work.

Dataset	Description
Dataset 1	The New York dataset, preprocessed and annotated by Xin et al. (2024), contains labels for Berth (78.1%), Anchorage (4.8%), and Underway (17.0%). It is primarily utilized to assess the performance of vessel stop behavior recognition algorithms.
Dataset 2	AIS data filtered from NOAA (National Oceanic and Atmospheric Administration), covering the Port of Los Angeles (longitude: -118.36 to -118.0; latitude: 33.55 to 33.80) for the entire year 2021. After preprocessing and geohash encoding, this dataset is used for functional area extraction in real-world scenarios at the Port of Los Angeles.

datasets and real AIS data collected near the Port of Los Angeles. The benchmark datasets enable a quantitative comparison with existing methods for vessel behavior recognition, while the case study based on the Los Angeles port region demonstrates PortMiner's effectiveness in extracting port functional areas. Furthermore, the spatial output of PortMiner is cross-validated with official nautical charts to validate the accuracy and reliability of the proposed approach. This dual-pronged experimental design ensures a thorough assessment of PortMiner's performance in both controlled and operationally realistic settings.

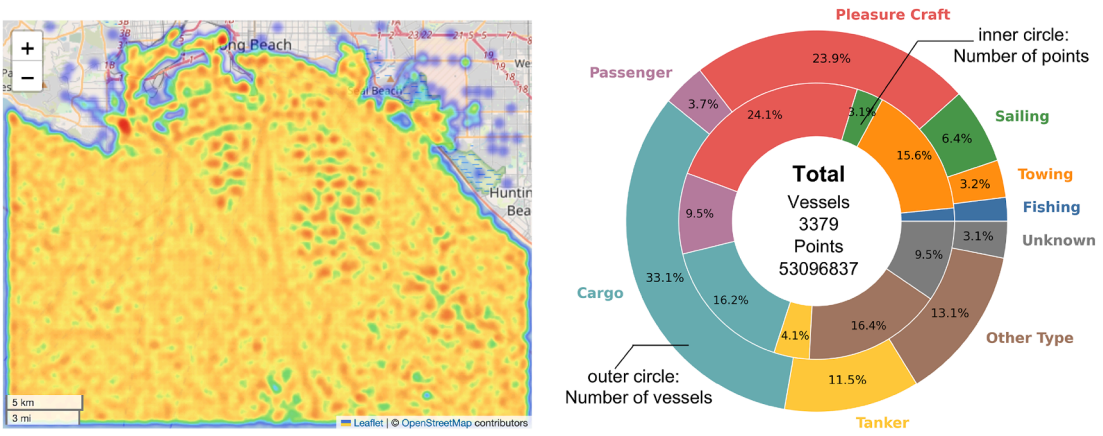
4.1. Experimental datasets

To evaluate PortMiner, two datasets are employed (see Table 2): (i) a benchmark dataset with labeled stop events by Xin et al. (2024), and (ii) a real-world AIS dataset collected from the Port of Los Angeles. Dataset 2 is used for case studies involving behavior identification and functional area extraction in the real-world maritime environments.

Dataset 1 (Port of New York, November 2021) contains AIS trajectories of cargo vessels operating within the Port of New York, comprising 186 vessels and 69,829 position reports. The AIS streams were pre-processed by Xin et al. (2024), including removal, interpolation, and enrichment with multi-source vessel attributes. Operational states (i.e., berth, anchorage, and underway) were annotated to form a high-quality benchmark. Owing to its controlled conditions and reliable state labels, this dataset is well suited for quantitative evaluation of stop-behavior recognition algorithms under standardized settings.

Dataset 2 (Port of Los Angeles[1], 1 January–31 December 2021, UTC) comprises raw, unfiltered AIS trajectories involving 3379 vessels and over 53.09 million position reports. As shown in Fig. 7 (a), the spatial distribution reveals faint traces of anchorage fields, while inbound and outbound channels are not visually discernible, reflecting the absence of clear navigational structure prior to processing. Fig. 7 (b) summarizes vessel-type distributions by both vessel count and trajectory points. In terms of vessel count, cargo ships form the largest group (39.06%), followed by pleasure craft (23.79%) and tankers (12.61%). By trajectory points, however, pleasure

[1] https://coast.noaa.gov/htdata/CMSP/AISDataHandler/2021/



(a) (b)

Fig. 7. Visualization of Dataset 2: (a) Heatmap showing the distribution of trajectory points in the Port of Los Angeles area; (b) Distribution of vessel types within the dataset.

craft dominate (24.00%), with cargo (21.37%) and towing vessels (19.34%) also exhibiting substantial presence. These pronounced class imbalances, combined with spatial structure and noise in the raw data, motivate the need for an unsupervised and parameter-adaptive framework. PortMiner is designed to automatically delineate functional stop areas from heterogeneous maneuvering patterns and extract coherent navigational channels without requiring labeled data or port-specific calibration.

4.2. Evaluation metrics

To effectively evaluate PortMiner, stop behavior recognition performance is assessed using standard classification metrics such as Accuracy, Precision, Recall, and F1-score. Functional area extraction quality is evaluated through structural quality and consistency with official Electronic Nautical Charts (ENCs), including channel continuity indicators.

4.2.1. Vessel stop behavior recognition metrics

The performance of stop behavior recognition methods is assessed using standard classification metrics: Accuracy, Precision, Recall, and F1-score (Bishop and Nasrabadi, 2006). These indicators collectively measure overall correctness, error tendencies, and balance between detection sensitivity and false-positive suppression. Their formal definitions are provided below:

- **Accuracy** quantifies the proportion of correctly identified samples relative to the total number of samples, providing an overall measure of classification correctness.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

where TP denotes true positives, TN true negatives, FP false positives, and FN false negatives.

- **Precision** reflects the proportion of predicted positive cases that are truly positive, indicating the method's ability to avoid false alarms.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

- **Recall** measures the proportion of actual positive cases that are correctly detected, capturing the model's sensitivity to true stop behavior.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

- **F1-score** is the harmonic mean of Precision and Recall, offering a balanced indicator when the cost of false positives and false negatives should be jointly considered.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

4.2.2. Port functional area extraction metrics

Functional-area quality is evaluated from two perspectives: (i) the morphological coherence of channel-like structures inferred from geohash cells, and (ii) the agreement with official ENCs.

(i) Morphology-based metrics (Grid graph).

Let $\mathcal{S}$ be the set of selected geohash cells for a target structure (e.g., channel). A 4-neighborhood grid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ on $\mathcal{S}$ is constructed, where two cells are adjacent if they share an edge. Let $\{\mathcal{C}_k\}_{k=1}^{K}$ be the connected components of $\mathcal{G}$, $N = |\mathcal{V}|$ the number of cells, and

$$L_{\text{skel}} = \sum_{k=1}^{K} \max\left(|\mathcal{C}_k| - 1,\, 0\right), \tag{6}$$

which serves as a proxy for the skeleton length (minimal within-component links). Four indicators are applied:

- **Fragmentation** (frag): fraction of cells outside the largest component,

$$\text{frag} = \begin{cases} 0, & N = 0, \\ 1 - \dfrac{\max_k |\mathcal{C}_k|}{N}, & N > 0. \end{cases} \tag{7}$$

  Lower values indicate a more coherent, less fragmented structure.
- **Spur ratio** (spur): proportion of degree-1 nodes,

$$\text{spur} = \begin{cases} 0, & N = 0, \\ \dfrac{\left|\{\, v \in \mathcal{V} : \deg_{\mathcal{G}}(v) = 1 \,\}\right|}{N}, & N > 0. \end{cases} \tag{8}$$

  This captures dangling ends and spurious branches; lower is better.
- **Gap density** (gap): scale-normalized density of discontinuities between components,

$$\text{gap} = \begin{cases} 0, & L_{\text{skel}} = 0, \\ \dfrac{\max(K-1, 0)}{L_{\text{skel}}}, & L_{\text{skel}} > 0, \end{cases} \tag{9}$$

  Interpretable as the number of breaks per skeleton step; lower implies higher channel continuity.
- **Skeleton steps** (steps): L_{skel}, used as a size/length proxy and as the normalizer for gap.

All quantities are computed on the 4-neighborhood induced by geohash cells. The adjacency construction is $O(N)$ (constant-degree grid), and node degrees are obtained by a single pass over neighbors.

(ii) Chart-consistency metrics (ENCs alignment).

Let $\widehat{\mathcal{R}}, \mathcal{R} \subset \Omega$ be the predicted and reference (ENCs-derived) regions in a common domain Ω. Areal overlap and axis-level geometry properties are quantified using the following measures:

- **Intersection-over-Union (IoU):**

$$\text{IoU}(\widehat{\mathcal{R}}, \mathcal{R}) = \frac{|\widehat{\mathcal{R}} \cap \mathcal{R}|}{|\widehat{\mathcal{R}} \cup \mathcal{R}|}, \tag{10}$$

  with the convention IoU = 1 if both regions are empty. Higher is better.
- **Normalized centerline distance** (HD_{norm}): extract centerlines $\widehat{\Gamma}, \Gamma \subset \Omega$ (e.g., morphological skeletons) and compute the symmetric Hausdorff distance

$$\text{HD}(\widehat{\Gamma}, \Gamma) = \max\left\{ \sup_{x \in \widehat{\Gamma}} \inf_{y \in \Gamma} \|x - y\|,\ \sup_{y \in \Gamma} \inf_{x \in \widehat{\Gamma}} \|x - y\| \right\}. \tag{11}$$

  To compare across scales, normalize by a characteristic length D of Ω (e.g., the diagonal of the evaluation bounding box):

$$\text{HD}_{\text{norm}} = \frac{\text{HD}(\widehat{\Gamma}, \Gamma)}{D} \in [0, 1]. \tag{12}$$

  Lower HD_{norm} indicates better geometric agreement of channel axes.

Collectively, these indicators provide a systematic assessment of the model's correctness and robustness in detecting vessel stop behaviors and extracting functional areas.

4.3. Parameter settings and optimisation

This section synthesises the key design choices and parameter configurations in PortMiner. It outlines the selection of geohash resolution for coastline representation, the calibration of the time-gap threshold used in inbound/outbound state recognition, and the data-driven derivation of temporal windows and adaptive thresholds in STASW. The use of hierarchical geohash levels for delineating stop areas is detailed, along with the resolution choices and density criteria employed for channel extraction. For each component, the rationale and parameter sensitivity are summarized to clarify their functional role in capturing functional port structures. The corresponding geohash levels and spatial resolutions are listed in Table 3.

Table 3
Geohash levels 1–8: spatial resolution and approximate positional error.

Level	Lat bits	Lon bits	Lat error (°)	Lon error (°)	Approx. error (km)
1	2	3	±23	±23	±2500
2	5	5	±2.8	±5.6	±630
3	7	8	±0.70	±0.7	±78
4	10	10	±0.087	±0.18	±20
5	12	13	±0.022	±0.022	±2.4
6	15	15	±0.0027	±0.0055	±0.61
7	17	18	±0.00068	±0.00068	±0.076
8	20	20	±0.000086	±0.000172	±0.01911

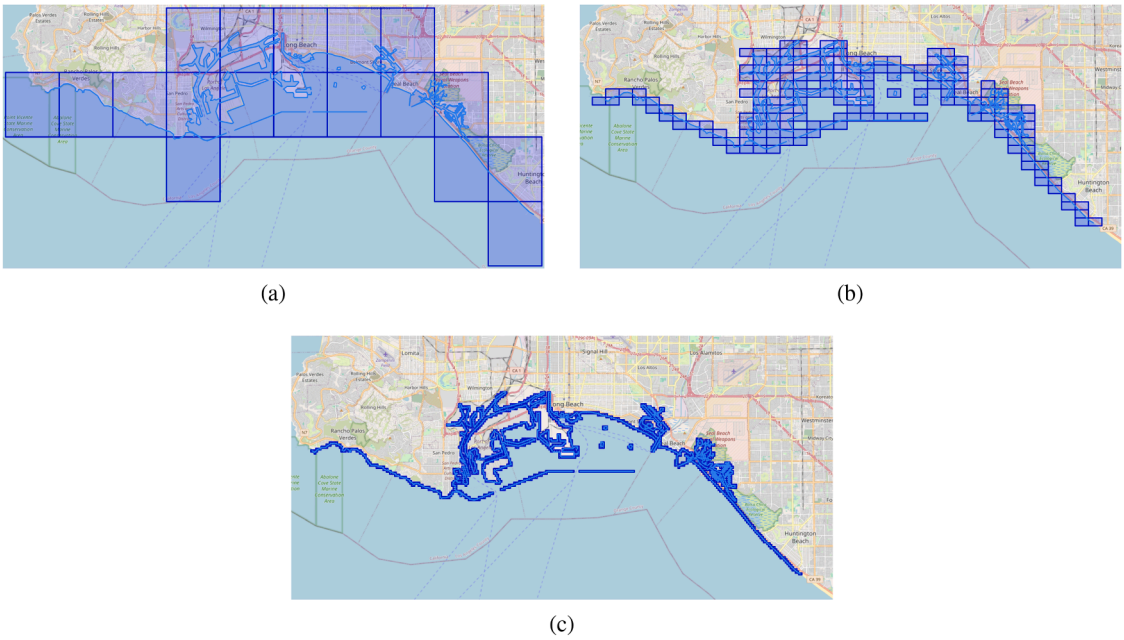


Fig. 8. Comparison of coastline geohash encoding at levels 5–7.

4.3.1. Geohash resolution selection for coastline encoding

Selecting an appropriate geohash resolution of coastline encoding requires balancing spatial fidelity with operational stability. As illustrated in Fig. 8, level 5 masks substantially over-cover offshore waters; using this resolution for in-port/out-of-port switching would incorrectly classify many offshore trajectories as in-port, degrading stop-recognition performance. Conversely, level 7 masks fragment navigable waterways into discontinuous narrow strips, leading to frequent oscillation between parameter sets during continuous voyages and compromising temporal stability required for STASW. level 6 provides a topologically coherent representation of the coastline along major shipping lanes and supports robust, low-frequency parameter switching.

From an encoding precision standpoint (Table 3), level 5 cells (≈2.4 km) exceed the transverse width of multi-channel widths and merge shoreline and offshore waters into coarse, mixed units. Level 7 cells (≈76 m), by contrast, are comparable to or smaller than the widths of individual channels or berths, causing spurious fragmentation of otherwise continuous lanes. Level 6 cells (≈610 m) provide an effective compromise: they span the full cross-section of major channels while preserving longitudinal continuity, thereby avoiding the spatial over-aggregation of level 5 and the excessive segmentation seen at level 7.

Efficiency measurements in Table 4 further reinforce this choice. The number of coastline cells increases from 191 at level 6 to 1,885 at level 7 (approximately tenfold), and the end-to-end per-point encoding plus lookup time rises from 7.1×10^{-5} s to 8.8×10^{-5} s. Although the absolute increment is small per operation, the cumulative overhead becomes significant at port scale and, combined with the instability induced by frequent switching, reduces overall throughput. Accordingly, level 6 is adopted as the default resolution for adaptive in-port/out-of-port parameter switching in STASW (Section 3.2.3). Once a stop is detected, however, the berth-versus-anchorage decision is localized near the coastline and does not require ongoing parameter adaptation; thus, a higher-resolution level 7 mask is used for this one-off classification (Appendix B).

4.3.2. Time-gap threshold analysis for inbound/outbound recognition

The objective of inbound/outbound recognition is to enable robust extraction of the principal entry and exit channels, rather than to model AIS reporting gaps directly. Accordingly, the evaluation of ϵ_t focuses on its downstream impact on the number of usable

Table 4
Coastline geohash levels: mask size and lookup efficiency (seconds).

Level	Geohash cells	Encode (s)	Land check (s)	Total (s)
5	18	0.000041	0.000019	0.000060
6	191	0.000048	0.000023	0.000071
7	1885	0.000050	0.000038	0.000088

Table 5
Time-gap thresholds (10–60 mins): extracted channel segments and trajectory points per route. The default setting is 30 mins.

