Expert Systems With Applications 312 (2026) 131525

Contents lists available at ScienceDirect

Expert Systems With Applications

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

MSIF-SSTR: A "Quick smuggler" smuggling speedboat trajectory recognition method based on multi-source information fusion

Zhuhua Hu [a], Yifeng Sun [a], Yaochi Zhao [b,*], Wei Wu [a], Lingkai Kong [a], Keli Chen [a]

[a] School of Information and Communication Engineering, Hainan University, No. 58, Renmin Avenue, Haikou, 570228, Hainan, China
[b] School of Cyberspace Security (School of Cryptology), Hainan University, No. 58, Renmin Avenue, Haikou, 570228, Hainan, China

A R T I C L E I N F O

Keywords:
Ship smuggling behavior
Offshore ship control
Multi-source information fusion
TCN
LSTM

A B S T R A C T

Cooperating with maritime administrative departments to identify smuggling activities and enhance the control ability of nearshore vessels holds significant practical significance. However, existing research mostly relies on basic AIS data and simple features, making it difficult to deal with complex vessel behaviors. Especially when identifying covert and flexible smuggling activities, it is prone to misjudgment and has limited effectiveness. In real-world enforcement, distinguishing truly suspicious "Quick Smuggler" smuggling from benign high-speed transit requires modeling subtle, deep-level spatio-temporal cues that couple motion dynamics with external conditions (e.g., wind, wave, visibility) and context. Simple linear mappings and shallow temporal encoders often overfit speed bursts or local detours, causing elevated false alarms. By contrast, dilated-convolutional receptive fields in TCNs capture multi-scale temporal dependencies efficiently, while KAN layers provide adaptive nonlinear function bases to fit complex, locally varying trajectory patterns. This synergy is particularly suited to covert nighttime operations under shifting sea states, where genuine smuggling exhibits trajectory micro-structures and weather-conditioned behaviors that are hard to emulate by normal craft. To address these challenges, this study proposes a Multi-Source Information Fusion-based "Quick Smuggler" Smuggling Speedboat Trajectory Recognition method (MSIF-SSTR). First, we construct the HN_BF dataset, comprising real-world nighttime radar trajectories from the Qiongzhou Strait and corresponding meteorological data. Next, parallel TCN networks are employed to separately extract motion features, and meteorological features, enabling the model to better capture global temporal dependencies during feature extraction. Finally, the fused features are fed into an LSTM for classification, while a Kolmogorov-arnold networks (KAN) module replaces traditional fully connected layers to improve the representation of complex trajectory patterns. Experimental results demonstrate that MSIF-SSTR achieves F1-scores exceeding 94.2% on the HN_BF dataset, outperforming state-of-the-art methods with higher computational efficiency. Field applications confirm the model's robustness.

1. Introduction

Review of Maritime Transport 2023, published by the United Nations Conference on Trade and Development (U. N. C. on Trade, 2023), states that,although global seaborne trade declined slightly by 0.4% in 2022, it will continue to grow slowly in the future as the economy recovers. Meanwhile, in the analysis of global maritime trends, it is mentioned that despite the major challenges posed by global crises such as the war in Ukraine, maritime trade grew by 2.4% in 2023 and will continue to grow by more than 2% per year from 2024 to 2028, reflecting the resilience and great potential of the seaborne trade industry.

As a major maritime country, China's ships are important military equipment and maritime transportation carriers that require focused control and monitoring(Xiyu Fan et al., 2024). Hainan Island, as the second largest island in China, has an independent geographical unit, and most goods need to be transported in and out of the island by ship. Due to the long coastline around Hainan Island, and the existence of a large number of unregistered ship names and numbers, ship certificates, ship nationality of the "three noes" of the ship's historical reasons, resulting in the current situation of Hainan Island on the ship supervision is difficult (Junfei Chen et al., 2024). As a result, these ships have also become an important carrier and channel tool for smuggling, and if they are

* Corresponding author.
E-mail addresses: eagler_hu@hainanu.edu.cn (Z. Hu), 2945528421@qq.com (Y. Sun), zhyc@hainanu.edu.cn (Y. Zhao), 21110810000025@hainanu.edu.cn (W. Wu), konglingkai2001@163.com (L. Kong), 2452930273@qq.com (K. Chen).

https://doi.org/10.1016/j.eswa.2026.131525
Received 27 August 2025; Received in revised form 17 January 2026; Accepted 2 February 2026
Available online 4 February 2026

not properly supervised around the coastal waters of Hainan Province, these loopholes are very likely to be utilized by lawless elements for smuggling offences.

The "Quick Smuggler" is a large motorboat that has been power-modified and is a common means of maritime transportation for smuggling activities around Hainan Island. When it's traveling at high speed on the sea, it looks like it's flying close to the surface of the sea, so it's commonly known as "Quick Smuggler". Generally "Quick Smuggler" will be equipped with multiple engines with 250 horsepower, its maximum speed can exceed 90 km per hour, smugglers often use it to carry out smuggling activities. The use of "Quick Smuggler" as a means of smuggling will bring about multiple hazards: First, smuggling activities may involve contraband, and the entry of these items into the country poses a serious threat to social security and national stability. Secondly, smuggling bypasses normal customs duties and import procedures, leading to a loss of national tax revenue and affecting the economic order. Large-scale smuggling may also undermine fair competition in the domestic market and impact legitimate industries. Finally, "Quick Smuggler" travels at sea at very high speeds, which can easily cause marine traffic accidents and jeopardize the lives of legitimate vessels and their crews. Therefore, timely detection of smuggling behavior is very necessary.

At present, ship trajectory data processing, analysis and application have gradually become an emerging field in the shipping industry. However, in the identification process of smuggling work, its degree of intelligence is obviously insufficient. There are a large number of false alarms in the detection of smuggling trajectories using rule-based methods, and they are often dealt with in the aftermath of law enforcement, which lacks the early warning of combating smuggling. The manual screening method is not only costly and inefficient, but also highly dependent on the operator, while this overly subjective method will also easily cause risks such as omission and misjudgment. Therefore it is necessary to introduce intelligent recognition methods in smuggling track detection (Lihang Chen et al., 2025). However, genuine smuggling trajectories differ from mere high-speed transit in nuanced ways: (i) burst durations co-occur with sharp yet coordinated heading changes; (ii) speed profiles exhibit non-stationary segments aligned to tide and wind windows; (iii) approach and egress paths show multi-scale periodicities reflecting rendezvous constraints. Capturing these deep temporal signatures benefits from TCN's dilated convolutions for long-range dependency without recurrence overhead, while KAN's adaptive spline-based nonlinearity approximates locally varying motion-weather couplings that standard MLP heads underfit. This provides the methodological motivation for the proposed architecture beyond raw speed thresholds. Deep learning based ship trajectory anomalous behavior detection methods have been widely researched in recent years due to the advantages of good results and fast computation in practice, but these researches only introduce basic ship trajectory datasets, such as AIS data. These methods are usually effective for other kinds of anomaly detection (Liao et al., 2024). However, as smuggling is covert and evasive, it often occurs late at night and in poor weather conditions. If the detection model does not take into account the influence of external objective factors on the ship's behavior, and only judges whether the ship's trajectory is suspected of smuggling from a few simple ship movement characteristics, it is easy to cause a large number of ships with similar behaviors to be judged as the "Quick Smuggler" trajectory of false alarms (Tianjiao Wei et al., 2025).