ϵ_t (min)	Segments					Points				
	entry 0	entry 1	exit 0	exit 1	total	entry 0	entry 1	exit 0	exit 1	total
10	101	166	240	300	807	10,450	14,348	15,861	18,403	59,062
20	101	166	240	299	806	10,450	14,348	15,863	18,329	58,990
30	99	164	240	299	802	10,346	14,337	15,863	18,329	58,875
40	96	163	240	300	799	10,230	14,225	15,863	18,400	58,718
50	96	158	240	299	793	10,230	14,100	15,863	18,330	58,523
60	93	152	242	300	787	10,217	14,071	16,021	18,403	58,712

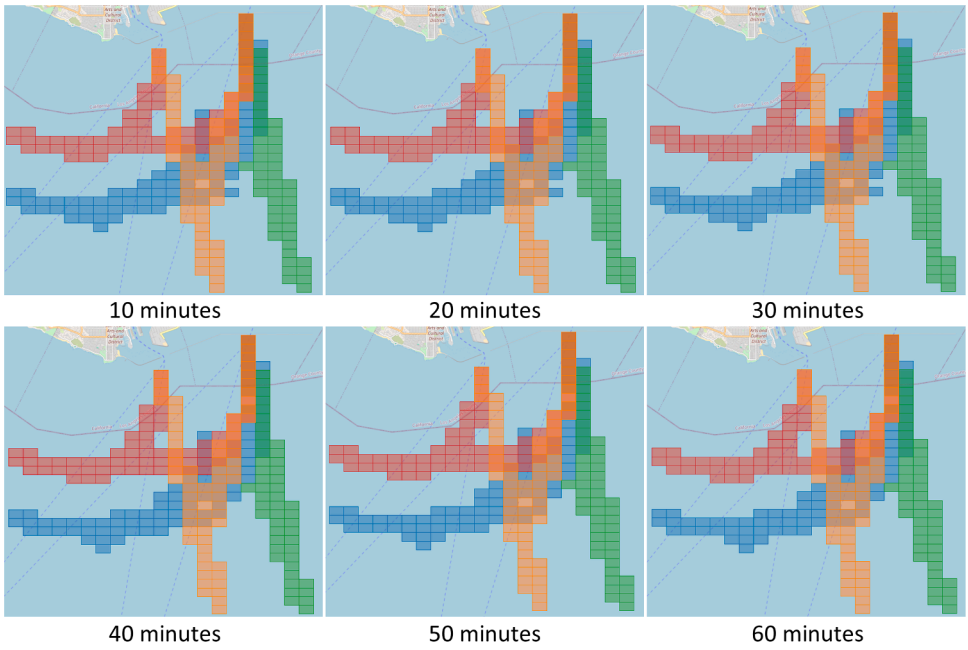


Fig. 9. Extracted inbound/outbound channels under different time-gap thresholds (10–60 mins).

(drift-free and non-missing) boundary-crossing trajectory segments generated by large vessels (length > 50 m), which dominate port-approach traffic and provide the most stable channel geometry.

Table 5 summarizes segment- and point-level statistics for ϵ_t values ranging from 10 to 60 mins. The results exhibit substantial stability across this range. The total number of segments varies only from 787 to 807 (mean approximately 799, within 1.5%), and the total number of points fluctuates by less than 0.5% (58,523–59,062). Per-route trends exhibit similar consistency. Exit 1 remains constant at 300 segments; Exit 0 varies minimally between 240 to 242; and Entry 0/1 show moderate reductions (approximately 8–9%) as shorter temporal gaps are merged under larger ϵ_t values are applied. Notably, the default 30-minute threshold produces 802 segments and 58,875 points, positioned near the center of this stability region. These observations confirm that the chosen ϵ_t value does not materially influence the structure of large-vessel boundary-crossing trajectories and therefore does not affect the robustness of subsequent channel-extraction procedures.

Fig. 9 illustrates channels extracted using $\epsilon_t = 20$ and 30 mins under identical level 6 geohash and density settings. Extraction results under other parameter combinations exhibit only minor differences. The resulting channel geometries are topologically identical, with indistinguishable centerline and uninterrupted spatial continuity. This demonstrates that, as long as ϵ_t falls within a broad operational range (10–60 mins), the effective set of large-vessel boundary-crossing segments is sufficient to reconstruct the same inbound and outbound corridors.

Table 6
Temporal window sweep: paired in-port/out-of-port settings and performance.

ω_{in}	ω_{out}	Accuracy	Precision	Recall	F1
5	5	98.803	98.920	98.930	98.925
5	15	**98.831**	**98.958**	98.943	**98.950**
10	10	98.687	98.634	**99.012**	98.823
15	15	98.602	98.528	98.969	98.748
5	10	98.801	98.940	98.907	98.923
5	15	98.572	98.487	98.956	98.721
3	5	98.809	99.001	98.858	98.929

Table 7
Course-angle thresholds: statistical sources and paired in/out (degrees).

Source	θ_{in}	θ_{out}	Accuracy	Precision	Recall	F1
P10 (out = 1 × in)	16.0	16.0	98.747	98.756	98.997	98.876
P10 (out = 2 × in)	16.0	32.0	98.831	98.958	98.943	98.950
P10 (out = 3 × in)	16.0	48.0	**98.857**	**99.072**	98.873	**98.973**
P5 (out = 2 × in)	5.5	11.0	98.682	98.614	**99.025**	98.819
P5 (out = 3 × in)	5.5	16.5	98.754	98.756	99.010	98.883
P20 (out = 2 × in)	41.0	82.0	98.804	99.239	98.608	98.923
P20 (out = 3 × in)	41.0	120.0	98.743	99.261	98.475	98.866

4.3.3. STASW parameter sensitivity analysis

This section evaluates the sensitivity of STASW performance to four principal parameters, including the temporal window sizes ω_{in} and ω_{out} used for in-port and out-of-port states, as well as the adaptive derivation mechanisms for the remaining thresholds. All experiments are conducted on Dataset 1 (Port of New York), with adaptive in-port/out-of-port parameter switching enabled via level 6 coastline geohash encoding (Section 4.3.1).

As shown in Table 6, model performance is largely invariant across different window sizes. Accuracy remains within 98.57–98.83 (Δ=0.26), and F1-score within 98.72–98.95 (Δ=0.23), indicating that the method is weakly sensitive to the choice of ω. Larger windows beyond $(10, 10)$ yield mild degradation in Precision and F1 due to over-smoothing near transition intervals, whereas a symmetric $(10, 10)$ configuration achieves the highest Recall (99.01) within a small reduction in Precision. Shorter windows such as $(3, 5)$ also perform competitively (F1= 98.93). The configuration $(5, 15)$ is adopted as the default because it provides the best combined Accuracy and F1, provides greater robustness to drift during anchorage events, and maintains stable temporal responsiveness without inducing oscillatory parameter switching.

Table 7 demonstrates that STASW is even less sensitive to heading variability than to temporal window size. Accuracy varies only within $98.682 - 98.857$ ($\Delta = 0.175$) and F1 within $98.819 - 98.973$ ($\Delta = 0.154$), both narrower than the corresponding ranges observed in the window-size sweep (Accuracy $\Delta = 0.26$, F1 $\Delta = 0.23$ in Table 6). This limited variation indicates that speed- and displacement-based constraints in STASW already eliminate most borderline underway states, and that heading dispersion mainly performs fine-grained pruning at stop boundaries, yielding only incremental improvements. Extremely tight in-port thresholds (P5) slightly increase Recall but reduce Precision due to over-fragmentation of stops, whereas overly permissive multipliers (P20 with 3×) admit low-speed maneuvering to be misclassified as stopping, lowering Precision without material Recall gains. The P10 (out = 3× in) setting (16°/48°) is therefore selected as the final configuration. It achieves the highest composite F1 (98.973) and Precision (99.072) while maintaining competitive Recall (98.873), offering a robust margin for anchorage drift without inflating false positives.

Table 8 compares statistical sources for the speed threshold ϵ_v under paired in-/out-of-port settings. Unlike the window (Table 6) and course sweeps (Table 7), the speed threshold exhibits the largest performance variability: Accuracy ranges 96.25–98.83 and F1 spans 96.60–98.95. Mean-based thresholds consistently outperform median- and percentile-based variants, indicating that the mean more effectively captures the central tendency of low-speed stop windows while remaining robustness to moderate drift noise in offshore environments. Configurations with very small offshore multipliers (out = 1×in–2×in) yield lower Precision (96.76–96.80), whereas overly permissive settings (out = 4×in) maximize Recall (99.61) but further dilute Precision (96.61). The balanced choice is the mean with ϵ_v^{in}=0.12 m/s and ϵ_v^{out}=3 × ϵ_v^{in}=0.37 m/s, delivering the best overall Accuracy/F1 (98.83/98.95) with a strong Precision-Recall trade-off (98.958/98.943). This configuration is therefore adopted as the final speed threshold setting (see Table A.1).

Table 9 presents the results of varying the distance threshold δ using different statistical sources. Similar to speed, distance exhibits a relatively wide performance range: Accuracy spans 96.66–98.83 and F1 varies 96.93–98.95. The P99-based setting (δ_{in}=135 m, δ_{out}=551 m) achieves the best overall performance (Accuracy 98.831, F1 98.950) with a balanced Precision-Recall profile (98.958/98.943). Lower percentiles (P98, P97, P95) progressively relax the thresholds, leading to diminished Accuracy and F1 due to increased false positives from maneuvering noise. The P99-based thresholds are therefore selected as the final configuration for δ in STASW (see Table A.1).

Table 8
Speed thresholds: statistical sources and paired in/out (m/s).

Source	v_{in}	v_{out}	Accuracy	Precision	Recall	F1
Mean (out = 1× in)	0.12	0.12	97.863	96.798	99.452	98.107
Mean (out = 2× in)	0.12	0.25	97.908	96.761	99.576	98.148
Mean (out = 3× in)	0.12	0.37	**98.831**	**98.958**	98.943	**98.950**
Mean (out = 4× in)	0.12	0.49	97.839	96.612	**99.612**	98.089
Median (out = 3× in)	0.05	0.14	96.974	97.664	96.883	97.272
P60 (out = 3× in)	0.06	0.19	97.149	97.531	97.343	97.437
P40 (out = 3× in)	0.04	0.11	96.254	97.746	95.473	96.596

Table 9
Distance thresholds: statistical sources and paired in/out (m).

Source	δ_{in}	δ_{out}	Accuracy	Precision	Recall	F1
P99	135	551	**98.831**	98.958	**98.943**	**98.950**
P98	87	512	98.701	99.024	98.639	98.831
P97	64	494	98.125	99.065	97.554	98.304
P95	45	464	96.658	**99.079**	94.879	96.933

Table 10
Geohash levels 5–8: construction time and per-point retrieval efficiency for stop functional area delineation.

Level	Construction					Retrieval	
	Encode + Count (s)	Aggregation (s)	Total (s)	Clusters	Geohash Cells	Encode(s)	Lookup(s)
5	31.3398	0.0002	44.3514	1	15	0.000060	0.000080
6	37.6420	0.0012	50.7841	3	176	0.000063	0.000082
7	43.2658	0.0122	56.4713	9	3753	0.000062	0.000132
8	48.7662	0.2627	62.2514	54	78,763	0.000061	0.001218

Table 11
Stop functional areas: morphology metrics across geohash levels (4-neighborhood).

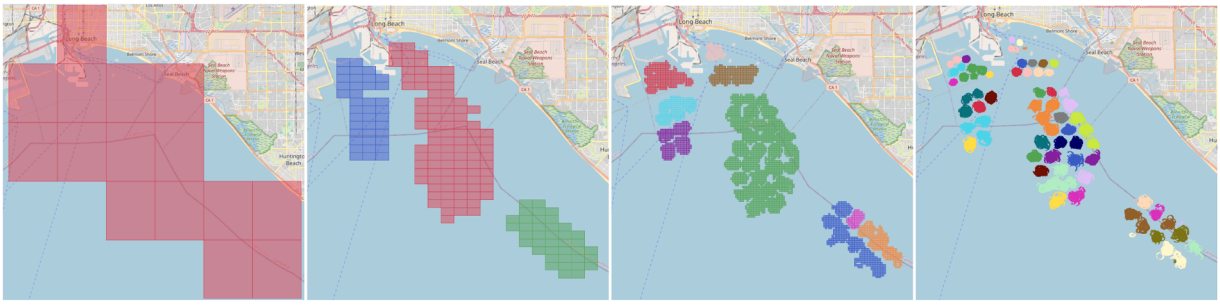
Level	Cells	Clusters	frag	spur	gap	steps
5	15	1	0.000	0.067	0.0000	14
6	176	3	0.443	0.011	0.0116	173
7	3753	37	0.536	0.015	0.0097	3716
8	78,763	522	0.927	0.013	0.0067	78,241

4.3.4. Geohash hierarchies for stop functional area delineation

Table 10 summarizes construction and per-point lookup costs for geohash levels 5–8. Encode-count time increases only moderately (31.34 s to 48.77 s; +55.6%), reflecting the essentially linear cost of hashing. The main escalation arises in the aggregation stage: at level 8 the time rises to 0.2627 s (≈ 21.5× higher than level 7) due to the explosion of fine cells requiring connectivity consolidation. The number of geohash cells (15 to 78,763) and clusters (1 to 54) grows super-linearly, indicating pronounced spatial fragmentation. Lookup latency follows the same pattern, increasing from 0.000132 s at level 7 to 0.001218 s at level 8 (≈ 9.2×), as candidate scans expand with the enlarged index. Morphology metrics (Table 11) confirm severe over-segmentation at level 8: fragmentation increases from 0.536 to 0.927, while spur and gap reductions diminish. Visual inspection (Fig. 10d) further shows that vessels frequently remain stationary between adjacent berths; extreme spatial granularity at level 8 isolates these transitional micro-regions and treats them as separate stop areas.

Lower resolutions (levels 5–6) are computationally light but insufficiently expressive. Level 5 (Fig. 10a) is overly coarse and merges extensive land areas with offshore zones; level 6 (Fig. 10b) still fails to delineate in-port stop regions accurately, with anchorage-level boundaries frequently extending into land. Level 7 achieves a more balanced operating point: total construction time remains manageable (56.47 s), per-point lookup remains well below one millisecond, spatial resolution is adequate to separate adjacent berths, and cluster counts remain stable enough for reliable connectivity analysis. However, Fig. 10 (c) indicates that level 7 cells can still miss fine-scale oscillatory motion near berth edges, particularly in offshore anchorage zones, where boundary jitter may exceed the cell width.

To balance the differing spatial and operational requirements of in-port and offshore environments, a hybrid geohash strategy is adopted: level 7 is used for in-port stop delineation, where fine spatial discrimination is essential to resolve berth-scale structures, whereas level 6 is applied to offshore anchorage zones, where coarser cells effectively suppress positional noise and enhance temporal stability. Level 8 is excluded from routine or real-time use because it produces excessive spatial fragmentation and incurs significant lookup overhead. Conversely, level 5 does not provide sufficient spatial fidelity for reliable in-port stop characterization. This hybrid design therefore achieves a suitable balance between resolution, robustness, and computational efficiency.



(a) (b) (c) (d)

Fig. 10. Stop functional area delineation using geohash levels (a) 5, (b) 6, (c) 7, and (d) 8.

Table 12
Geohash levels 5–8: construction time and per-point retrieval efficiency for channel delineation.

Level	Construction				Retrieval
	Encode + Count (s)	Aggregation (s)	Total (s)	Geohash Cells	Lookup / point (s)
5	0.2576	0.0002	0.2721	30	0.000003
6	0.3207	0.0011	0.3349	312	0.000009
7	0.3632	0.0234	0.4062	7309	0.000226
8	0.4517	0.0209	0.5053	6889	0.000137

4.3.5. *Geohash level strategy for inbound/outbound channel extraction*

Table 12 evaluates geohash levels 5–8 for channel delineation. Level 5 is too coarse relative to typical channel widths (Table 3), producing inflated footprints that absorb extensive non-channel waters. Level 6 provides a markedly better balance: it preserves longitudinal continuity without shoreline spillover, requires modest construction time (0.3349 s), and maintains very low lookup latency (9×10^{-6} s/point). Finer levels introduce substantial overhead: lookup cost rises by $\approx 25\times$ at level 7 (2.26×10^{-4} s/point) and remains an order of magnitude above level 6 at level 8 (1.37×10^{-4} s/point), despite similar build times.

Morphology metrics in Table 13 further indicate that level 6 achieves the most consistent structural fidelity: IoU spans $[0.855, 0.909]$, $\mathrm{HD}_{\mathrm{norm}}$ remains low ($[0.036, 0.072]$), and fragmentation is zero across route components. Level 7 begins to over-segment corridor edges, yielding fragmentation and narrow spurs (e.g., Inbound 0: $\mathrm{frag} = 0.489$, $\mathrm{spur} = 0.132$). Level 8 undergoes severe structural breakup ($\mathrm{frag} \geq 0.912$) and minimal chart overlap (IoU ≤ 0.109), indicating that channel signatures collapse under excessive granularity. These findings are consistent with the efficiency analysis, which shows that level 6 provides the best balance between geometric fidelity and computational stability, preserving channel morphology without introducing grid-induced artifacts. It is noted that the exact coordinates of ENCs channel polygons are not publicly released; therefore, channel boundaries used for comparison were manually aligned for evaluation purposes. This alignment process may introduce small positional discrepancies in the IoU and HD measurements, although the overall topological correspondence between extracted and reference channels remains robust.

Fig. 11 illustrates the mechanisms behind degradation at fine resolutions. At level 7, vessel slowdown and lateral dispersion near berths make per-cell counts highly localized; en-route motion concentrates near the centerline, leaving sparse edges. The resulting grid extracts a narrow, discontinuous axis with enlarged inter-segment gaps. At level 8, dispersion dominates and the main-channel signature fragments into micro-cells. Conversely, level 5 over-expands the footprint, masking corridor shape by merging heterogeneous waters.

Inbound-outbound asymmetry is also evident in Fig. 11: inbound flows often traverse anchorage fields before berthing, amplifying densities around anchorage-connected approaches, whereas outbound tracks follow more direct and distributed branches. Level 6 captures this asymmetry while maintaining continuous axes and strong chart agreement, consistent with the high IoU and low $\mathrm{HD}_{\mathrm{norm}}$ in Table 13.

Considering accuracy, structural coherence, and computational efficiency, level 6 is adopted as the default for inbound/outbound channel extraction. Level 7 and level 8 are not recommended due to unstable, overly narrow axes and degraded chart consistency, whereas level 5 is insufficiently detailed to isolate channel geometry.

4.4. *Accuracy of vessel stop behavior recognition*

To assess the reliability of the behavioral inference module, this subsection evaluates the accuracy and robustness of STASW in recognizing vessel stop behavior. Using the annotated Dataset 1 from the Port of New York, STASW is evaluated against representative rule-based baselines, clustering-based methods, and graph neural network models. The subsequent analyses provide a comprehensive comparison of performance, assess parameter sensitivity, and quantify the improvements achieved through the proposed adaptive refinement strategies.

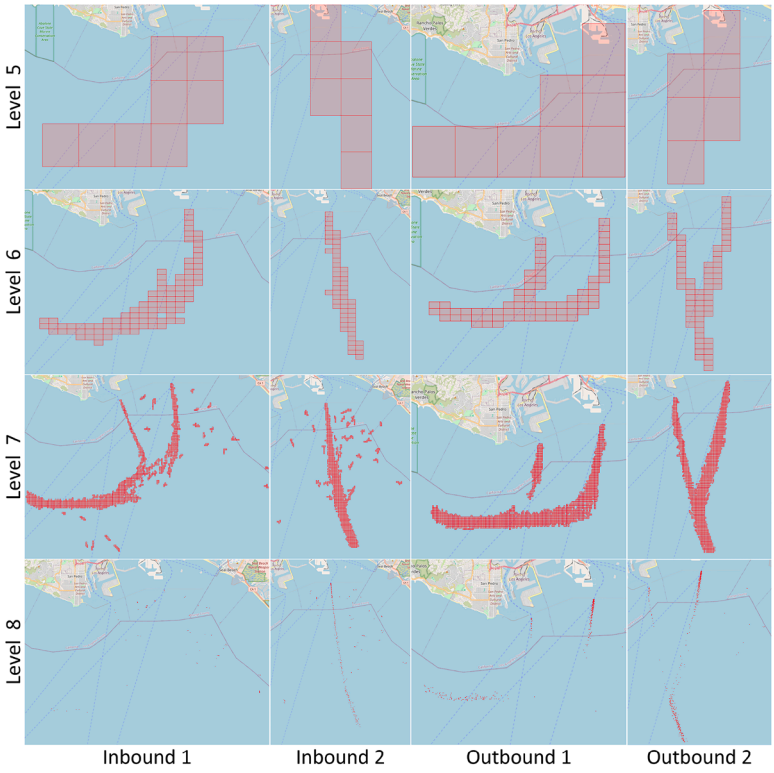


Fig. 11. Inbound/outbound channel extraction using geohash levels 5–8.

Table 13
Channels: morphology and chart-consistency metrics across geohash levels (4-neighborhood).

Level	Route	Cells	Clusters	frag	spur	gap	steps	IoU	HD_{norm}
5	Inbound 1	8	1	0.000	0.125	0.0000	7	0.4015	0.4253
	Inbound 2	7	1	0.000	0.286	0.0000	6	0.2768	0.5968
	Outbound 1	8	1	0.000	0.250	0.0000	7	0.4957	0.4667
	Outbound 2	6	1	0.000	0.333	0.0000	5	0.4022	0.4221
6	Inbound 1	96	1	0.000	0.042	0.0000	95	0.9093	0.0357
	Inbound 2	56	1	0.000	0.071	0.0000	55	0.8970	0.0723
	Outbound 1	74	1	0.000	0.027	0.0000	73	0.8553	0.0554
	Outbound 2	71	1	0.000	0.042	0.0000	70	0.8958	0.0362
7	Inbound 1	1475	42	0.489	0.132	0.0286	1433	0.7847	0.1573
	Inbound 2	1356	28	0.203	0.100	0.0203	1328	0.8716	0.1486
	Outbound 1	1510	2	0.103	0.056	0.0007	1508	0.8145	0.1324
	Outbound 2	1534	1	0.000	0.048	0.0000	1533	0.7678	0.1291
8	Inbound 1	185	60	0.941	0.649	0.4720	125	0.0518	0.2189
	Inbound 2	800	268	0.984	0.677	0.5019	532	0.1088	0.1747
	Outbound 1	564	196	0.947	0.725	0.5299	368	0.0502	0.1728
	Outbound 2	1049	318	0.912	0.642	0.4337	731	0.0059	0.2342

4.4.1. Baseline methods

A systematic evaluation of seven baseline methods was conducted, encompassing rule-based strategies, unsupervised techniques, and graph-based learning approaches, as presented in Table 14. These baselines represent state-of-the-art approaches across various categories of vessel behavior recognition.

4.4.2. Classification accuracy

As shown in Table 15, the proposed STASW method demonstrates superior performance across all evaluation metrics, achieving the highest accuracy of 98.857% and an F1-score of 98.973%, surpassing all benchmarked models. In comparison, the rule-based SBC approach achieved a respectable accuracy of 97.258%; however, its relatively lower precision suggests a tendency to overpredict stop points. The DBSCAN clustering algorithm yielded significantly lower accuracy (81.224%) and recall, reflecting its sensitivity to parameter settings and its difficulty in handling varying spatial densities along vessel trajectories.

Table 14
Summary of baseline methods.

Method	Category	Key Characteristics	Reference
GCN	Graph-based Learning	Graph Convolutional Networks (GCN); captures global structural features; requires predefined graph structure; limited capacity for modeling temporal dependencies	Kipf (2016)
ChebNet	Graph-based Learning	Chebyshev Spectral CNN (ChebNet); captures multi-hop dependencies efficiently; sensitive to graph structure changes	Defferrard et al. (2016)
GAT	Graph-based Learning	Graph Attention Networks (GAT); adaptive node weighting; selectively integrates informative local features; requires significant labeled data	Veličković et al. (2017)
GraphSAGE	Graph-based Learning	Graph Sample and Aggregate Networks (GraphSAGE); handles heterogeneous node types; effective with sparse connectivity patterns	Liu et al. (2020)
SBC	Rule-based	Uses Speed-Based Constraint (SBC) approach to identify stop; low computational requirements; highly sensitive to threshold selection; limited adaptability to complex scenarios	Yan et al. (2022)
DBSCAN	Clustering-based	Aggregates spatially dense trajectory points to detect stop; capable of identifying irregular stop patterns; highly parameter-dependent; struggles with varying density distributions	Rong et al. (2024)
FDSVBR	Rule-based	Ultra-Fast and Data-Efficient Single-Vessel Behavior Recognition (FDSVBR) combines speed and distance constraints within sliding windows; offers higher robustness than single-parameter methods	Qiang et al. (2025b)

Table 15
Performance comparison of various approaches.

Method	Accuracy	Precision	Recall	F1-score
DBSCAN	81.224	97.716	67.836	80.080
SBC	97.258	96.119	99.071	97.573
GAT	98.353	97.603	**99.483**	98.534
GCN	98.173	97.629	99.122	98.370
ChebNet	97.545	97.050	98.584	97.811
FDSVBR	98.738	98.707	99.030	98.868
GraphSAGE	98.807	98.421	99.452	98.933
STASW	**98.857**	**99.072**	98.873	**98.973**

Table 16
Test-set performance: STASW versus the best-performing GNN configuration per model (selected from Appendix E).

Model	Accuracy	Precision	Recall	F1
ChebNet	98.840	99.120	99.329	99.225
GAT	98.596	98.651	99.483	99.065
GCN	98.339	98.296	99.502	98.895
GraphSAGE	99.012	99.066	99.617	99.341
STASW	**99.241**	**99.350**	**99.636**	**99.493**

Table 17
Performance of DBSCAN and SBC under different parameters.

Eps	MinPts	Accuracy	Velocity(m/s)	Accuracy
0.0003	300	81.224	0.12	91.757
0.0002	300	79.511	0.2	96.181
0.0015	100	58.050	0.37	97.258

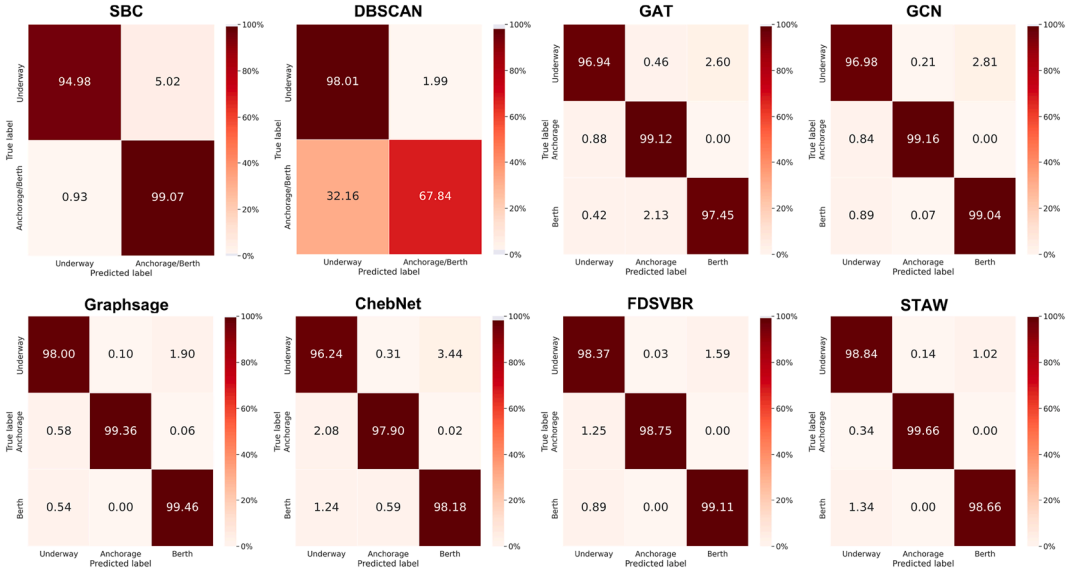


Fig. 12. Confusion matrices for vessel stop behavior recognition across different methods.

The four GNN models, GCN, GAT, GraphSAGE, and ChebNet, showed consistently strong performance. Among them, GraphSAGE achieved the highest accuracy (98.807%) within this group. However, it is important to note that all GNN models were trained using 80% of the data, which may give them an advantage in recognizing behavior patterns present in the training set. In contrast, STASW was evaluated in a data-driven unsupervised setting using only previously unseen data, which underscores its generalizability and robustness. The slightly lower accuracy observed in some GNN-based methods may be attributed to their limitations in modeling gradual behavioral transitions and long-range temporal dependencies, which are inherent in maritime trajectories. By contrast, both FDSVBR and STASW leverage localized motion patterns along with broader contextual features to more effectively capture vessel behavior. While GNN-based methods require a pre-constructed graph structure and lack real-time adaptability, FDSVBR's performance is influenced by parameter sensitivity and a constrained observation window, limiting its scalability. Overall, STASW exhibits superior flexibility and adaptability, making it particularly well-suited for real-world maritime applications.

For comparability with the neural baselines, STASW is additionally evaluated on the same 10% test split used by the GNN models. As reported in Table 16, STASW achieves the highest performance across all metrics (Accuracy 99.241%, Precision 99.350%, Recall 99.636%, F1 99.493%). Compared with the best GNN configuration (GraphSAGE; Accuracy 99.012%, F1 99.341%), STASW provides gains of +0.229 percentage points in Accuracy and +0.152 percentage points in F1, while delivering nearly identical Recall (+0.019). These results indicate that the unsupervised, parameter-adaptive STASW framework can outperform supervised GNN models trained on labeled data under a shared evaluation protocol.

Fig. 12 displays the confusion matrices for each method, illustrating the distribution of correct and incorrect classifications across the three vessel behavior classes. Notably, DBSCAN, despite its simplicity, performs well in recognizing the underway state. However, its ability to detect stop events and to distinguish between berth and anchorage is limited. For instance, the GAT model frequently misclassified berth as anchorage with a notable misclassification rate of 2.13%, and all GNN models tend to confuse underway states with berthing, particularly in the case of ChebNet. In contrast, FDSVBR and STASW demonstrate stronger capability in distinguishing between anchorage and berth, owing to their use of sequential trajectory patterns and behavior-specific thresholds. STASW, in particular, outperforms all other methods in accurately identifying both underway and anchorage behaviors, contributing to its overall superior accuracy.

Table 17 reports the sensitivity analysis of SBC and DBSCAN under various parameter configurations. While SBC is suitable for real-time applications due to its simplicity, its performance is highly dependent on the careful tuning of its speed threshold. In this study, the speed threshold was tested with three velocity thresholds (0.12 m/s, 0.2 m/s, and 0.37 m/s), where the best performance (97.258% accuracy) was achieved at 0.37 m/s. Similarly, DBSCAN achieved its optimal performance at Eps = 0.0003 and MinPts = 300, resulting in 81.224% accuracy. However, DBSCAN's reliance on spatial density and its sensitivity to hyperparameters limit its generalizability across diverse maritime environments.

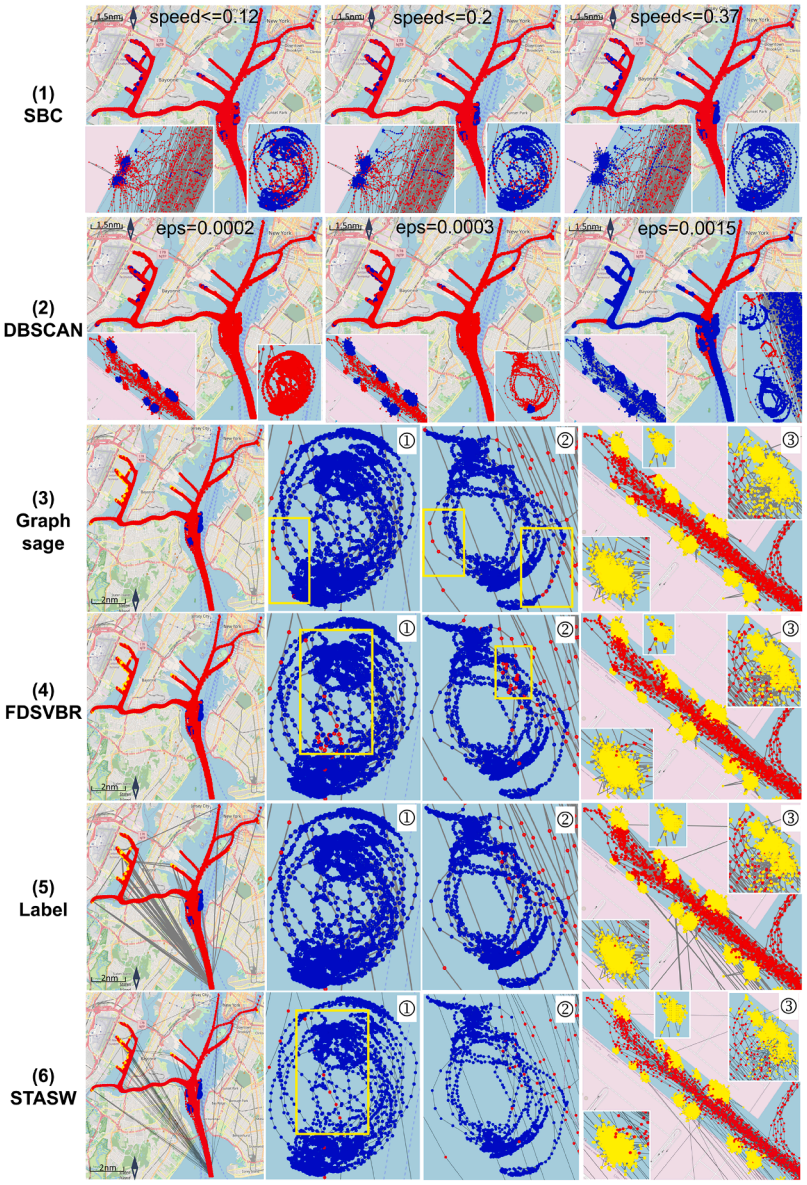


Fig. 13. Visualization of stop behavior recognition by six methods.