Due to the influence of a variety of factors, such as ship type, voyage purpose, sea state, etc., which are intertwined together, real ship trajectory sequences usually show a high degree of complexity and diversity. The ship's motion pattern not only has large fluctuations, but also changes significantly with time and environment. Such complex dynamic features relying only on simple linear mapping and nonlinear activation functions are difficult to comprehensively characterize the trajectory change law, and at the same time, it is also difficult to adequately capture the deep-level features of ship behavior and its potential spatial-temporal dependence, which leads to the model's performance being limited in responding to the actual complexity of the situation, and it is unable to accurately identify the potential anomalous behaviors in the narrower sense, especially in responding to the smuggling activities with a high degree of concealment and flexibility.

Therefore, we summarize two key issues in the smuggling behavior recognition task:

- At present, most of the studies only use the basic AIS data for the detection of abnormal ship behavior. Although this type of method is applicable to most abnormal behaviors, smuggling activities often take place in hidden environments, especially late at night or in bad weather. If the model ignores the influence of external factors on ship behavior and judges based on a small number of trajectory movement characteristics alone, it is easy to lead to a large number of normal trajectories being misjudged as smuggling trajectories.
- Ship trajectories are influenced by multiple factors and exhibit complexity and diversity, and simple linear mapping and nonlinear function have difficulty in capturing trajectory movement patterns, especially when dealing with smuggling activities that are stealthy and flexible, a problem that leads to limited modeling effectiveness.

Recently, KolmogorovArnold Networks (KAN) (Liu et al., 2024)have generated a lot of attention in the field of deep learning.This network proposes a new neural network architecture and suggests a new scheme that can replace the existing multilayer perceptron (MLP).The most innovative aspect of KAN compared to the traditional MLP is that it is no longer a pre-fixed activation function The most innovative feature of KAN compared to traditional MLP is that it is no longer a pre-fixed activation function, but a parameterisable and learnable activation function. In layman's terms, compared to MLPs that learn to draw straight lines and then sum them for nonlinear activation, KAN chooses to learn to draw curves and then sum them. This unique structure allows for more efficient capture of complex patterns in dynamic trajectories. Trajectory data often contain highly nonlinear and locally correlated features, and KAN's B-spline function can be flexibly adapted locally in the feature space to effectively capture local variations in the trajectory. This design makes KAN more sensitive to the response of a small range of features, which improves the recognition accuracy of anomalous trajectories.

So, we propose a deep learning method for multi-source information fusion for ship smuggling trajectory recognition, named MSIF-SSTR.The main contributions of our work are as follows:

- In order to standardize data modeling for "Quick Smuggler" smuggling boats, we created a real-world "Quick Smuggler" trajectory identification dataset, named HN_BF, which collects night-time radar data near the Qiongzhou Strait and labels ship trajectories suspected to be "Quick Smuggler".
- In order to adress the high misjudgment of normal trajectories as smuggling ones caused by over-reliance on basic AIS data and neglect of external factors, we propose a multi-source information fusion network for detecting suspected smuggling trajectories, named MSIF-SSTR. It employs independent TCN branches to separately extract motion and weather features from trajectory sequences, fuses the dual-modal features, and feeds them into an LSTM network integrated with a Multi-Head Attention mechanism-enhancing processing of long sequences and highlighting critical information for trajectory discrimination.
- In order to the limited modeling performance arising from simple linear/nonlinear functions' inability to capture complex smuggling trajectory patterns, we introduce the KAN module to replace traditional fully connected layers. Leveraging adaptive B-spline functions for flexible nonlinear mapping, it more accurately characterizes the dynamic, nonlinear trajectory patterns of stealthy and flexible smuggling vessels.

The rest of this paper is organized as follows: The Section 2 describes the relevant work of ship abnormal behavior detection methods. The

MSIF-SSTR model proposed in this work is described in detail in Section 3. The Section 4 gives the details and discussion of the experiment. The Section 5 summarizes the work.

2. Related work

This chapter systematically reviews the definition of abnormal ship behavior and the main detection methods, summarizes the limitations of existing studies, and further clarifies the differences between this study and previous work, highlighting the innovative value and practical significance of this research.

2.1. Definition of abnormal ship behavior

In the field of maritime management, the detection of abnormal ship behavior is an important part of ensuring the safety of shipping and preventing illegal activities. The term “abnormal ship behavior” encompasses a wide range of behavioral patterns, and the definition of so-called “abnormality” has different meanings and manifestations in different temporal and spatial dimensions.

Martineau and Roy (Martineau & Roy, 2011) classified ship behaviours into two types: motion anomalies and positional anomalies. Portnoy et al (Portnoy, 2000) defined anomalous behaviours based on the discrepancy between mathematical models of normal ship behaviours and actual ship data. Zhang and Tang (Zhang & Tang, 2015) described anomalous ship behaviours as behaviours that do not conform to the normal patterns of navigation. Laxhammar (Laxhammar, 2008) further defines abnormal ship behaviour to include abnormal deviations in heading and course, sudden acceleration and deceleration, and presence in prohibited areas. Literature (Kazemi et al., 2013), in consultation with a number of experts in the field, has developed 11 rules for unusual ship behaviour. In contrast, literature (Terroso-Saenz et al., 2016) classifies abnormal ship behaviour into abnormal behaviour of a single ship and abnormal behaviour of multiple ships.

From the above, it can be seen that different studies have different focuses on the definition of abnormal ship behavior. Generally speaking, abnormal ship behavior can be divided into two definitions, broad and narrow, according to the existence of a priori knowledge. The broad definition of abnormal ship behavior is to first define the behavioral pattern of “normal” state under specific situation and data, and then regard all the behaviors deviating from this “normal” state as abnormal, such as speeding, etc. The narrow defined abnormal behavior of ships, on the other hand, is based on the pre-defined unconventional behavioral patterns such as the legitimacy of the ship’s identity, the harmfulness of its actions, the danger of its state, etc., and determines the behavioral patterns that are not in line with the routine, such as positional conflicts or illegal intrusion.

For the above two definitions of ship abnormal behavior, the direction of ship abnormal behavior detection is also different. For the broad abnormal ship behavior detection task, it is necessary to take the normal behavior pattern of a specific ship or a specific type of ship as the basis, and compare it with the current trajectory that needs to be detected, in order to detect whether the current behavior of the ship is abnormal or not. For the narrow abnormal ship behavior detection task, it is necessary to formulate relevant knowledge, experience, international practices and laws and regulations to predefine the abnormal ship behavior pattern, and then the current ship behavior is compared with the predefined abnormal behavior pattern, so as to judge whether the behavior of the ship concerned belongs to abnormality or not. Obviously, smuggling is an abnormal ship behavior in the narrow sense.

2.2. Ship anomaly behavior detection methods

Currently, the existing methods for ship abnormal behavior detection mainly include rule-based methods, cluster analysis methods, statistical modeling methods, machine learning methods, and so on. In order to show the research history of ship abnormal behavior detection more clearly, we have organized a RoadMap to record the related research, as shown in Fig. 1.

Rule-based approaches focus on detecting dynamic features of ship trajectories or the evolution of a particular trajectory by means of pre-defined rules and thresholds, e.g., literature (Kazemi et al., 2013; Terroso-Saenz et al., 2016; Zhang & Tang, 2015). However, the development of these rules requires the knowledge of domain experts, which may introduce significant human bias in the analysis process In addition, the rules and thresholds for the same anomalous behavior may even vary due to the effects of different types of ships, different traffic vessel flows, and different water navigation environments. Therefore, the scope of application of rule-based methods is often limited.

Among the clustering-based methods for detecting abnormal behaviours of ships, Zhao and Shi (2019) proposed a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and recurrent neural network scheme for detecting abnormal behaviours in the multidimensional features of a ship. Wang et al. (2021), on the other hand used an electronic charting system to collect data on the recurring abnormal behaviours of ships during navigation, and processed these data through spatio-temporal analysis and state determination methods to detect the ship’s trajectory characteristics. In addition, literature (Zhen et al., 2017) uses clustering methods to process ship trajectories, and then combines the clustering results with a simple Bayesian classifier to achieve the detection of abnormal ship behaviour. Aiming at the problem of possible bias in raw AIS data, Lei and Mingchao (2018) used DBSCAN to extract mathematical models from the trajectory clustering results to identify abnormal behaviours. However, clustering methods are often affected by factors such as initial parameter settings, which may significantly reduce the accuracy of experimental results. In addition, clustering methods usually face the challenges of high computational and time complexity in the process of data collection and analysis. In the statistical modeling-based approach, Xiao et al. (2015) presented some AIS dynamic information (lateral position, speed, heading, time interval) in a statistical manner to characterize ship traffic. Literature (Kowalska & Peel, 2012) proposed a nonparametric Bayesian model based on Gaussian process and used an active learning paradigm to select an information-rich subsample to model normal ship behavior for abnormal trajectory detection. Laxhammar (Laxhammar, 2008) enhanced the Holst method by integrating expectation maximization with a Gaussian mixture model. This algorithm is effective in identifying certain notable abnormal behaviors; however, it relies on a strong data distribution. When the data does not conform to a Gaussian distribution, the results of the anomaly detection are less effective. Rong et al. (2020) proposed a data mining vector-based probabilistic characterization of ship routes to characterize offshore marine traffic and allow real-time anomalous behavior detection. However, these methods also tend to suffer from the drawbacks of high complexity, difficult parameter selection, and long running time, and they basically can only recognize for specific data and lack generalization performance. Pallotta et al. (2013) used a statistical approach to extract and analyze AIS data, and then used unsupervised and incremental learning methods to learn ship motion patterns and perform ship behavior recognition.

Among the machine learning based methods, Zhang et al. (2023) first extracted normal trajectories using clustering method with noise application and used it as a canonical model, and then trained recurrent neural network to realize ship anomaly detection. Dan et al. (2022) proposed an LSTM-FCN model, which predicts multidimensional feature trajectories with significantly better accuracy than traditional recurrent neural networks. Vespe et al. (2012) uses an unsupervised learning algorithm to automatically recognise ship movement patterns from AIS data. The algorithm has the ability of incremental learning, which can dynamically adjust the model according to the changes in the environment, and thus effectively carry out the anomaly detection of ship trajectories. Gu et al. (2024)used a two-tower Transformer network to process time series in trajectory data and multidimensional information in feature space

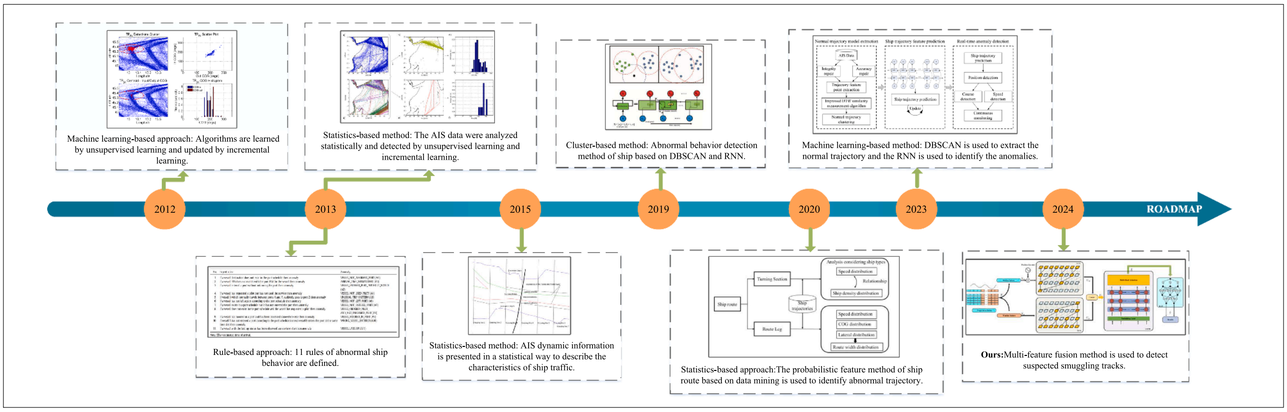


Fig. 1. Development of ship anomaly behavior detection.

to enable recognition of specific types of trajectories. Literature (Liang & Zhang, 2020; Ren et al., 2019; Tang et al., 2022) presents a method for ship trajectory prediction using recurrent neural networks and long and short-term memory models. When the ship's trajectory is predicted correctly, anomalous behaviours are judged against a set threshold. Although these models perform well in handling trajectory sequence data, the choice of the threshold value has a significant impact on the final result. For the smuggling identification task, literature (Wen et al., 2017) developed dependent and independent classification tree models and Bayesian network models for investigating the dependencies between smuggled goods (e.g., fishery products, weapons, and drugs) and other variables (e.g., home port information of the vessel, the age of the vessel owner, and the arrival and departure dates of the vessel) in order to identify vessels with the highest smuggling value. Literature (Wen et al., 2012) analyses data on historical smuggled fishing vessels in Taiwan using the artificial neural network (ANN) model in information technology and the logistic regression (LR) model in statistics, thus constructing an identification model for different fishing vessel types. However, the two methods of detecting smuggled vessels described above are aimed at regular vessels that are registered and recorded. As a matter of fact, many of the vessels used in smuggling incidents are illegally assembled vessels, which will not be registered, so the above methods have great limitations. Moreover, the above method mainly focuses on the risk assessment of smuggling behavior by ships, and is of little help in combating and early warning of smuggling behavior in real scenarios. At present, the detection of smuggling track is still a blank field.

At present, the detection task of smuggling mainly relies on law enforcement departments to carry out daily patrols, and screen out tracks and conduct manual screening based on rules, and then rely on hardware facilities such as port, dock video surveillance or traffic bayonet transportation monitoring for manual duty. Very lack of information support, lack of early warning against smuggling, while consuming a lot of manpower, material resources and time costs.