4.4.3. Classification visualization

Fig. 13 provides a comparative visualization of classification outcomes across different methods. DBSCAN and SBC are limited to recognizing only two states: underway (red) and stop (blue). In contrast, deep learning models, FDSVBR, and STASW are capable of simultaneously identifying berth (yellow), anchorage (blue), and underway (red) states. While both SBC and DBSCAN can deliver reasonable performance under optimal parameter settings, their rigid parameter dependencies hinder adaptability. For example, a low velocity threshold in SBC (e.g., 0.12 m/s) results in under-recognition of stops, whereas a high threshold (e.g., 0.37 m/s) leads to overestimation by misclassifying low-speed sailing as stop behavior. Similarly, DBSCAN's fixed spatial criteria make it incapable of distinguishing between anchorage and berth using a single configuration: a small Eps fails to capture loosely distributed anchorage points, while a larger value risks falsely labeling moving vessels as stationary. In contrast, FDSVBR integrates multiple parameter constraints (i.e., speed, distance, and trajectory direction) within a sliding window, improving interpretability and accuracy. For example, in (4)-①, it correctly identifies a vessel passing through the anchorage area that was otherwise mislabeled. However, its reliance on parameter tuning and its limited temporal scope reduce its scalability and effectiveness in dynamic, real-time environments.

Graph-based supervised learning models, such as GraphSAGE, demonstrated strong classification capabilities by learning from complex graph structures. However, they suffer from limitations when applied to sequential data like vessel trajectories. As shown in regions (3)-① and (3)-②, the lack of temporal context led to unstable predictions. Furthermore, these methods are highly depen-

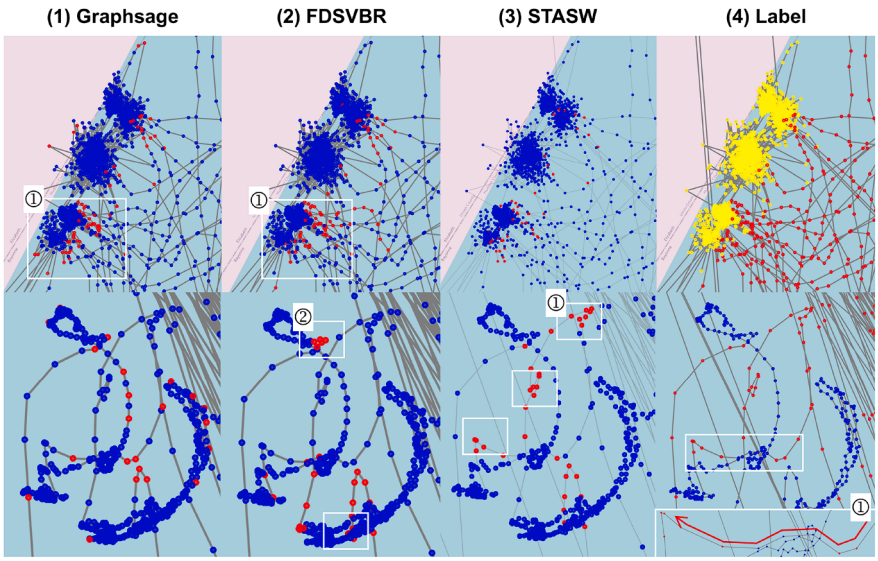


Fig. 14. Visualization of misclassification points for stop recognition.

dent on label quality: in region (5)-①, where mislabeled anchorage data were provided, the model failed to detect inconsistencies, highlighting a lack of robustness to label noise.

By contrast, STASW excels in modeling sequential transitions and does not rely on labeled data. Fig. 14 illustrates the misclassification points of GraphSAGE, FDSVBR, and STASW for the recognition of the berth and the anchorage. The rightmost column displays the ground truth labels: underway (red), anchorage (blue), and berth (yellow). Correctly classified points are shown in blue, while misclassified points are shown in red. Most misclassifications occurred in ambiguous segments during transitions between stop and underway states (e.g., in the STASW berth panel). In comparison, GraphSAGE exhibited clustering-related misjudgments, and FDSVBR occasionally misclassified underway states as stop (e.g., region (2)-①). All three methods showed varying degrees of error in anchorage recognition: GraphSAGE struggled with temporal dependencies, and FDSVBR's short observation windows introduced misclassification risk.

As illustrated in region (4)-①, when a vessel rapidly alternates between stop and underway within a confined area, STASW generally classifies the segment as underway, while FDSVBR and GraphSAGE often mislabel the entire sequence as stop. This behavior highlights the impact of window size, with smaller windows resulting in noisy predictions (e.g., region (2)-②), and larger windows failing to capture fine-grained transitions (e.g., region (3)-①). Despite these limitations, STASW's adaptive parameterization, sequential modeling capability, and independence from labeled data contribute to its robustness and practical applicability in real-world maritime behavior recognition tasks.

4.4.4. Ablation study of STASW components

Table 18 quantifies the contribution of two core STASW components: adaptive in-port/out-port parameter switching and the secondary recheck. Using only out-of-port parameters greatly increases Recall (99.727%) but causes substantial Precision loss (96.280%), producing the weakest overall Accuracy/F1 (97.703/97.973). The decline stems from permissive offshore thresholds that absorb low-speed coastal maneuvering and inflate false positives. Using only in-port parameters is more conservative yet still degrades performance (Accuracy/F1 = 98.463/98.617), as strict near-coast constraints fail to capture offshore drifting variability and reduce Recall (98.392%).

Removing the secondary recheck also harms Precision (97.461%) and lowers Accuracy/F1 to 98.188/98.389; without this boundary-pruning step, underway points around stop onsets and offsets leak into stop segments. In contrast, the full model, combining adaptive switching with the recheck, achieves the best balance, improving over the out-only variant by +1.154 Accuracy and +1.0 F1, and over the no-recheck variant by +0.669 Accuracy and +0.584 F1. These results show that adaptive regime switching is essential for handling distinct drift behaviors inside and outside ports, while the recheck module effectively suppresses boundary contamination, increasing Precision without compromising Recall.

4.5. Vessel stop functional areas

4.5.1. Vessel berth functional areas

Fig. 15 presents heatmaps of in-port stop trajectory points for three primary vessel types: towing vessels, cargo vessels, and tankers. Towing vessels, which assist with berthing and maneuvering operations, exhibit a wide spatial distribution of stop points throughout the port. This pattern reflects their flexible operational role in supporting a diverse range of vessels. While smaller tugs provide

Table 18
Ablation study of STASW components.

Variant	Accuracy	Precision	Recall	F1
STASW (full)	**98.857**	**99.072**	98.873	**98.973**
Only out-port parameters	97.703	96.280	**99.727**	97.973
Only in-port parameters	98.463	98.842	98.392	98.617
w/o secondary recheck	98.188	97.461	99.334	98.389

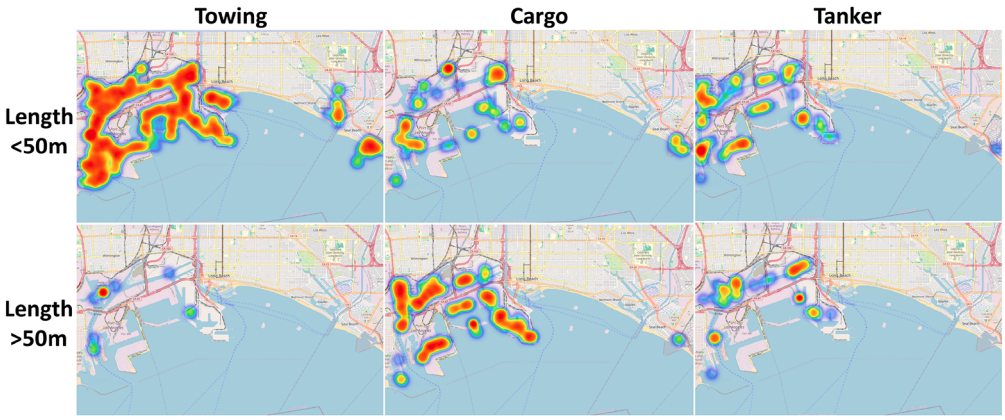


Fig. 15. Heatmaps of in-port stop trajectory points.

maneuvering support across the port, larger towing vessels concentrate their activities near turning basins and deeper waters, where high propulsion is required for assisting large vessels.

In contrast, cargo vessels show more clustered stopping behavior, with larger vessels tending to concentrate in areas equipped with container handling infrastructure, such as cranes. These locations are associated with longer dwell times due to coordinated loading and unloading operations. Tankers display a distinct spatial pattern: smaller tankers show a high density of stop points within the port, reflecting their role as service vessels engaged in bunkering and supply activities. Their relatively small size allows them to navigate efficiently throughout the port, in contrast to large tankers primarily used for oil transportation.

Fig. 16 further visualizes the geohash-based spatial distribution of all in-port stop trajectory points. The left panel illustrates the dominant vessel type within each geohash, determined by the highest proportion of trajectory points of that type within the grid. The right panel displays the distribution of berth trajectory points by vessel length. To facilitate further exploration, an interactive website has been developed (https://portminer.netlify.app/), enabling users to click on each geohash to view the proportions of different vessel types and access additional relevant data. Towing vessels are excluded from this analysis due to the widespread dispersion of their stop points across the study area. The left panel identifies the dominant vessel type in each geohash grid, based on the highest proportion of stop points. The right panel visualizes the distribution of berth points by vessel size. Analysis reveals that:

(1) Cargo vessels prefer stopping at waterway branches and deeper port zones to avoid interference with other vessels.

(2) Pleasure craft are concentrated along port flanks, minimizing interaction with commercial traffic and enhancing safety.

(3) Functional vessels such as anti-pollution and dredging vessels stop at locations aligned with their tasks, such as near port entrances for emission control or estuary zones for maintenance.

Additionally, vessel size influences berth location preference: large vessels tend to stop at secondary channels, while smaller vessels berth closer to main navigational routes, optimizing port access and task execution.

Table 19 reveals strong inter-type heterogeneity driven by heavy-tailed dwell distributions. Leisure and low-intensity classes (pleasure craft, sailing) exhibit extreme mean–median divergence (e.g., 241.76 h vs 34.17 h; 364.48 h vs 118.58 h), indicating a small set of ultra-long stays that dominate total occupancy. Commercial classes (cargo and tanker) show far tighter spread (85.62 h vs 59.61 h; 26.44 h vs 20.26 h), consistent with structured turnaround regimes, while fishing and passenger vessels combine predominantly short service calls with sporadic multi-day layovers. Segment frequency further differentiates operational roles: towing and passenger vessels generate large numbers of short events (34,161; 15,411), producing high berth churn, whereas pleasure craft accumulate the largest total berth time (≈1.61 M h) despite moderate counts, creating latent pressure on commercial quay capacity. Maximum dwell values exceeding 4,000–8,000 h for sailing, pleasure craft, towing, and passenger categories highlight concentrated long-stay risk that can distort naive utilization metrics unless separately tracked.

These temporal signatures carry direct planning implications. High-frequency short events from support vessels call for dedicated quick-turn berths, while ultra-long leisure or maintenance stays justify segregated secondary zones to preserve prime frontage for commercial rotations. Stable mean–median alignment for cargo and tanker also validates the adaptive thresholds of STASW, which correctly capture operational dwell without merging underway pauses, while avoiding spurious fragmentation in heavy-tailed leisure

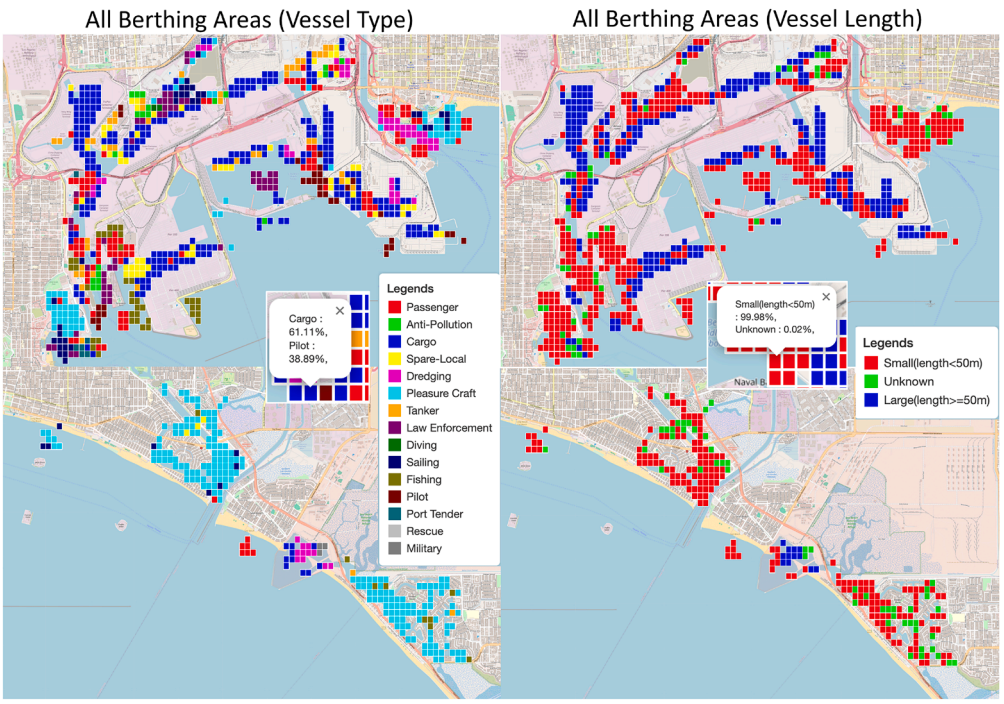


Fig. 16. Visualization of all vessel berth areas using geohash grids.

Table 19
Berth dwell statistics by vessel type (hours).

Type	Unique vessels	Segments	Points	Total (h)	Mean (h)	Median (h)	Min (h)	Max (h)
Fishing	57	1135	542,909	85662.48	75.47	5.74	0.25	4265.99
Towing	111	34,161	7,401,297	239885.86	7.02	1.50	0.25	7219.67
Sailing	169	1118	1,474,829	407483.08	364.48	118.58	0.26	8585.79
Pleasure Craft	610	6654	11966375	1608677.59	241.76	34.17	0.25	8759.77
Passenger	107	15,411	2,591,419	243003.05	15.77	1.02	0.25	4279.96
Cargo	1081	3504	5,345,553	300013.38	85.62	59.61	0.25	5827.88
Tanker	303	1433	652,489	37889.07	26.44	20.26	0.26	332.56

categories. Overall, Table 19 quantifies the multi-modal nature of berth usage and provides actionable guidance for differentiated zoning, anomaly screening of prolonged occupancy, and vessel-type-specific forecasting of berth demand.

4.5.2. Vessel anchorage functional areas

Fig. 17 presents heatmaps of anchorage trajectory points for the same three vessel types. For towing vessels, small tugs operate both within and beyond port boundaries, whereas large tugs are more concentrated inside the port due to their role in assisting large vessels under higher navigational risk.

For cargo vessels, small vessels typically anchor sparsely within the port, often waiting for docking opportunities. In contrast, large cargo vessels exhibit denser anchorage patterns, reflecting port congestion and insufficient berth availability. Tankers display fewer anchorage points outside the port compared to cargo vessels. This is because some tankers operate as in-port service vessels that can depart promptly after task completion. Notably, small tankers are observed anchoring exclusively within the port, consistent with their operational roles in bunkering and support services.

To ensure high-quality anchorage areas extraction, only trajectory points from vessels longer than 50 m were used. The left panel of Fig. 18 shows their spatial distribution using geohash grids. Outlier regions, where vessels anchor temporarily to avoid congestion, are identified and removed based on frequency and connectivity criteria. The middle panel presents the final delineated anchorage areas, segmented into six spatially connected subregions. A finer geohash level (level 7) is used within the port to capture local detail, while a coarser level (level 6) is adopted outside the port to enhance computational efficiency.

Comparison with official nautical charts demonstrates the successful identification of all designated anchorage areas, demonstrating the validity of the proposed method. Importantly, additional anchorage areas not shown on the charts (highlighted in red) are also detected. These previously unmarked zones likely represent overflow anchorage areas used during peak congestion or scheduling mismatches, when vessels are forced to wait outside the port before docking. This new finding underscores the practical value of PortMiner in revealing evolving operational patterns and aiding port management.

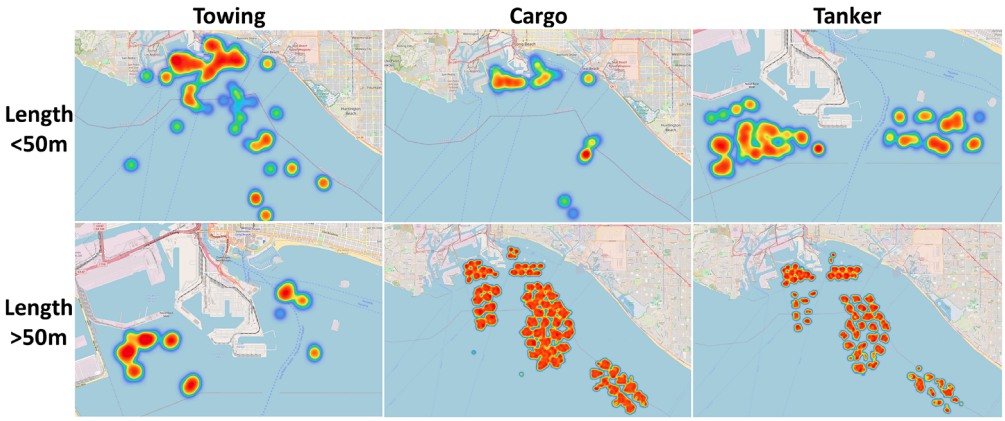


Fig. 17. Heatmaps of anchorage trajectory points.

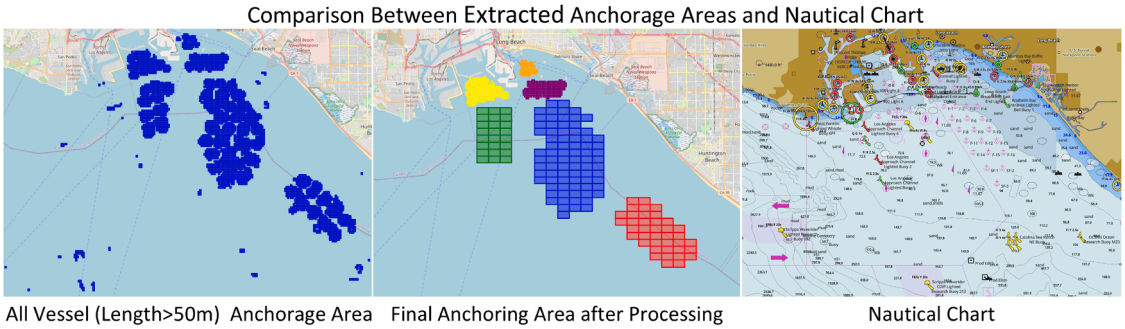


Fig. 18. Visualization of all vessel anchorage areas.

Table 20 indicates a clear stratification of anchorage behavior driven jointly by distance from operational centers and coupling to approach flows. Offshore zones accumulate substantially larger total dwell, out-port-1 and out-port-2 together exceed 237,000 h, reflecting their role as primary buffers that absorb arrival surges and stabilize sequencing before vessels commit to in-port maneuvering. Their dwell distributions show moderate skew (mean/median ratios ≈1.2–1.3), consistent with structurally extended yet predictable waiting envelopes. In-port areas display sharper asymmetry (ratios >2), mixing short staging pauses with occasional service-related delays. Extreme dwell values reinforce this gradient: the most distant offshore field hosts week-scale stays (>1500 h), whereas in-port maxima remain markedly lower (<750 h), preserving berth accessibility. Segment and vessel counts further differentiate functions: the high-throughput offshore buffer exhibits continuous rotational intake (over 1300 segments), whereas the compact inner in-port area shows few vessels but elevated mean dwell, indicating a capacity-limited or service-specialized niche.

These statistical patterns provide operational implications. The sustained offshore throughput suggests prioritizing predictive queue monitoring and dynamic capacity signaling in the large buffer zone, while exceptionally long offshore dwell events warrant anomaly screening to distinguish lay-up from congestion. In-port regions require tighter micro-management: frequent short stays near maneuvering corridors demand rapid-turnover scheduling, whereas the small, high-mean-dwell pocket should be safeguarded for specialized maintenance or provisioning tasks. The moderate offshore mean–median compression validates the robustness of the adaptive speed and displacement thresholds used for anchorage detection, while the higher in-port skew underscores the need for regime-aware switching to avoid misclassifying brief transitional maneuvers.

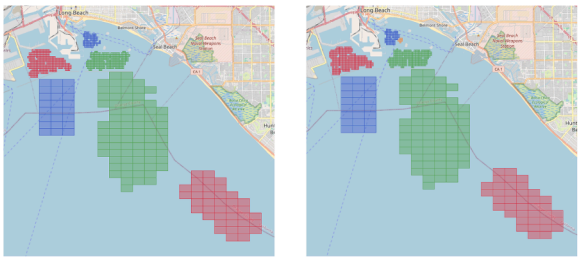
Overall, the dwell statistics reveal a multi-tier anchorage system, where distance and approach-flow integration jointly shape dwell magnitude, variability, and functional role, consistent with the hierarchy inferred from the geohash-based extraction. By capturing these unsupervised dwell patterns, PortMiner provides quantitative inputs that can inform metaheuristic diagnostic models used in recent studies for optimizing demurrage and congestion (Kweon et al., 2022; Yoon et al., 2025; Yun et al., 2025).

The diurnal maps (Fig. 19) are nearly identical. Offshore (out-of-port) anchorages show no visible change in footprint or connectivity. In-port anchorages present only slight boundary deviations: at night, lower in-port mooring density causes a few edge cells to drop below the connectivity/frequency threshold, yielding one small, newly isolated micro-region. Crucially, the centroids and main extents of all anchorage patches remain stable, indicating that the method is insensitive to diurnal sampling variability.

In contrast to the diurnal case, seasonal variation (Fig. 20) is more pronounced. Winter exhibits the densest and most expansive anchorage footprint, consistent with longer dwell caused by adverse metocean conditions (storms/swell) and holiday-period operational slowdowns and arrival bunching. Spring, summer, and autumn all show varying degrees of spatial contraction: improved weather and higher terminal throughput shorten waiting times, shrinking in-port anchorage envelopes; offshore patches also contract coherently

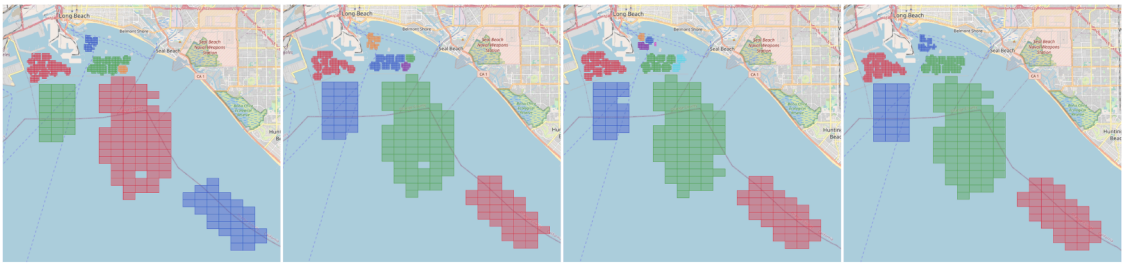
Table 20
Anchorage dwell statistics per region (hours). Colors correspond to the final anchorage sub-regions in Fig. 18: in-port (yellow, purple, orange) and out-port (red, blue, green).

Region	ID	Color	Vessels	Segments	Points	Total (h)	Mean (h)	Median (h)	Min (h)	Max (h)
in-port	0	yellow	687	1437	730,930	38034.81	26.47	11.30	1.25	483.82
in-port	1	purple	431	1106	680,672	36389.08	32.90	14.28	1.17	729.00
in-port	2	orange	43	113	152,583	8535.13	75.53	34.65	1.65	505.99
out-port	0	red	200	321	834,050	44696.07	139.24	111.60	1.60	1535.96
out-port	1	blue	635	1362	3,334,098	178381.93	130.97	96.78	1.12	1044.45
out-port	2	green	320	502	1,095,091	59466.00	118.46	89.20	1.23	591.05



(a) (b)

Fig. 19. Anchorage areas extraction results during (a) daytime and (b) nighttime.



(a) (b) (c) (d)

Fig. 20. Anchorage areas extraction results during different seasons: (a) spring, (b) summer, (c) autumn, and (d) winter.

but retain their topology. Overall, the seasonal results suggest demand-driven expansion/contraction rather than instability of the algorithm, major subregions persist across seasons while their boundaries adapt smoothly to traffic intensity.

4.6. Vessel inbound and outbound port channels

To extract the primary navigational pathways within the port, geohash grids exhibiting substantial imbalances between inbound and outbound trajectory point densities were identified. This analysis resulted in the detection of four key regions (the blue grids in the bottom panel): two primarily associated with inbound movements and two with outbound. By linking these regions with stop trajectory points, four corresponding heatmaps were constructed to visualize inbound and outbound vessel trajectories, as shown in the upper panel of Fig. 21.

In these visualizations, red grids represent geohash cells with trajectory densities below the global mean, while white grids denote isolated cells not connected to the main navigational channels. The resulting comparison reveals an asymmetrical structure: a single dominant inbound route is observed, whereas outbound traffic is distributed along two distinct branches. This pattern reflects operational characteristics of port entry and departure. As discussed in Section 4.5, vessels, particularly tankers, frequently anchor before berthing, often in designated in-port anchorage zones. The right-hand entrance channel, located closer to these zones, exhibits a higher concentration of inbound traffic, especially for vessels engaged in bunkering operations. In contrast, outbound vessels tend to depart directly, contributing to the formation of two separate outbound routes. This behavioral asymmetry is quantitatively supported by the large-vessel direct flow ratios reported in Table 21: only 41.83% of arrivals berth without an anchorage/waiting phase (indicating prevalent pre-berth staging), whereas 92.67% of departures exit the port without intermediate stopping, explaining the bifurcated, rapidly dispersing outbound branches versus the single, staging-weighted inbound approach. These patterns highlight operational distinctions between inbound and outbound flows and demonstrate the utility of trajectory-based spatial analysis in revealing real-world vessel movement dynamics.

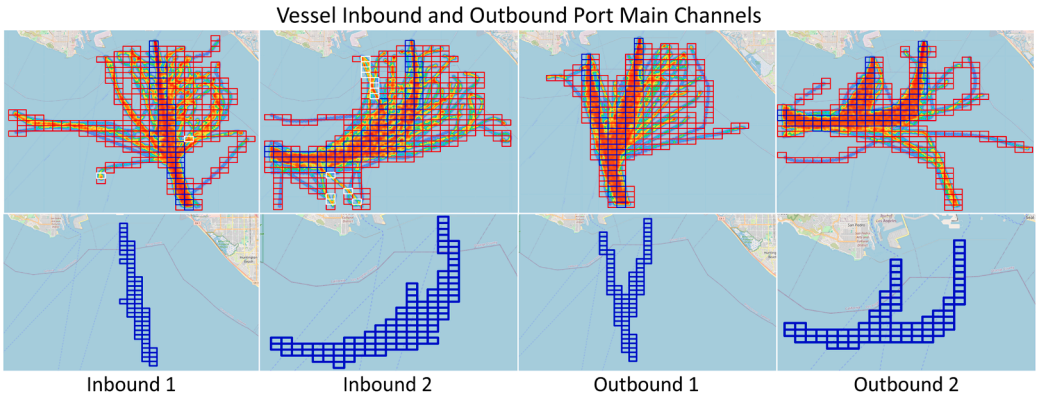


Fig. 21. Inbound and outbound port channels extracted from vessel trajectory data.

Table 21
Direct berth arrival and direct departure ratios for large vessels (length > 50 m). Small vessels excluded to avoid bias from frequent immediate in/out movements.

Metric	Ratio	Definition
Direct berth arrival	0.4183	Fraction of arrivals berthing without anchorage / waiting
Direct departure	0.9267	Fraction of departures leaving port without intermediate stop / waiting

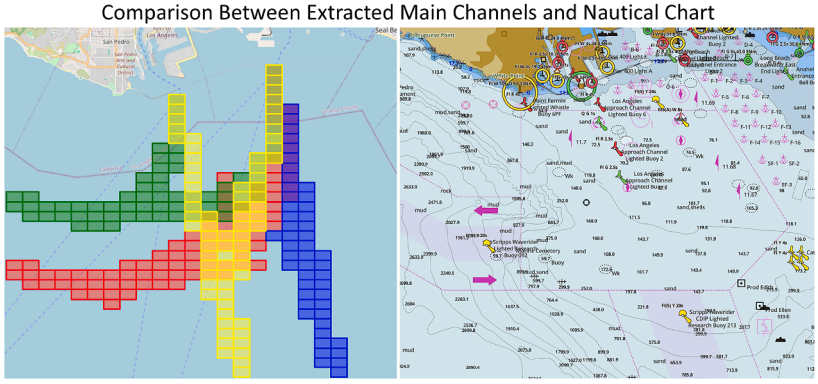


Fig. 22. Comparison between the extracted main inbound and outbound channels and nautical charts.

Fig. 22 presents a comparison analysis between the extracted inbound and outbound channels and official nautical charts of the Port of Los Angeles. The AIS data used in this study covers a broader area than the port boundaries defined in the charts. The left-side entrance channels on the chart show a downward inclination, which is accurately captured by the green and red paths in the left panel, which show a consistent downward-then-horizontal progression. Likewise, the yellow and blue outbound trajectories align with the upward-sloping trends observed in the chart.

While the nautical chart defines the entrance and exit channels, it lacks detail on internal routing patterns. The data-driven extraction approach applied here reveals key in-port navigational pathways extending from the entrance channels to operational areas. Moreover, the internal channels identified near the two primary port entrances correspond closely with the positions of navigation buoys shown on the chart, providing further validation of the approach.

Overall, these results demonstrate that the proposed method not only complements traditional chart-based information but also enhances maritime situational awareness by uncovering internal routing structures not previously documented. This added insight holds significant value for vessel traffic management, operational planning, and navigational safety enhancement.

The diurnal differentiation shown in Fig. 23 is pronounced rather than marginal. In Inbound 1, daytime traffic concentrates on the right-hand corridor because vessels arriving during peak service hours often detour to the adjacent in-port anchorage cluster for queuing, bunkering, pilot boarding, or minor adjustments (Section 4.5.2). At night, inbound flows become bilateral as reduced anchorage buffering, more synchronized pilotage windows, and diminished interference from small service and pleasure craft enable direct progression via the left corridor. Outbound 1 displays a similar contrast: daytime traffic favors the right-hand branch due to departures near bunkering and support facilities, while nighttime flows shift left as batch departures occur after service queues diminish. Outbound 2 retains a persistent right-side dominance across the diurnal cycle, consistent with its alignment with deeper-water

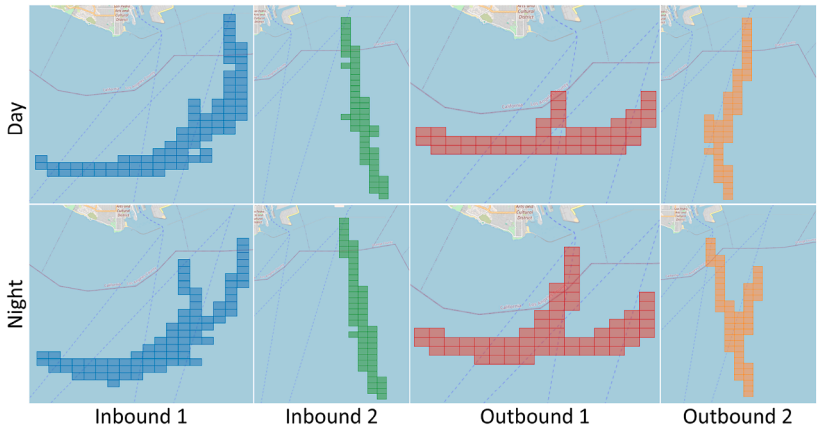


Fig. 23. Inbound and outbound port channels extraction results during daytime and nighttime.