Based on the limitations of the various studies mentioned above, this paper uses multi-source information fusion, supplemented by external meteorological factors to determine whether there is a smuggling risk in the trajectory sequence, aiming at constructing an accurate, efficient, generalized and intelligent smuggling trajectory recognition model.

Temporal Convolutional Networks (TCN) (Bai et al., 2018) have been adopted in a range of trajectory and time-series anomaly detection tasks for their stable gradients and efficient parallelism. In maritime analytics, convolutional encoders have been leveraged to model AIS/radar sequences for route characterization and abnormality scoring, offering competitive accuracy and inference latency compared to recurrent baselines. Beyond AIS-based detection (Pallotta et al., 2013; Rong et al., 2020; Zhang et al., 2023), recent multi-modal designs integrate environmental covariates, where convolutional receptive fields help align long-range seasonal/meteorological influences with local kinematic cues. Compared with Transformers (Nie et al., 2023; Wu et al., 2023a) that require careful sequence length scaling, TCNs provide controllable receptive fields via dilation and often excel under limited compute or latency constraints, making them practical for near real-time coastal surveillance.

Lightweight forecasters (DLinear, LightTS) (Zeng et al., 2023; Zhang et al., 2022) and patch-wise Transformers (Nie et al., 2023) achieved progress in long-horizon forecasting, but anomaly detection in high-speed maritime pursuit involves abrupt regime shifts and multi-scale patterns. Empirically, convolutional sequence models remain strong baselines for event localization, while hybrid pipelines-convolutional encoders with attention fusion-balance robustness and capacity. Our design follows this evidence, adopting TCN backbones with attention-based fusion and a KAN classifier to enhance nonlinear separability.

3. Proposed method

To achieve precise identification of the smuggling boat trajectories of 'Quick Smuggler', this chapter systematically presents the complete technical solution of the MSIF-SSTR method from three perspectives: 'problem formalization - dedicated dataset construction - model architecture design.' First, the task objective of 'Quick Smuggler' trajectory recognition is clearly defined through mathematical formalization, providing a rigorous logical starting point for subsequent research; then, the construction process of the HN_BF dataset is described in detail, addressing the lack of real-world scenario data in existing studies; finally, the core modules of the MSIF-SSTR model are deconstructed, illustrating how it overcomes the limitations of traditional detection methods-such as high false alarm rates and weak nonlinear modeling capabilities in complex environments-through multi-source information fusion and network structure optimization, ensuring the method's scientific validity and practical applicability.

3.1. Problem definition

The ultimate goal of multivariate time series anomalous behavior detection is to detect anomalous behavior at the entity level (Su et al., 2019).

The content of this subsection is mainly to formalize the "Quick Smuggler" trajectory recognition problem. A ship's trajectory can be viewed as a set of trajectory points in the form shown in Eq. 1:

$$\begin{cases} D = \{X_1, X_2, \ldots, X_n\} \\ X_i = \{x_{i1}, x_{i2}, \ldots, x_{iT}\} \\ x_{ij} = (\text{ Latitude }_{ij}, \text{ Longitude }_{ij}, \text{ Speed }_{ij} \ldots) \end{cases} \tag{1}$$

where the dataset D is a collection of multiple ship trajectory sequences, with each X representing a complete trajectory sequence. For a ship trajectory sequence $X_i \in R^{T*m}$ whose length is T and number of features is m in ship trajectory data set D, the trajectory points sampled by X_i at time j is expressed as $x_{ij} = (Latitude_{ij}, Longitude_{ij}, Speed_{ij} \ldots) \in R^{1*m}$, where $Latitude_{ij}$, $Longitude_{ij}$ and $Speed_{ij}$ denote the latitude, longitude, and speed of the trajectory sequence X_i at the $j-th$ moment. In addition, x_{ij} contains other ship trajectory characteristics such as time, heading and bow direction at that moment.

In the dataset, each trajectory X has a corresponding label y, as shown in Eq. 2.

$$[X_1, y_1], [X_2, y_2], \ldots, [X_n, y_n] \tag{2}$$

In the task of identifying trajectories of "Quick Smuggler", the model learns the data distribution of each trajectory sequence X and its corresponding label y in the training phase. In the testing phase, the model generates a predicted label $\hat{y}$ for each trajectory sequence X in a specific category to determine whether the trajectory sequence has a suspected smuggling feature, as shown in Eq. 3. Finally, the predicted label $\hat{y}$ is compared with the real label y is compared to determine whether the recognition is correct or not, and each evaluation index is calculated.

$$X \xrightarrow{input} Network \xrightarrow{Predict} \begin{cases} \hat{y} = 1 \\ \hat{y} = 0 \end{cases} \tag{3}$$

3.2. Dataset

The quality of the dataset is the basis of the deep learning approach. Raw trajectory data has various problems that make it difficult to feed directly into the model for learning, so the raw trajectories need to be processed so that the model can maximize the trajectory features extracted from the data, which will lead to an improvement in modeling performance. The raw trajectory data used in this study is based on ship trajectories near the Qiongzhou Strait in China, sampled from March to May 2024, and the trajectory data used are all pure radar data because smuggled ships do not install AIS equipment.

During the production of the dataset in this study, we adopted a multilevel calibration approach to the labeling of ship data (Sun et al., 2024). In the first step, a rules-based method was used to collect the non-AIS, radar-only tracks with a speed higher than 16 knots and lasting for more than 10 minutes near the Qiongzhou Strait between 21:00 at night and 06:00 the next day. Usually, trajectories with these characteristics are recognized as suspected "Quick Smuggler" trajectories and are continuously tracked, and a total of 5665 trajectories are collected. In the second step, the trajectories were data cleaned to remove the abnormal and missing trajectories, and 5337 trajectories were obtained. In the third step, these tracks were manually verified and labeled, and before this step, we went through professional training and analysis of past confirmed smuggling cases, and finally obtained a total of 1473 tracks suspected to be "Quick Smuggler", while the other 3864 tracks were considered normal tracks with no smuggling risk.

After that, these trajectories are preprocessed, and the steps include: grouping the sampling points of different trajectories and sorting them according to the timestamps to transform them into a spatio-temporal sequence of trajectories; resampling each trajectory with a time interval of 1min; eliminating other features captured by the sensors, retaining longitude, latitude, speed, and heading features as the ship's motion features; deleting the null values. The preprocessed trajectory data eliminates redundant and useless information compared with the original data, which enables the model to better learn the trajectory feature information.

Considering the external weather factors have some influence on the appearance of smuggling behavior. Our method expands on the basis of the original ship trajectory movement features and constructs a dataset with more features. The weather features of offshore sea area weather forecast from China Central Weather Station are selected as the data source, and the selected weather features include weather phenomena, wind direction, wind force, wave height and visibility, and the update frequency of weather is 12 hours. Meanwhile, due to past cases, smuggling vessels often enter the Qiongzhou Strait from the Beibu Gulf, thus adding the sea area where the trajectory is located as a feature. After that, the discrete and disordered features (weather phenomena, wind direction and sea area) are converted into One-hot coding so that the model can better understand the objective factors. The format of the dataset is shown in Table 1, where the columns "latitude" to "course" are the motion features of the trajectory, and the columns "windPower" to "Wind_8" are the external environment features of the time period, the "id" column is used to distinguish different tracks, and the "label" column is used to distinguish whether the track is suspected to be the "Quick Smuggler".

The baseline meteorological source is the China Central Weather Station offshore forecast with 12-hour updates. To improve temporal alignment, we downscaled to hourly resolution via time-aware interpolation: (i) continuous variables (e.g., wind force, wave height, visibility) are linearly interpolated and then smoothed with a small-window median filter to damp sensor jitters; (ii) directional variables (wind direction) are interpolated on the unit circle using circular statistics; (iii) categorical phenomena are forward-filled within stability windows and reconciled with a mode filter when transitions occur. We then align hourly weather to 1-minute resampled trajectories by nearest-neighbor snapping within a ± 30 minute tolerance. This preserves sharp changes that may co-trigger high-speed maneuvers. When hourly feeds are unavailable, we fall back to the 12-hour bins and tag them so the model can learn uncertainty.

The curated HN_BF contains 1473 suspected "Quick Smuggler" trajectories and 3864 normal ones (ratio ≈ 1:2.6). During training, we mitigate imbalance by: (i) stratified 6:2:2 splits that maintain class proportions; (ii) class-weighted cross-entropy with inverse-frequency weights; and (iii) light over-sampling of the minority class in the batch sampler, improving recall without degrading precision.

3.3. Model architecture

The MSIF-SSTR network structure is shown in Fig. 2. The model can be divided into three parts. The Trajectories Series on the far left side of the figure represents the HN_BF dataset, i.e., the set of trajectories D, which contains multiple sequences of trajectories X, which are formatted as shown in Table 1 presented in Section 3.2. The blue part of the Trajectories Series represents the motion feature X_M of the trajectory, and the brown part represents the weather feature X_W in which the trajectory is running during the time period. X_M and X_W are fed into different TCN networks for feature extraction, where X_M enters into a modified TCN network with additive position coding. After that, the feature information U_M and U_W extracted from the two network modules are spliced to obtain the intermediate hidden vector representation U, which is then fed into the LSTM network with the attention mechanism to be fused to obtain the multi-head attention output S. Finally, the prediction result $\hat{y}$ on the trajectory is output through the KAN layer. Motion kinematics and weather signals exhibit distinct temporal statistics and sampling noise. Sharing early filters can cause negative transfer: motion branches benefit from high-frequency kernels to detect micro-acceleration and heading curvature, while weather branches favor lower-frequency, smoother filters to encode mesoscale conditions. Parallel TCNs decouple these priors, letting each branch set dilation schedules and receptive fields tailored to its domain. Our ablations (see Section 4.3) show that early fusion or serial stacks underperform, while parallel extraction followed by attention-based fusion yields higher F1 and better latency. Concretely, the motion TCN employs additive positional encoding to stabilize long-range order awareness under large dilations, whereas the weather TCN omits positional inflation to avoid amplifying low-SNR fluctuations. Post-extraction, concatenation and Attention-LSTM fuse asynchronous patterns: attention emphasizes segments where motion-weather interactions become discriminative (e.g., bursts aligned with favorable wind). The next subsections elaborate on the functions and features of each module in the MSIF-SSTR model.

Table 1
Format of the HN_BF dataset.

latitude	longitude	speed	course	windPower	WaveHeight	Visibility	…	Wind_8	id	label
20.38347	110.49097	25.85	199.1	2.5	0.5	12	…	0	1	0
20.37561	110.48781	26.71	200.5	2.5	0.5	12	…	0	1	0
…	…	…	…	…	…	…	…	…		
20.33628	110.47085	27.72	201.6	3.5	0.5	12	…	0	1	0
19.56155	108.90628	15.61	189.7	3	1	8	…	1	2	1
19.55562	108.90571	19.55	186.1	3	1	8	…	1	2	1
…	…	…	…	…	…	…	…	…	…	…

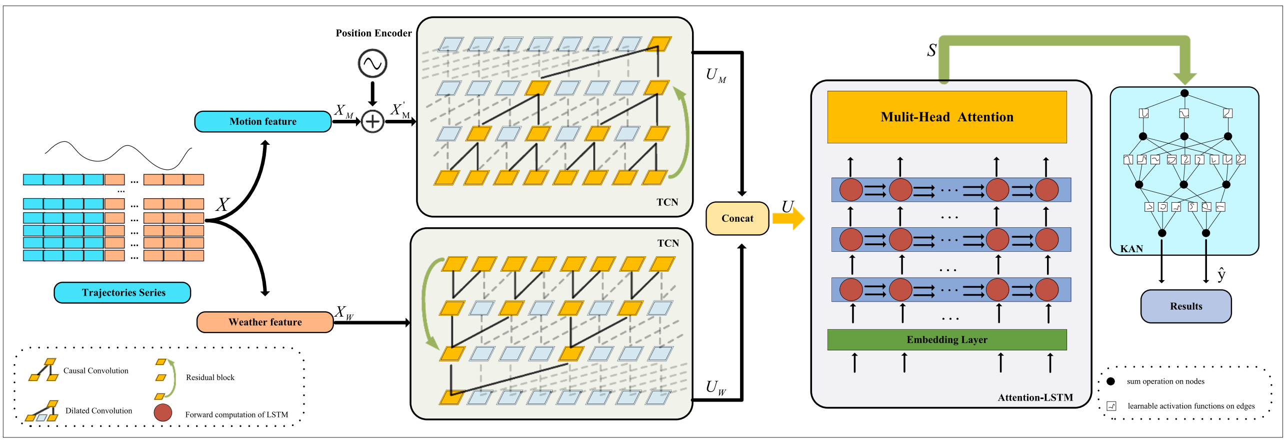


Fig. 2. Overall architecture of the MSIF-SSTR model.

3.3.1. Improved TCN network

TCN is a deep learning model specially designed to process temporal data,it uses a 1-dimensional convolutional structure to capture temporal features in sequences. Its parallel computation, and the advantage of being able to avoid the long-dependent gradient vanishing problem compensates for the shortcomings that exist in LSTM networks. The main features of TCN are causal convolution and dilated convolution. Causal convolution guarantees causality and the model relies only on current and past inputs when learning features from time-series data without leaking future information. When extracting trajectory sequence features, the dilated convolution design of TCN shows significant advantages, which effectively expands the receptive field and helps the model capture long time dependencies with fewer layers. Meanwhile, the residual connection mechanism can jump connections between multiple convolutional layers, mitigating the problem of gradient vanishing and enabling the model to learn more complex temporal patterns of the trajectory sequence.