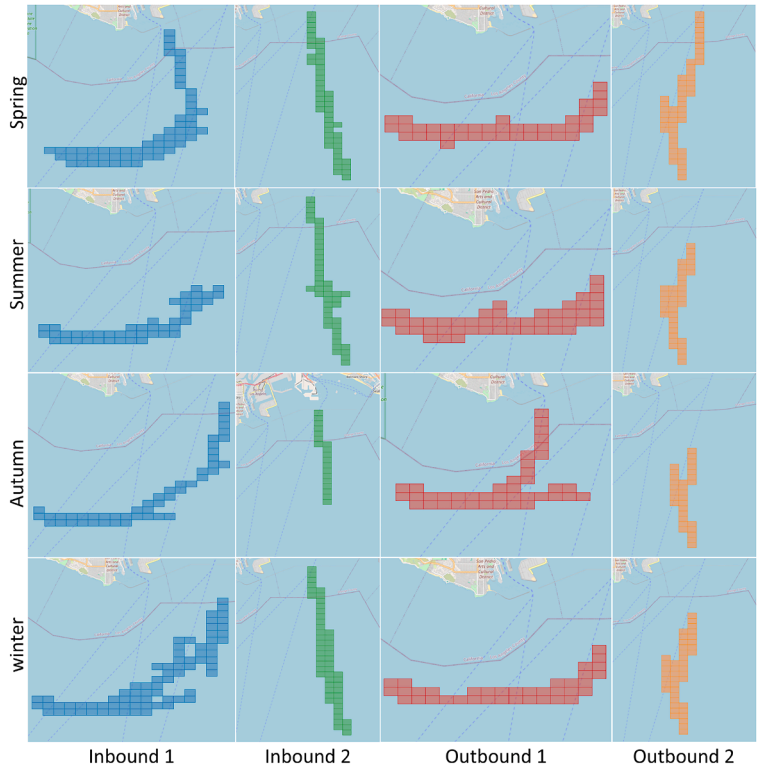


Fig. 24. Inbound and outbound port channels extraction results during different seasons.

outer lanes and stable navigational aids. These directional asymmetries reflect coherent operational regimes rather than stochastic density fluctuations.

Seasonal variation, illustrated in Fig. 24, exhibits an even stronger influence on channel utilisation, especially in Inbound 1. In spring, traffic shows increased use of the left-hand branch, a pattern plausibly associated with prevailing wind conditions that stabilise left-side approaches, coupled with reduced anchorage congestion following winter backlog clearance. In contrast, autumn and winter display a reversed configuration: right-hand inbound flow intensifies, and several fine-scale lateral branches emerge toward both overflow and primary anchorage areas (Fig. 18). These shifts are consistent with pre-holiday cargo aggregation, weather-induced speed reductions that increase holding requirements, and precautionary routing along deeper and better-marked water. Summer exhibits the most balanced bidirectional utilisation. Higher overall throughput and efficient berth turnover distribute flows across both branches, while density filtering suppresses minor lateral deviations, several low-density cells fall below the global mean threshold

Table 22
Comparison of different functional area extraction methods based on their real-time performance.

Method	Description	Time (s)
Quadrilateral (Rectangle)	Polygon contains (4 vertices)	0.0751
Pentagon	Polygon contains (5 vertices)	0.0878
Hexagon	Polygon contains (6 vertices)	0.0860
Geohash level 7 (lat/lon)	Encode lat/lon and set lookup	0.0628
Geohash level 7 (encoded)	Set lookup with pre-encoded geohash7	0.0016
Disconnected Rectangles	Polygon contains 2 rectangles	0.2118
Geohash level 7 (2 quads)	Encode lat/lon and set lookup for 2 rectangles	0.0606

and are therefore excluded. Outbound 2 shows a markedly different behavior, maintaining consistently higher right-side utilisation across all seasons. This stability reflects its function as a deep-water exit channel with minimal seasonal reconfiguration.

Across inbound and outbound channels, the spatio-temporal variability observed in Figs. 23 and 24 reflects the combined effects of operational and environmental drivers. Day-night differences are shaped by anchorage occupancy cycles (daytime staging, nighttime release) and by increased interference from service and small craft during daylight hours. Seasonal shifts arise from autumn–winter expansions of anchorage demand and from metocean conditions that alter safety margins and influence preferred routing patterns. Batch-arrival scheduling contributes to nighttime bilateral clearance and to pre-holiday right-side surges. Despite these variations, the core axial lanes remain persistent across all temporal partitions, indicating that the extracted structures capture authentic traffic-regime dynamics rather than artifacts introduced by spatial resolution, density thresholds, or extraction procedures.

4.7. Real-time efficiency comparison of functional area extraction methods

To evaluate the operational practicality of the proposed Geohash-based approach, a comparison was conducted against the traditional polygon-based method in terms of real-time query efficiency. Table 22 lists the time required to determine whether 10,000 trajectory points fall within predefined functional areas using both approaches.

For polygon-based evaluation, the Polygon.contains() function from the optimized Python Shapely library was employed. Various polygon types were tested, including rectangles, pentagons, hexagons, and configurations involving multiple disconnected regions. For the Geohash-based approach, spatial regions were encoded using level 7 Geohash, consistent with the resolution applied throughout this study.

To ensure comparability, identical spatial regions were represented using both polygon and Geohash formats. A single rectangular area corresponds to 21,462 level 7 Geohash cells, while two disconnected rectangles translate to 465 cells.

The results are summarized as follows:

(1) The polygon-based rectangle query required 0.0751 s, while the Geohash-based method using latitude/longitude encoding took 0.0628 s, yielding a 16.38% improvement.

(2) When trajectory points were pre-encoded as Geohash, the query time was reduced to 0.0016 s, representing a 39.25× speedup.

(3) Among polygon types, hexagons performed slightly better than pentagons, likely due to more efficient point-in-polygon algorithms optimized for regular, convex shapes.

(4) For disconnected spatial regions, the polygon-based method required separate evaluations for each region, resulting in a computation time of 0.2118 s, whereas the Geohash-based method maintained consistent performance (0.0606 s for two regions).

These results underscore the computational efficiency and scalability of the Geohash-based method, particularly in large-scale and real-time spatial query scenarios. The approach is well-suited for applications such as vessel behavior monitoring, automated functional area classification, and port traffic analysis, where both speed and accuracy are critical.

4.8. Discussions and implications

The proposed PortMiner framework provides a comprehensive and unsupervised approach to extracting and analyzing port functional areas, offering substantial improvements in operational awareness, decision-making, and digital transformation within port environments. The findings of this study reveal significant implications for various stakeholders across the maritime industry, including port authorities, shipping operators, and technology developers, as follows:

(1) Implications for port authorities.

Port authorities stand to benefit considerably from the implementation of the PortMiner framework. By providing automated, real-time, and high-resolution identification of berths, anchorages, and navigational channels, the system enhances situational awareness and strengthens monitoring capabilities. The ability to dynamically detect stop points and vessel routing patterns enables port managers to identify boundary violations, unauthorized mooring, and operational inefficiencies with minimal human intervention.

Furthermore, the framework facilitates data-driven berth allocation planning by identifying dominant vessel types and usage patterns in each spatial region. This supports more equitable and efficient resource distribution, reduces bottlenecks, and improves

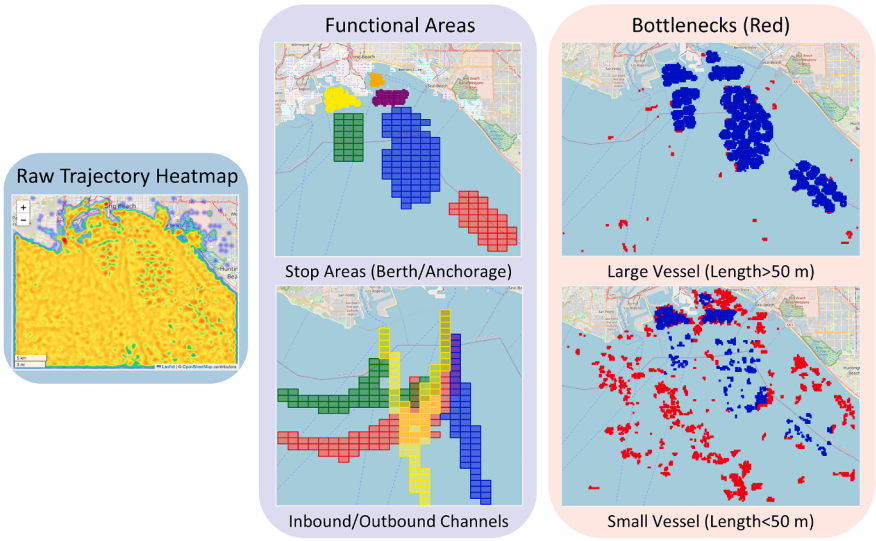


Fig. 25. Diagnostic illustration: raw AIS trajectories versus PortMiner-derived functional structures, including berths, anchorages, channels, and congestion zones.

port throughput. Over time, the insights generated by PortMiner can inform infrastructure upgrades, capacity planning, and policy development, contributing to the long-term resilience and sustainability of port operations.

(2) Implications for shipping operators.

Shipping companies can leverage the behavioral insights generated by PortMiner to optimize fleet scheduling, arrival planning, and resource allocation. By understanding the spatial-temporal distribution of vessel stop behavior and channel utilization, operators can identify congestion-prone areas, anticipate delays, and adjust routing strategies accordingly.

In particular, the detection of anchorage usage patterns and waiting times allows for more accurate turnaround time estimation and improved just-in-time arrival coordination. This capability reduces unnecessary fuel consumption, minimizes idle times, and enhances service reliability. Ultimately, the integration of behavior-aware decision support enabled by PortMiner contributes to operational cost reduction and enhanced service competitiveness.

(3) Implications for technology developers and solution providers.

From a technological standpoint, PortMiner introduces several innovations that can be adopted and extended by developers of maritime analytics platforms. The unsupervised STASW algorithm for stop recognition and the use of geohash-based spatial indexing offer a scalable and transferable alternative to traditional rule-based and supervised learning methods. These components reduce dependency on labeled data and manual parameter tuning, making the framework more adaptable to diverse port environments and dynamic traffic conditions.

Moreover, the framework's ability to generalize across ports with varying geographies and vessel types significantly lowers deployment complexity for intelligent port management systems. Solution providers can integrate these methods into existing port digitalization platforms or develop standalone applications for real-time monitoring, predictive analytics, and port simulation.

(4) Implications for managerial diagnostic.

Beyond the spatial extraction of port functional areas, PortMiner provides diagnostic insights into operational efficiency. By linking the extracted anchorage areas and channels to port-level KPIs, such as anchorage dwell time (reflecting waiting efficiency), berth occupancy profiles (represented by average berth-level dwelling duration), and inbound/outbound asymmetry (the proportion of vessels that berth immediately upon arrival versus those that exit directly), the framework supports data-driven assessment of port performance.

Fig. 25 illustrates the managerial value of this capability. Prior to processing, raw AIS trajectories form a dense, visually cluttered point cloud that obscures spatial bottlenecks and operational hierarchies. Following PortMiner's processing, geohash-based connectivity clustering and stop-behavior segmentation reveal (i) congestion hotspots in non-stop transit corridors, especially near berth frontages and at the mouths of inbound and outbound channels where large-vessel flows converge, and (ii) asymmetric maneuvering zones where small craft repeatedly pause or yield near turning basins. These extracted structures can be directly mapped to operational KPIs, anchorage dwell (queue dynamics), berth occupancy (turnaround efficiency), and inbound/outbound asymmetry (routing balance), thereby enabling targeted interventions in berth scheduling, dynamic anchorage allocation, and navigational safety management.

Overall, the PortMiner framework contributes to the broader goal of enhancing automation, efficiency, and intelligence in port operations. Its systematic, data-driven approach aligns with the global movement toward smart port ecosystems and supports the digital transformation of the maritime industry. By enabling real-time functional area recognition and behavior-based decision support, PortMiner promotes safer navigation, more informed management practices, and greater operational agility.

5. Conclusions

This study presents PortMiner, a novel unsupervised and systematic framework for functional area mining in port waters. Designed to address key limitations in existing approaches, such as heavy reliance on labeled data, the need for manual parameter tuning, and inefficiencies in traditional polygon- or network-based spatial methods, PortMiner introduces a scalable, data-driven solution for vessel behavior analysis and spatial pattern recognition in complex port environments. Central to the framework is the STASW algorithm, which enables adaptive and annotation-free recognition of vessel stop behaviors by jointly analyzing spatial and temporal dimensions. The framework further integrates a geohash-based spatial representation with connectivity-based clustering, enabling efficient multi-level indexing and the robust delineation of berths, anchorages, and navigational channels. Extensive experiments on benchmark datasets and real AIS data from the Port of Los Angeles validate the effectiveness of PortMiner. The framework achieves new state-of-the-art accuracy in stop behavior recognition while significantly reducing the need for manual intervention. The extracted functional areas align closely with official nautical charts and, notably, uncover additional high-traffic regions that the existing studies and charts fail to document. These findings underscore the potential of PortMiner to enhance situational awareness, support intelligent port management, and contribute to improved maritime safety. A publicly available interactive platform has been developed to facilitate further exploration and practical use of the results.

Although PortMiner is designed as an unsupervised framework, certain limitations warrant attention. First, its parameter derivation relies on statistically informed heuristic rules; while this eliminates the need for manual tuning, the approach is better characterized as data-driven heuristic adaptation rather than strictly parameter-free modeling. This design achieves a balance between generalization and empirical robustness but introduces dependence on the statistical properties of the underlying dataset. Second, although the framework demonstrates strong performance in large container and mixed-use ports, its transferability to geographically complex or operationally atypical settings, such as estuary ports with strong tidal currents or bulk terminals with highly irregular loading and anchoring patterns, remains to be fully validated. In such environments, extreme current-induced drift or unconventional maneuvering behavior may exceed standard adaptive thresholds, potentially requiring the incorporation of hydrodynamic features or localized calibration to reliably distinguish operational stops from environmentally driven motion.

Future work could extend the current framework in two directions. First, analysis of correlations between extracted functional regions and contextual features, such as seasonal variations in waiting times or entry behavior by vessel type, could provide deeper operational insights. Second, incorporating models of multi-vessel interaction, including congestion dynamics and collision avoidance behavior, may further broaden the applicability of PortMiner to support risk-aware maritime operations.

CRediT authorship contribution statement

Huimin Qiang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Wenlong Niu:** Writing – review & editing, Visualization, Validation, Formal analysis, Investigation; **Xiaodong Peng:** Writing – review & editing, Visualization, Validation, Supervision, Formal analysis, Investigation, Methodology; **Huanhuan Li:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization; **Zaili Yang:** Writing – review & editing, Visualization, Validation, Project administration, Funding acquisition, Formal analysis.

Data availability

The datasets used in this paper are publicly available from the NOAA Coast website (https://coast.noaa.gov/htdata/CMSP/AISDataHandler/2021/) and from a publicly accessible GitHub repository (https://github.com/destiny1103/DT-GNN).

Declaration of competing interest

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

Acknowledgement

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 864724).

Appendix A. Global parameter summary and STASW threshold derivation

Table A.1 consolidates all key parameters utilized in the PortMiner framework, including their selected values or statistical derivation rules, along with justifications for their choices and roles within the overall system.

Table A.1
Consolidated parameters used in PortMiner, including selected values or statistical derivation rules and their justification.