We believe that the spatio-temporal information embedded in ship trajectory data is a key feature of its motion process, and at the same time, the order in which the trajectory data are arranged implies important temporal attributes. Although TCN is able to capture both local and global temporal features through the convolutional receptive domain, it does not have an explicit characterization of each position of the sequence, and thus the ability to perceive the order is weakened when dealing with long sequences, especially when it is necessary to accurately understand the global temporal relationships of the trajectory sequence. Therefore, the present model ensures that the characterization of each time step is not only limited to the neighboring time steps within the convolution window by introducing a new positional encoding (Foumani et al., 2024), the mathematical expression of which is shown in Eq. 4 and Eq. 5:

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{pos}}}} * \frac{d_{pos}}{\max\_\ len}\right) \tag{4}$$

$$\mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{pos}}}} * \frac{d_{pos}}{\max\_\ len}\right) \tag{5}$$

where pos represents the position of the word vector, max_len represents the maximum sequence length among all trajectory sequences and d_{pos} represents the embedding dimension of the position encoding. $\frac{d_{pos}}{max\_len}$ is the scaling factor for the location encoding, which is used to adjust the frequency of the location encoding. Although the position coding adopted in this model is similar to the position coding based on sine and cosine functions in the Transformer model, there are still some differences in details. Transformer's location coding covers the entire range of sequence lengths, the encoding scale depends on sequence length and embedding dimension, and is encoded at a fixed frequency. In contrast, the position encoding adopted in this model introduces an additional scaling factor $\frac{d_{pos}}{max\_len}$, so that the frequency of the position encoding can be dynamically adjusted according to the sequence length and embedding dimension. This design makes the scale of the position code match the numerical range of the input features, so as to better adapt to the trajectory data of different lengths, and improve the generalization ability of the model to the trajectory data of different lengths. In addition, this position coding design is more suitable for TCN model based on local convolutional receptive field. By dynamically adjusting the frequency, this position coding can make the coding frequency match the size of the receptive field of the convolutional kernel, thus enhancing the model's ability to capture local and global motion patterns. Specifically, when the features X_M and X_W of the trajectory sequence X are input to the independent TCN modules respectively, X_M will be firstly projected to a higher d_{pos} dimension, and then a position encoding operation will be performed to obtain the vector representation X'_M, and after that the operation steps of X'_M and X_W are the same, and both of them will be passed into the TCN for feature extraction. Taking X_W as an example, the TCN performs temporal convolution of the input sequence by means of a multilayer residual convolution block, where the overall operation of the residual convolution block can be expressed as:

$$\begin{aligned}\mathrm{Conv}_{res,i} &= f(W_i * f(W_{i-1} * \dots f(W_1 * X_W + b_1) \\ &\quad \dots + b_{i-1}) + b_i) + downsample(X_W)\end{aligned} \tag{6}$$

where W is the convolution kernel, b is the bias, i denotes the index of the current residual convolution block, and $f()$ is the activation function,in this paper, the ELU activation function is used. Dilated convolution is used in TCN to expand the receptive field by setting the expansion factor, which is able to capture the dependency between time steps. The residual connection structure is specified by adding the input and the output after multiple convolution operations. These structures ensure that information can be passed efficiently through the deeper network.

The final TCN module outputs are the feature mappings U_M and U_W corresponding to X_M and X_W, and U_M and U_W are also the output of the last residual convolution block.

3.3.2. Attention-LSTM network

LSTM is a recurrent neural network (RNN) specifically designed to process and predict time series data. It manages the information flow using three mechanisms: input gates, forgetting gates, and output gates to efficiently capture the dependencies between time series data. During forward computation, LSTM accepts three inputs: the input value at the current moment, the output value at the previous moment, and the unit state at the previous moment. Correspondingly, its output includes the output value at the current moment and the cell state at the current moment.

Attention mechanism is a human visual neural system-based mechanism in deep learning, which can help the model to focus on the important parts of the sequence and reduce the interference of irrelevant information by calculating the corresponding weights of the hidden layer vectors of the input sequence at different time steps. The multi-head attention mechanism is divided into multiple attention heads, each of which independently learns different features or patterns in the sequence. This parallel processing allows the model to learn more dimensional information than just a single sequence relationship. Meanwhile LSTM ignores some distant dependencies when processing long sequences, while the multi-head attention mechanism allows the model to focus on different parts of the sequence in different subspaces, which enhances the ability to capture global information.

The LSTM module with multi-head attention mechanism incorporates the long-term dependency modeling capability of LSTM and the global information capturing capability of multi-head attention mechanism. First, the input to the LSTM is the vector $U = \{u_1, u_2, \dots, u_T\} = concat(U_M, U_W)$ combined from the outputs of the two TCN modules, in which T is the length of the merged vector and also the length of the trajectory sequence. At each time step t, the LSTM will combine the hidden state h_{t-1} and cell state c_{t-1} of the previous time step and the inputs u_t of the current time step to compute the hidden state h_t and cell state c_t of the current time step. This process can be represented by the following recursive formula:

$$h_t, c_t = \mathrm{LSTM}(u_t, h_{t-1}, c_{t-1}) \tag{7}$$

By traversing the entire sequence of trajectories, the LSTM will generate a hidden state sequence $H = \{h_1, h_2, \dots, h_T\}$, where $H \in R^{T*d_h}$ and d_h denotes the dimension of the LSTM hidden layer. Hidden state sequence H will be used as an input to the multi-head attention mechanism.

In the multi-head attention mechanism, the output H from the LSTM layer undergoes a linear transformation to produce the query vector Q, the key vector K, and the value vector V. This process can be represented by the following equation:

$$Q = W_Q H, K = W_K H, V = W_V H \tag{8}$$

where W_Q, W_K, W_V are all linear transformation matrices used to project the input dimensions into the space of queries, keys and values. Next, the attention weight α_i is computed for each head, which is done through the dot product attention mechanism, i.e., the similarity is computed using the query vector Q and the key vector K, and normalized by the softmax function, as shown in Eq. 9.

$$\boldsymbol{\alpha}_i = \mathrm{Softmax}\left(\frac{Q_i * K_i^T}{\sqrt{d_K}}\right) \tag{9}$$

where d_K is the dimension of the key vector. After obtaining the attention weight α_i, it is applied to the value vector V to compute the context vector $C_i = \alpha_i * V_i$ for each head. The key to the multi-head attention mechanism is to process multiple heads in parallel, where each head will compute the attention weights and context vectors independently. The context vectors of all heads are spliced together to form a richer representation of the attention context. The result of the splicing is passed through a linear transformation matrix to get the final multi-head attention output S , as shown in Eq. 10.

$$S = \mathrm{MultiHead}(Q, K, V) = W_O\left[C_1; C_2; \dots; C_H\right] \tag{10}$$

where W_O is the output linear transformation matrix used to combine information from different attention heads.

3.3.3. Kolmogorov-Arnold networks module

In general, for time series tasks, the final output layer is basically chosen to be the fully connected layer. However, the fully-connected layer only performs linear transformations on the input data, which is slightly insufficient in capturing complex nonlinear features, while the trajectory data is highly nonlinear and dynamic. KAN (Liu et al., 2024) is able to learn the complex features of the input data through curve fitting, has stronger nonlinear modeling ability, captures subtle pattern changes in the trajectory data, and is more suitable for dealing with variable trajectory sequences with complex behaviors.

The idea of KAN is mainly based on adaptive grid points and B-spline function to perform nonlinear feature mapping to achieve efficient nonlinear representation of input features for more flexible and nonlinear modeling. The whole KAN network consists of multiple layers of linear transformations and B-spline interpolation, and each layer performs nonlinear mapping of features once.

The input to the KAN network is the vector S. The final output vector of KAN is $\hat{y} \in R^2$ through multiple layers of KAN linear layers. The main operation of each KAN linear layer are shown in Eq. 11:

$$h^{(l+1)} = \Phi_l h^{(l)} = \begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix} h^{(l)} \tag{11}$$

where $h^{(l+1)}$and $h^{(l)}$ respectively represent the first l KAN linear transformation layer of output and input; $\boldsymbol{\phi}_l$ represents the function matrix of the KAN layer of the l layer; n_l indicates the number of nodes in the l layer; $\phi_{l,j,i}(\cdot)$ will represent the first l layer first i node output into the first $l+1$ layer first j node of the input; $\phi(\cdot)$ represents the residual activation function, which is calculated as follows:

$$\phi(x) = W * (\sigma(x) + B(x)) \tag{12}$$

where σ is the basic function, and SiLU function is used in this study. The existence of the basic function is to ensure the smoothness of the output. $B(x)$ stands for spline function; W stands for the weight matrix, and in fact, the existence of the weight matrix is superfluous because it can be absorbed into $\sigma(x)$ and $B(x)$. This factor is still included in order to better control the overall amplitude of the output.

The spline function $B(x)$ is generated by combining the control point $P_i$and the corresponding B-spline basis function $B_i(x)$ that needs to be continuously optimized during training. Its general form is:

$$B(x) = \sum_{i=0} P_i B_i(x) \tag{13}$$

$B(x)$ uses the B-spline basis function to interpolate the control points by weighted summation, so that the model can better fit the features of the trajectory sequence. The B-spline basis function defines the distribution of input features at each grid point. The recursion formula of its calculation method is shown as follows:

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \tag{14}$$

$$B_{i,0}(x) = \begin{cases} 1 & t_i \le x < t_{i+1} \\ 0 & otherwise \end{cases} \tag{15}$$

where t_i represents the first i grid point, and the grid point determines the segment position of the spline basis function; k is the spline order, which determines the complexity and smoothness of the polynomial on each interval; $B_{i,0}(x)$ represents the piecewise constant, which has a value of 1 when x is in the corresponding interval, and 0 otherwise. The final output of the MSIF-SSTR network $\hat{\mathbf{y}}$ is the output of the last KAN linear layer. The loss function formula for each batch is shown in Eq. 16:

$$\text{loss}\ (y, \hat{y}) = -\frac{1}{batch\_size} \sum_{i=1}^{batch\_size} y \log \hat{y} \tag{16}$$

By calculating the average cross-entropy between the predicted labels $\hat{y}$ and the true labels y within each batch as a loss function, this process involves only the computation of real vectors, thus improving the efficiency of the model without increasing the computational burden.

4. Experimental results and analysis

This chapter systematically evaluates the overall performance of the MSIF-SSTR model through the design and execution of comparative experiments, ablation studies, qualitative analysis, and field applications. We conducted empirical validation from four dimensions: accuracy, efficiency, module effectiveness, and practical adaptability. The results show that the model achieved an F1 score of 94.26% on the HN_BF dataset, significantly outperforming baseline models. Ablation studies further clarified the core contributions of position encoding, the attention mechanism, and the KAN module. Moreover, in real-world applications in the Qiongzhou Strait, the model effectively reduced manual review costs and demonstrated strong early warning capabilities, proving its substantial practical value.

4.1. Experimental setup

The experimental environment for this paper used an Intel(R) Core(TM) i7-7800X six-core CPU at 3.5GHz and an NVIDIA TITAN XP graphics processor. The implementation was performed on Windows 10 x64 bit operating system using the PyTorch deep learning framework.

The model hyperparameters are chosen as follows: for the TCN feature extraction module, the convolution size is 3, the number of output channels of the hidden layer of the residual network is set to [16,32,64], and the d_{pos} of the position encoding is 512, respectively. For the LSTM network, the dimensionality of the hidden layer of the LSTM is set to 64, the number of the LSTM layers is set to 1, the dimensionality of the hidden layer of the attention mechanism is set to 64, the number of the multi-heads' attention mechanism number of heads is 4. During model training, the learning rate is set to 0.0001, the batch size is 128, and the Dropout rate for all is set to 0.1. We apply class-weighted cross-entropy consistent with the dataset skew (see Section 3.2), and a stratified sampler to keep mini-batch balance. We observed improved recall of the minority class without degrading precision.

In order to avoid overfitting of the model during the training process, we systematically divided the dataset in our experiments. Specifically, we firstly disrupted the order of the trajectory samples by random number seeds, and then divided the dataset into training, validation and test sets in the ratio of 6:2:2. This division ensures that there is no data overlap between the training, validation and test sets. We choose to use Precision (P), Recall (R) and F1-Score (F1) as evaluation metrics to assess the performance of each model. They are calculated corresponding to Eq. 17, Eq. 18, and Eq. 19, respectively, and higher values of the three metrics indicate superior model performance.

$$Precision \ = \ \frac{TP}{TP + FP} \tag{17}$$

$$Recall \ = \ \frac{TP}{TP + FN} \tag{18}$$

$$F1 - Score = \ 2 * \frac{Precision * Recall}{Precision + Recall} \tag{19}$$

where TP and FP denote truly detected and misdetected anomalies, respectively, and FN refers to misclassified normal samples.Precision measures the proportion of samples predicted to be positive that are actually positive. Recall, on the other hand, indicates the proportion of the number of actual positive samples out of the samples predicted to be positive out of the total number of positive samples.The F1-score is a weighted reconciled average of precision and recall, and is used to assess the performance of the two together. We chose to use precision rather than accuracy because in many cases precision is more appropriate for evaluating machine learning models, especially in the face of uneven numbers of positive and negative samples, where accuracy can lead to misleading results. (Wu et al., 2023b). In addition to the above evaluation metrics, we introduced the number of parameters and average time/sample as other evaluation metrics to demonstrate the complexity of the model. The 'Parameters' refers to the total number of parameters required to train the model, while the 'average Time/Sample' indicates the average time (in seconds) required to test each sample.

To reduce the chance of the experimental results, we trained each model five times independently with different random seeds throughout the experiment. Subsequently, we tested the models separately and calculated the mean and standard deviation of the results of the five tests, which are presented in the format of 'Mean ± Standard Deviation'.