Parameter	Value / Derivation Rule	Justification / Role in Framework
Coastline geohash level	Level 6	Provides stable coastline representation and enables reliable adaptive parameter switching while maintaining computational efficiency.
Berth classification geohash	Level 7	Higher spatial resolution is required to distinguish berth locations from adjacent fairways.
Stop-area delineation geohash	In-port: Level 7; Out-port: Level 6	In-port stays require fine granularity to avoid mistakenly covering on-land areas; out-port stays benefit from coarser cells to reduce noise and fragmentation.
Channel extraction geohash	Level 6	Preserves continuous channel morphology with low fragmentation; Level 7 is overly fragmented while Level 5 is too coarse.
Inbound/outbound time threshold ϵ_t	1800 s	Based on AIS gap statistics (cargo vessels >50 m have a mean gap of 33.97 min), capturing meaningful interruptions while filtering noise.
Adaptive STASW thresholds (statistical derivation)		
Window ω	In-port(ω_{in}): 5; Out-port(ω_{out}): 15	Short windows capture compact berth stays; longer windows stabilize stay detection under larger drifting variance at anchorages.
Speed threshold ϵ_v	In-port ($\epsilon_v^{\text{in}} = 0.12$ m/s): Mean speed of points inside Level 7 coastline cells; Out-port($\epsilon_v^{\text{out}} = 0.37$ m/s): $3 \times \epsilon_v^{\text{in}}$	Ships near the coastline (Level 7) are almost always berthing, giving stable in-port speed estimates; out-port mean speed reflects drifting conditions.
Course variability ϵ_θ	In-port ($\epsilon_\theta^{\text{in}} = 16°$): P10 of $\lvert\Delta\theta\rvert$; Out-port($\epsilon_\theta^{\text{out}}$): $3 \times \epsilon_\theta^{\text{in}}$	In-port heading changes (coastline, Level 7) are minimal; the 10th percentile gives a robust lower bound. Out-port threshold is doubled to accommodate wind- and wave-induced drift.
Displacement threshold ϵ_δ	In-port($\epsilon_\delta^{\text{in}} = 135$ m): P99 of in-port (coastline, Level 7) $d_{i,i+1}$; Out-port($\epsilon_\delta^{\text{out}} = 551$ m): P99 of out-port	the 99th percentile to approximate the upper bound of normal short-step movements, filtering out anomalous jumps while retaining the full range of legitimate drifting behavior.
Adaptive switching criterion	Switch to in-port if >0.5 of Level 6 coastal cells appear in the outer window	A majority-based decision prevents oscillatory switching, and the Level 6 coastline encoding provides stable in-port/out-port regime transitions.
Secondary recheck (Type 1)	Sliding-window + threshold combination	Refines segment boundaries by jointly re-evaluating speed, heading, and displacement; suitable for high-quality AIS data where fine-grained boundary adjustment is reliable.
Secondary recheck (Type 2)	Density filtering (DBSCAN-like), radius = mean pairwise distance, MinPts = $\omega/2$	Removes low-density edge points and enhances robustness under noisy AIS observations, particularly when outliers or sparse segments occur.

Appendix B. Stop point recheck algorithm

As illustrated in Fig. B.1, when no secondary check is performed, trajectory segments surrounding vessel stops may include underway points misclassified as stop points, particularly those immediately preceding or following actual stop behaviors. To address this issue, two recheck methods are proposed, offering flexibility based on data quality and experimental conditions.

The first method, detailed in Algorithm B.1, applies an adaptive thresholds-based sliding window technique that iteratively corrects misclassified points from both the start and end of a candidate stop segment. This approach is particularly suitable for high-quality AIS data where adaptive thresholds can be reliably determined. It is most effective in contexts where vessels do not exhibit additional non-operational stops outside the port (e.g., yielding or fishing), and thus is adopted for baseline comparisons on clean benchmark datasets.

In Algorithm B.1, the process begins by classifying the trajectory segment's stop type, berth, anchorage, or non-stop, based on the segment's spatial location, using level 6 and level 7 geohash grids (Steps 1–12). Segments falling within geohash grids adjacent to the port coastline are classified as berths; otherwise, they are designated as anchorage areas. The algorithm then checks boundary points of each segment, removing those with speeds exceeding predefined thresholds to eliminate misclassified underway points (Steps 13–22).

Subsequently, a point-by-point verification is conducted using a sliding window (Steps 23–29). For each stop type, the corresponding window size and speed threshold are applied. During verification (via the Point-by-Point Check function), the method evaluates each point using features such as speed, course change, and the proportion of compliant points within the segment (Steps 30–44). For berths, points are flagged if their speed exceeds the defined speed threshold and course change is below the in-port threshold, or if their speed exceeds 75% of the threshold and the course change is below the out-port threshold (Step 34). For anchorages, points are flagged if the speed exceeds 50% of the threshold and the course change is below the in-port threshold (Step 35). If more than 50% of the segment's points satisfy the underway conditions, the algorithm labels the entire subsegment up to the last qualifying point as non-stop (Steps 37–40). The window then shifts forward, repeating the process until a subsegment no longer satisfies the underway condition, signifying the resumption of valid stop behavior. This process is conducted from both the start and the end of the segment (Steps 25 and 28).

Algorithm B.1 Stop point recheck.

Input: SEG, a segment of trajectory points with candidate stop labels; ω, sliding window size (with ω.in and ω.out for in-port and out-port); ϵ_v, speed threshold (with in-port and out-port values); ϵ_θ, course threshold (with in-port and out-port values).
Output: SEG, the segment with refined stop point states (berth, anchorage, or non-stop) for each trajectory point.

Reset SEG index to consecutive integers
if any point in SEG is located within a level 6 geohash corresponding to the coastline **then**
  if any point in SEG is located within a level 7 geohash corresponding to the coastline **then**
    $stop\_label \leftarrow 1$ ▷ Berth
    $\epsilon_v.tem \leftarrow \epsilon_v$.in
  else
    $stop\_label \leftarrow 0$ ▷ Not a stop segment
  end if
else
  $stop\_label \leftarrow 2$ ▷ Anchorage
  $\epsilon_v.tem \leftarrow \epsilon_v$.out
end if
SEG.$state \leftarrow stop\_label$ ▷ Set initial state
$start, end \leftarrow 0, \text{len(SEG)} - 1$
while $start < end$ and $\text{SEG}_{start}.v > 1.5\epsilon_v.tem$ **do**
  $\text{SEG}_{start}.state \leftarrow 0$ ▷ Not a stop point
  $start \leftarrow start + 1$
end while
while $end > start$ and $\text{SEG}_{end}.v > 1.5\epsilon_v.tem$ **do**
  $\text{SEG}_{end}.state \leftarrow 0$ ▷ Not a stop point
  $end \leftarrow end - 1$
end while
if $stop\_label == 1$ **then**
  POINT-BY-POINTCHECK(SEG, start, ω.in, ϵ_v.in, ϵ_θ.in, ϵ_θ.out) ▷ Check from the beginning
  POINT-BY-POINTCHECK(Reverse(SEG), len(SEG)-1-end, ω.in, ϵ_v.in, ϵ_θ.in, ϵ_θ.out) ▷ Check from the end
else if $stop\_label == 2$ **then**
  POINT-BY-POINTCHECK(SEG, start, ω.out/2, ϵ_v.out, ϵ_θ.in, ϵ_θ.out)
  POINT-BY-POINTCHECK(Reverse(SEG), len(SEG)-1-end, ω.out/2, ϵ_v.out, ϵ_θ.in, ϵ_θ.out)
end if
function POINT-BY-POINTCHECK(SEG, start, ω, ϵ_v, $\epsilon_\theta.in$, $\epsilon_\theta.out$)
  $P \leftarrow \text{SEG}_{start:start+\omega}$
  $break\_flag \leftarrow 1$
  while $break\_flag$ **do**
    $cond_{berth} \leftarrow (P.v > \epsilon_v$ **and** $|P.\theta| < \epsilon_\theta$.in) **or** ($P.v > 0.75\,\epsilon_v$ **and** $|P.\theta| < \epsilon_\theta$.out)
    $cond_{anchorage} \leftarrow P.v > 0.5\,\epsilon_v$ **and** $|P.\theta| < \epsilon_\theta$.in
    $ind \leftarrow$ the index list of trajectory points satisfying $cond$ ▷ Select $cond$ according to the stop type
    if $\text{len}(ind) > \omega/2$ **then**
      $P_{start:ind[-1]}.state \leftarrow 0$ ▷ Not a stop point
      $start \leftarrow ind[-1] + 1$
      $P \leftarrow \text{SEG}_{start:start+\omega}$
    else
      $break\_flag \leftarrow 0$
    end if
  end while
end function

The second method, shown in Fig. B.2, is inspired by the DBSCAN clustering algorithm. It determines whether to reclassify points based on local density calculations. This method is more robust for lower-quality AIS data and complex operational environments involving diverse vessel types and behaviors (e.g., fishing, dredging, or yielding). In this study, the DBSCAN-inspired approach is applied to real-world AIS data from the Port of Los Angeles to enhance classification reliability.

In this method, the neighborhood radius is set as the mean pairwise distance among stop points within a segment, and the minimum density threshold is defined as half the sliding window size. The window size is dynamically adjusted according to the spatial location of trajectory points. Further classification is performed based on spatial context and stop duration. If over 95% of points in a segment fall within coastline-level geohash grids and the stop duration exceeds 15 mins, the segment is classified as berth. If fewer than 50% of the points are near the coastline and the stop duration exceeds 1 h, the stop is classified as anchorage. Segments not meeting either

condition are designated as other stop behaviors. Whether a point is considered near the coastline is determined by its presence in a level 6 geohash cell corresponding to the coastline boundary.

These two recheck methods enhance the reliability and accuracy of stop behavior recognition across diverse port scenarios and data qualities, ensuring more robust downstream functional area analysis.

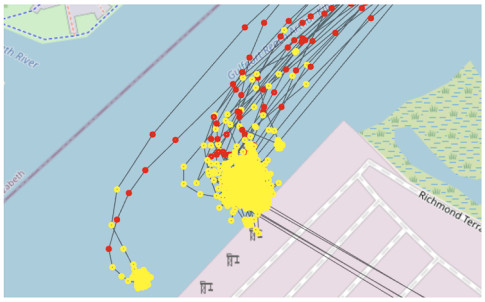


Fig. B.1. Experimental results of stop point recognition without secondary recheck. Red points represent underway trajectories, while yellow points indicate stop points identified by the method. Without rechecking, vessel trajectories immediately before stopping and after leaving are often misclassified as stop points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

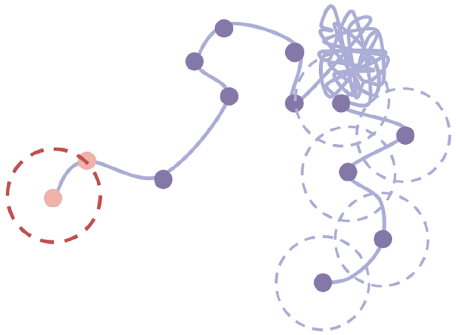


Fig. B.2. Illustration of the second stop point recheck method inspired by DBSCAN. To correct misclassified stop points at the segment boundaries caused by the sliding window approach, points with higher speed and greater distance from the stop cluster (analogous to low-density regions in DBSCAN) are identified and removed. The process checks whether boundary points are core points and stops when a non-core point is found.

Appendix C. Time and space complexity analysis of STASW

Problem scale and notation. Let N be the number of chronologically ordered trajectory points, K the number of detected stop segments, and M_k the length of the k-th segment ($\sum_{k=1}^{K} M_k \le N$). STASW uses constant window sizes for in-port and out-of-port regimes ($\omega_{\text{in}} = 5$ and $\omega_{\text{out}} = 10$). Coordinate conversion (UTM) and geohash encoding are per-point constant-time operations.

Base scan (Algorithm 2). For each trajectory point $i=1,\dots,N$, the algorithm updates three incremental statistics: (1) mean speed via sliding-window updates, (2) spatial dispersion using running minima and maxima in x and y, and (3) a single distance comparison against the current stop center. Each operation executes in $O(1)$ time, yielding an overall computational cost of $T_{\text{base}}=O(N)$ for the full pass, with auxiliary memory $S_{\text{base}}=O(1)$ in addition to the stored trajectory. Coordinate conversion to UTM is also $O(N)$ and can be cached for reuse.

Adaptive coastline switching. A majority rule over level 6 coastline cells is evaluated once per outer window to switch in-/out-of-port parameter sets. Each point contributes a single counter update, adding $O(N)$ time with negligible memory overhead.

Secondary rechecks. Type 1 performs boundary refinement with sliding thresholds inside each confirmed stop segment. Its total cost is $\sum_{k=1}^{K} O(M_k) = O(N)$ since segment lengths sum to N. Type 2 adopts a DBSCAN-like density filter. A naive implementation is $O(M_k^2)$ due to all-pairs distances. Using geohash bucketing (querying only same and adjacent cells) yields a constant-size candidate set per point and reduces the expected segment cost to $O(M_k)$, hence $O(N)$ overall in expectation. The $O(N^2)$ worst case occurs only when all points collapse into a single bucket without spatial indexing, which is not observed in port stop distributions.

Percentile estimation and adaptive thresholds. Mean-based thresholds are updated incrementally in $O(1)$ time per point. Percentile-based thresholds (e.g., P10, P99) are computed over bounded auxiliary buffers or candidate sets of size $B \ll N$, resulting in an $O(B \log B)$ cost per evaluation. Because B is fixed and independent of the trajectory length, these operations do not alter the overall linear-time complexity of the procedure.

Overall complexity. With spatial hashing for neighborhood queries, the total time is

$$T_{\mathrm{STASW}} = O(N) + O(N) + O(N) = O(N), \tag{C.1}$$

and the space is

$$S_{\mathrm{STASW}} = O(N)\ (\text{trajectory}) + O(K)\ (\text{segment indices}) + O(1)\ (\text{running statistics}) = O(N). \tag{C.2}$$

Thus STASW is linear-time and linear-space in expectation. The quadratic worst-case for density recheck does not arise in practice due to spatial dispersion and bucketing.

Implementation notes. Several practical measures are employed to ensure efficiency: (1) prefix sums and running extrema are maintained to eliminate window re-scanning; (2) UTM conversion is performed lazily, only when a point first qualifies as a candidate for geometric checks; (3) geohash bucketing with bounded neighbor expansion is used to restrict local searches; (4) early termination is applied within each window as soon as any stopping constraint is violated; and (5) percentile-based thresholds are computed in batches at the segment level to amortize sorting overhead.

Conclusion. With these design choices, the overall procedure achieves expected $O(N)$ time and $O(N)$ space complexity while retaining high recognition accuracy through adaptive thresholds and dual-stage refinement. This computational profile enables scalable application of the method to multi-million-point AIS datasets typical of large port environments.

Appendix D. Geohash-based boundary point identification

This appendix summarizes the geohash-based boundary indexing procedure used in Algorithm 1. Based on the level 5 edge mask shown in Fig. D.1, all geohash cells intersecting the functional-area outline are enumerated, after which only the outer-envelope cells are retained as the boundary set $\mathcal{B}$. This set is stored in a hash table to enable constant-time membership queries. Each AIS point reuses its level 5 geohash g_i to assign a boundary flag $b_i = [g_i \in \mathcal{B}]$, producing a sparse index list suitable for direct time-gap evaluation without resorting to point-in-polygon computation. The workflow parallelizes naturally: each worker performs independent hash lookup and conditional appends, and the resulting lists are merged in linear time, preserving overall $O(N)$ complexity with minimal contention. Memory requirements remain low because $\mathcal{B}$ typically contains only several dozen coarse cells, and the index list grows only with actual boundary-crossing events. The adopted level 5 resolution avoids excessive spatial fragmentation while reliably covering both channel mouths, and is tolerant to minor ENCs alignment offsets. This boundary-indexing layer significantly reduces the number of candidate checks required in Algorithm 1, providing a lightweight and scalable preprocessing mechanism that supports robust inbound/outbound recognition in large AIS datasets.

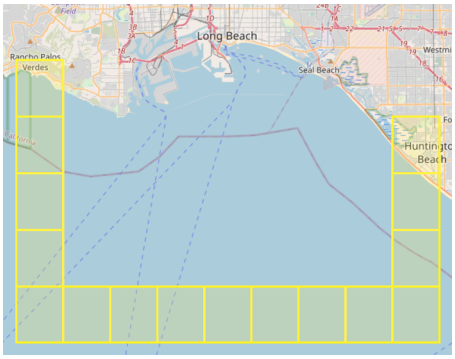


Fig. D.1. Visualization of the boundary of functional areas using level 5 geohash grids. The edge geohash cells are used to identify vessel inbound and outbound points for the corresponding region, supporting the extraction of inbound and outbound trajectories.

Appendix E. GNN model performance under different settings

This section evaluates the performance of the GNN baselines under multiple experimental configurations. All experiments adopt an 80/10/10 train-validation-test split, and the results reported correspond to test-set accuracy. As shown in Table E.1, performance is highly stable across random seeds, with accuracy ranging from 97.29% to 99.01%. Among the models, GraphSAGE attains the highest accuracy (99.01%, seed 8209).

Hidden-dimension sensitivity (Table E.2) remains limited: medium-sized embeddings (32–64) consistently yield the strongest performance, with GraphSAGE achieving an F1-score of 99.23% at 32 dimensions. By contrast, very small representations (e.g., 16 dimensions) reduce accuracy, indicating under-parameterization.

Learning rate exerts the greatest influence on model behavior (Table E.3). Extremely low learning rates (0.001) substantially impair ChebNet, GAT, and GCN, resulting in early optimization stagnation. Rates within the 0.005-0.05 range preserve high accuracy across all architectures, with peak performance observed for ChebNet (98.81% at 0.05) and GraphSAGE (98.74% at 0.01).

Overall, the GNN models demonstrate robustness to random initialization, prefer moderate hidden-dimension sizes, and require avoidance of excessively small learning rates to ensure stable and reliable generalization on the 10% held-out test set.

Table E.1
GNN performance across random seeds (hid_dim = 64, lr = 0.01).

Seed	Model	Accuracy	Precision	Recall	F1
3078	ChebNet	98.525	98.724	99.310	99.016
	GAT	98.425	98.408	99.502	98.952
	GCN	98.339	98.296	99.502	98.895
	GraphSAGE	98.940	98.935	**99.655**	99.293
4902	ChebNet	97.494	98.800	97.834	98.315
	GAT	98.439	98.519	99.406	98.960
	GCN	97.780	98.302	98.735	98.518
	GraphSAGE	98.883	98.859	**99.655**	99.256
5462	ChebNet	97.637	98.859	97.969	98.412
	GAT	98.539	98.687	99.368	99.026
	GCN	98.052	98.290	99.118	98.702
	GraphSAGE	98.783	98.765	99.617	99.189
8209	ChebNet	97.293	98.854	97.509	98.177
	GAT	98.382	98.425	99.425	98.923
	GCN	97.938	98.269	98.984	98.625
	GraphSAGE	**99.012**	**99.066**	99.617	**99.341**

Table E.2
GNN performance across hidden dimensions (seed = 5333, lr = 0.01).

hid_dim	Model	Accuracy	Precision	Recall	F1
16	ChebNet	92.653	98.336	91.721	94.913
	GAT	97.665	98.042	98.850	98.445
	GCN	97.307	98.106	98.294	98.200
	GraphSAGE	98.582	98.540	99.578	99.056
32	ChebNet	96.763	98.750	96.895	97.814
	GAT	97.981	97.978	99.348	98.658
	GCN	97.952	98.324	98.946	98.634
	GraphSAGE	**98.840**	98.803	**99.655**	**99.227**
64	ChebNet	98.596	98.818	99.310	99.063
	GAT	98.396	98.389	99.483	98.933
	GCN	97.680	98.318	98.582	98.450
	GraphSAGE	98.740	98.690	99.636	99.161
128	ChebNet	**98.840**	**99.120**	99.329	99.225
	GAT	98.568	98.650	99.444	99.046
	GCN	98.181	98.329	99.253	98.789
	GraphSAGE	98.783	98.746	99.636	99.189

Table E.3
GNN performance across learning rates (seed = 5333, hid_dim = 64).

lr	Model	Accuracy	Precision	Recall	F1
0.001	ChebNet	38.456	78.656	24.224	37.040
	GAT	41.908	84.136	27.443	41.387
	GCN	40.791	80.484	27.424	40.909
	GraphSAGE	95.317	98.416	95.266	96.816
0.005	ChebNet	97.221	98.141	98.141	98.141
	GAT	97.809	97.828	99.272	98.545
	GCN	98.066	98.217	99.214	98.713
	GraphSAGE	98.639	98.707	99.483	99.093
0.01	ChebNet	98.596	98.818	99.310	99.063
	GAT	98.396	98.389	99.483	98.933
	GCN	97.680	98.318	98.582	98.450
	GraphSAGE	98.740	98.690	**99.636**	99.161
0.05	ChebNet	**98.811**	**99.214**	99.195	**99.205**
	GAT	98.596	98.651	99.483	99.065
	GCN	98.339	98.388	99.406	98.894
	GraphSAGE	98.596	98.725	99.406	99.064

References

Bai, X., Xie, Z., Xu, X., Xiao, Y., 2023. An adaptive threshold fast DBSCAN algorithm with preserved trajectory feature points for vessel trajectory clustering. Ocean Eng. 280, 114930.
Bishop, C.M., Nasrabadi, N.M., 2006. Pattern recognition and machine learning. Vol. 4. Springer.
Chu, Z., Yan, R., Wang, S., 2025. Vessel arrival time to port prediction via a stacked ensemble approach: fusing port call records and AIS data. Transp. Res. Part C Emerg. Technol. 176, 105128.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. Adv. Neural Inf. Process. Syst. 29, 1–9.
Filipiak, D., Milena, S., Węcel, K., Abramowicz, W., Steidel, M., 2021. Application of AI and in-memory computing for extracting vessel movement patterns from historical data. In: 14th NATO Operations Research and Analysis (OR&A) Conference: Emerging and Disruptive Technology: Meeting Proceedings, North Atlantic Treaty Organization.
Filipiak, D., Węcel, K., Milena, S., Michalak, M., Abramowicz, W., 2020. Extracting maritime traffic networks from AIS data using evolutionary algorithm. Bus. Inf. Syst. Eng. 62 (5), 435–450.
Forti, N., Millefiori, L.M., Braca, P., 2019. Unsupervised extraction of maritime patterns of life from automatic identification system data. In: OCEANS 2019-Marseille. IEEE, pp. 1–5.
Gao, M., Shi, G.-Y., 2020. Ship-handling behavior pattern recognition using AIS sub-trajectory clustering analysis based on the t-SNE and spectral clustering algorithms. Ocean Eng. 205, 106919.
Gong, J., Li, H., Jiao, H., Yang, Z., 2025. Uncertainty-aware ship trajectory prediction via spatio-temporal graph transformer. Transp. Res. Part E Logist. Transp. Rev. 203, 104315.
Guo, Z., Qiang, H., Xie, S., Peng, X., 2024. Unsupervised knowledge discovery framework: from AIS data processing to maritime traffic networks generating. Appl. Ocean Res. 146, 103924.
Huang, L., Wan, C., Wen, Y., Song, R., van Gelder, P., 2024. Generation and application of maritime route networks: overview and future research directions. IEEE Trans. Intell. Transp. Syst. 26, 620–637.
Jiao, H., Gong, J., Li, H., Lam, J. S.L., Shu, Y., Wang, J., Yang, Z., 2026. Llm4stp: a large language model-driven multi-feature fusion method for ship trajectory prediction. Transp. Res. Part E Logist. Transp. Rev. 207, 104599.
Khalifeh, M., Caliskan, A., 2025. The role of port smartness in achieving sustainable development goals. Maritime Pol. Manage. 52 (1), 106–120.
Kipf, T.N., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
Kweon, S.J., Hwang, S.W., Lee, S., Jo, M.J., 2022. Demurrage pattern analysis using logical analysis of data: a case study of the ulsan port authority. Expert Syst. Appl. 206, 117745.
Li, H., Lam, J. S.L., Yang, Z., Liu, J., Liu, R.W., Liang, M., Li, Y., 2022. Unsupervised hierarchical methodology of maritime traffic pattern extraction for knowledge discovery. Transp. Res. Part C Emerg. Technol. 143, 103856.
Li, H., Xing, W., Jiao, H., Yang, Z., Li, Y., 2024. Deep bi-directional information-empowered ship trajectory prediction for maritime autonomous surface ships. Transp. Res. Part E Logist. Transp. Rev. 181, 103367.
Li, H., Yang, Z., 2023. Incorporation of AIS data-based machine learning into unsupervised route planning for maritime autonomous surface ships. Transp. Res. Part E Logist. Transp. Rev. 176, 103171.
Li, H., Zhang, Y., Li, Y., Lam, J. S.L., Matthews, C., Yang, Z., 2025. Deep multi-view information-powered vessel traffic flow prediction for intelligent transportation management. Transp. Res. Part E Logist. Transp. Rev. 197, 104072.
Liu, J., Ong, G.P., Chen, X., 2020. GraphSAGE-based traffic speed forecasting for segment network with sparse data. IEEE Trans. Intell. Transp. Syst. 23 (3), 1755–1766.
Liu, L., Zhang, M., Liu, C., Yan, R., Lang, X., Wang, H., 2025a. Shipping map: an innovative method in grid generation of global maritime network for automatic vessel route planning using AIS data. Transp. Res. Part C Emerg. Technol. 171, 105015.
Liu, M., Lai, K.-h., Wong, C. W.Y., Xin, X., Lun, V. Y.H., 2025b. Smart ports for sustainable shipping: concept and practices revisited through the case study of China's Tianjin port. Maritime Econ. Logist. 27 (1), 50–95.
Liu, Z., Gao, H., Zhang, M., Yan, R., Liu, J., 2023a. A data mining method to extract traffic network for maritime transport management. Ocean Coastal Manage. 239, 106622.
Liu, Z., Zhang, B., Zhang, M., Wang, H., Fu, X., 2023b. A quantitative method for the analysis of ship collision risk using AIS data. Ocean Eng. 272, 113906.
Lu, N., Liang, M., Zheng, R., Liu, R.W., 2020. Historical AIS data-driven unsupervised automatic extraction of directional maritime traffic networks. In: 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA). IEEE, pp. 7–12.
Ma, J., Zhou, Y., Chang, Y., Zhu, Z., Liu, G., Chen, Z., 2024. TG-PGAT: an ais data-driven dynamic spatiotemporal prediction model for ship traffic flow in the port. J. Mar. Sci. Eng. 12 (10), 1875.
Ma, Q., Lian, S., Zhang, D., Lang, X., Rong, H., Mao, W., Zhang, M., 2025. A machine learning method for the recognition of ship behavior using AIS data. Ocean Eng. 315, 119791.
Ma, Y., Zhang, Z., Chen, S., Pan, Y., Hu, S., Li, Z., 2021. Investigating the impact of spatial-temporal grid size on the microscopic forecasting of the inflow and outflow gap in a free-floating bike-sharing system. J. Transp. Geogr. 96, 103208.
MISM, 2020. The use of AIS data to identify dark activity for marine auditing purposes. https://mismarine.com/wp-content/uploads/2020/02/The-Use-of-AIS-Data-to-Identify-Dark-Activity-for-Marine-Auditing-Purposes-v3.pdf.
Mujal-Colilles, A., Guarasa, J.N., Fonollosa, J., Llull, T., Castells-Sanabra, M., 2022. Covid-19 impact on maritime traffic and corresponding pollutant emissions. the case of the port of Barcelona. J. Environ. Manage. 310, 114787.
Niu, B., Dong, J., Wang, H., 2024. Smart port vs. port integration to mitigate congestion: ESG performance and data validation. Transp. Res. Part E Logist. Transp. Rev. 191, 103741.
Pallotta, G., Vespe, M., Bryan, K., 2013. Vessel pattern knowledge discovery from AIS data: a framework for anomaly detection and route prediction. Entropy 15 (6), 2218–2245.
Qiang, H., Guo, Z., Chu, Z., Xie, S., Peng, X., 2025a. Motion-inspired spatial–temporal transformer for accurate vessel trajectory prediction. Eng. Appl. Artif. Intell. 148, 110391.
Qiang, H., Guo, Z., Peng, X., Jia, C., 2025b. Fdbr: ultra-fast and data-efficient behavior recognition of port vessels using a statistical framework. Ocean Eng. 315, 119737.
Rong, H., Teixeira, A.P., Soares, C.G., 2022. Maritime traffic probabilistic prediction based on ship motion pattern extraction. Reliab. Eng. Syst. Saf. 217, 108061.
Rong, H., Teixeira, A.P., Soares, C.G., 2024. A framework for ship abnormal behaviour detection and classification using AIS data. Reliab. Eng. Syst. Safety 247, 110105.
Santos, M.Y., Silva, J.P., Moura-Pires, J., Wachowicz, M., 2012. Automated traffic route identification through the shared nearest neighbour algorithm. In: Bridging the Geographic Information Sciences: International AGILE'2012 Conference, Avignon (France), April, 24–27, 2012. Springer, pp. 231–248.
Song, Z.-Y., Lin, C.-W., Feng, X., Lee, P. T.-W., 2024. An empirical study of the performance of the sixth generation ports model with smart ports with reference to major container ports in mainland China. Transp. Res. Part E Logist. Transp. Rev. 184, 103460.
Suo, Y., Wang, Y., Cui, L., 2025. Ship anomalous behavior detection based on BPEF mining and text similarity. J. Mar. Sci. Eng. 13 (2), 251.
Teixeira, A.P., Guedes Soares, C., 2018. Risk of maritime traffic in coastal waters. In: International Conference on Offshore Mechanics and Arctic Engineering. Vol. 51326. American Society of Mechanical Engineers, V11AT12A025.
UNCTD, 2024. Review of Maritime Transport 2024. https://unctad.org/system/files/official-document/rmt2024_en.pdf.
Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.
Wei, Z., Gao, Y., Zhang, X., Li, X., Han, Z., 2024. Adaptive marine traffic behaviour pattern recognition based on multidimensional dynamic time warping and DBSCAN algorithm. Expert Syst. Appl. 238, 122229.
Wijaya, W.M., Nakamura, Y., 2024a. Port performance indicators construction based on the AIS-generated trajectory segmentation and classification. Int. J. Data Sci. Analyt. 20, 1–20.
Wijaya, W.M., Nakamura, Y., 2024b. Ship navigational status classification based on the geometrical and spatiotemporal features of the AIS-generated trajectory. In: 2024 The 9th International Conference on Big Data Analytics, pp. 103–112.
Wu, L., Xu, Y., Wang, Q., Wang, F., Xu, Z., 2017. Mapping global shipping density from AIS data. J. Navig. 70 (1), 67–81.

Xiao, G., Wang, Y., Wu, R., Li, J., Cai, Z., 2024. Sustainable maritime transport: a review of intelligent shipping technology and green port construction applications. J. Mar. Sci. Eng. 12 (10), 1728.
Xin, R., Pan, J., Yang, F., Yan, X., Ai, B., Zhang, Q., 2024. Graph deep learning recognition of port ship behavior patterns from a network approach. Ocean Eng. 305, 117921.
Yan, Z., Cheng, L., He, R., Yang, H., 2022. Extracting ship stopping information from AIS data. Ocean Eng. 250, 111004.
Yang, D., Wu, L., Wang, S., Jia, H., Li, K.X., 2019. How big data enriches maritime research–a critical review of automatic identification system (AIS) data applications. Transp. Rev. 39 (6), 755–773.
Yoon, S., Jung, S.U., Kweon, S.J., Lee, S., Na, H.S., 2025. A strategic plan to provide management services for urban green spaces during heat waves using a collaborative truck-and-robot system. Expert Syst. Appl. , 128572.
Yun, Y., Na, H.S., Kweon, S.J., 2025. Optimal routing strategies for last-mile delivery with a fleet of a single truck and multiple heterogeneous robots. Transp. Res. Rec. 2679 (10), 506–534.
Zhang, L., Meng, Q., Fwa, T.F., 2019. Big AIS data based spatial-temporal analyses of ship traffic in Singapore port waters. Transp. Res. Part E Logist. Transp. Rev. 129, 287–304.
Zhang, Z., Huang, L., Peng, X., Wen, Y., Song, L., 2022. Loitering behavior detection and classification of vessel movements based on trajectory shape and convolutional neural networks. Ocean Eng. 258, 111852.
Zrnić, D.N., Dragović, B.M., Radmilović, Z.R., 1999. Anchorage-ship-berth link as multiple server queuing system. J. Waterway Port Coast. Ocean Eng. 125 (5), 232–240.