4.2. Comparative experiment

To assess the performance of the MSIF-SSTR model in detecting suspected smuggling behavior in maritime trajectories, we compared it with the following deep learning (DL) models:

- ResNet (He et al., 2016): ResNet is one of the landmark models in the history of deep learning (Ismail Fawaz et al., 2019), and its proposed residual join effectively mitigates the gradient vanishing problem, making deep neural networks easier to train and improving model performance.
- LSTM (Hochreiter, 1997): LSTM network is able to capture temporal dependencies in time series through its built-in memory units and gating mechanisms, and shows good adaptability to time series data in a wide range of temporal tasks, and is one of the backbone networks of our model.
- TCN (Bai et al., 2018): TCN efficiently extracts patterns in sequences through 1D convolution and extended causal convolution with the advantage of parallelization, especially suitable for capturing feature information in time series, and is also one of the backbone networks of our model.
- Dlinear (Zeng et al., 2023): DLinear is a model that focuses on long time series, modelling trends and seasonality by performing segmented linear transformations of the input series, improving the computational efficiency and interpretability of the model while avoiding complex deep structure.
- LightTS (Zhang et al., 2022):LightTS uses continuous and interval sampling techniques to capture short-term and long-term patterns of time series, effectively reducing information redundancy. At the same time, it replaces complex deep networks with lightweight convolutional structures, which is suitable for resource-constrained environments and low-latency time series tasks.

Table 2
Performance of each model on the HN_BF dataset (MEAN ± SD (%)).

Methods	P	R	F1	Parameters	Avg Time/sample
ResNet(He et al., 2016)	92.7858 ± 0.2872	92.8243 ± 0.2682	92.805 ± 0.2766	35376398	0.000745
LSTM(Hochreiter, 1997)	64.7102 ± 8.2578	74.6642 ± 0.3479	69.1004 ± 4.7533	213762	0.000144
TCN(Bai et al., 2018)	92.5615 ± 0.2374	92.4195 ± 0.1395	92.4903 ± 0.1673	**67186**	0.000255
Dlinear(Zeng et al., 2023)	86.1953 ± 0.7016	86.2557 ± 0.8192	86.2254 ± 0.754	198002	**0.000092**
LightTS(Zhang et al., 2022)	91.6071 ± 0.2931	91.4259 ± 0.2553	91.5163 ± 0.2445	121882	0.000246
PatchTST(Nie et al., 2023)	88.6843 ± 0.385	88.5556 ± 0.3935	88.6197 ± 0.3629	674434	0.000903
TimesNet(Wu et al., 2023a)	93.6156 ± 0.25	93.6063 ± 0.2315	93.6109 ± 0.233	37579906	0.028795
LSTM-FCN(Dan et al., 2022)	92.1532 ± 0.3124	92.0875 ± 0.2891	92.1203 ± 0.2986	286534	0.000318
MFGTN(Gu et al., 2024)	93.1265 ± 0.2748	93.0987 ± 0.2563	93.1126 ± 0.2615	894258	0.001245
MSIF-SSTR(Ours)	**94.3008 ± 0.394**	**94.2226 ± 0.4664**	**94.2616 ± 0.4276**	127908	0.000166

- PatchTST (Nie et al., 2023): PatchTST is a Transformer model for long time series, which divides the time series data into a series of patches and inputs them into the Transformer structure to capture long time dependencies. This approach not only enhances the expressive power of the model, but also maintains the localisation of the time series.
- TimesNet (Wu et al., 2023a): TimesNet finds different periods in the data by Fast Fourier Transform, transforms the original one-dimensional time series into a two-dimensional space to learn the temporal variations within a period versus between periods, and performs unified modelling to better capture the periodicity and trend of the data.

The specific performance of each model on the dataset HN_BF is shown in Table 2, bold text in the table indicates the best results.

Table 2 presents a comprehensive comparison of the performance of MSIF-SSTR (Ours) with nine state-of-the-art models (including the newly added LSTM-FCN and MFGTN) on the HN_BF dataset, evaluated by precision (P), recall (R), F1-score, number of parameters, and average inference time per sample.In terms of core recognition metrics, MSIF-SSTR achieves the highest performance across all indicators: its precision reaches 94.3008 ± 0.394%, recall is 94.2226 ± 0.4664%, and F1-score hits 94.2616 ± 0.4276%. This outperforms the second-ranked TimesNet by 0.69% in F1-score, and also exceeds the two newly added models-MFGTN and LSTM-FCN-by 1.15% and 2.14% in F1-score respectively. This indicates that our model more accurately identifies suspected smuggling trajectories while reducing both false positives and false negatives. Specifically, MFGTN (F1 = 93.1126 ± 0.2615%) relies on graph transformation to model trajectory correlations, but lacks effective fusion of multi-source features; LSTM-FCN (F1 = 92.1203 ± 0.2986%) fuses LSTM's temporal modeling and FCN's local feature extraction, yet fails to capture long-range trajectory dependencies. Compared to ResNet and TCN (which focus on spatial feature extraction and temporal sequence modeling respectively), MSIF-SSTR still shows advantages of 1.46% and 1.77% in F1-score, further demonstrating the effectiveness of integrating multi-source features and attention mechanisms for complex trajectory discrimination.

Notably, MSIF-SSTR maintains superior efficiency while ensuring high accuracy. With 127,908 parameters, it is significantly lighter than not only deep models like TimesNet (37.58 million parameters) and ResNet (35.38 million parameters) (reducing parameter scale by 99.66% and 99.64% respectively), but also the newly added MFGTN (894,258 parameters) and LSTM-FCN (286,534 parameters)-cutting parameter counts by 85.7% and 55.3% respectively. In terms of inference speed, its average time per sample is 0.000166 seconds: this is 173 times faster than TimesNet (0.028795 seconds), 4.5 times faster than ResNet (0.000745 seconds), 7.5 times faster than MFGTN (0.001245 seconds), and 1.9 times faster than LSTM-FCN (0.000318 seconds). Although Dlinear achieves the fastest inference speed (0.000092 seconds), its F1-score (86.2254 ± 0.754%) is 8.04% lower than MSIF-SSTR. This highlights the trade-off between speed and accuracy that our model effectively balances-outperforming both lightweight models (in accuracy) and high-performance models (in efficiency), including the newly added LSTM-FCN and MFGTN.

Overall, the experimental results validate that MSIF-SSTR not only excels in recognizing suspected smuggling trajectories but also maintains high computational efficiency, making it more suitable for practical maritime law enforcement scenarios with real-time and resource-constrained requirements.

4.3. Ablation study

In order to further evaluate the effectiveness of each component in the MSIF-SSTR network, several sets of performance analysis experiments are designed in this paper, and the experimental results are shown in Table 3. During the experiment, only the components were changed between the models, and all other parts were consistent.

From the experimental results, we found that the introduction of location coding into the TCN network not only improved the evaluation index, but also significantly accelerated the testing speed of the model, and even accelerated the training speed to a certain extent during the training process, which was beyond our initial expectation.

After the introduction of the Attention mechanism in LSTM network, all evaluation indicators are significantly improved, which is the largest improvement among all single modules. Although the introduction of the Attention mechanism increases the number of parameters in the model, its contribution to improving performance is also the most obvious, making the model perform better in the suspected smuggling identification task.

The introduction of the KAN network instead of the traditional fully connected network also resulted in a significant improvement in all evaluation metrics. Although the increase in the number of parameters is relatively small, we notice an increase in the testing speed of the model. In addition, the model convergence speed was also significantly slower in the presence of KAN network during the training process. This suggests that KAN networks are more difficult to train and converge compared to fully connected networks, and thus the relationship between performance improvement and computational overhead needs to be weighed when designing the network structure.

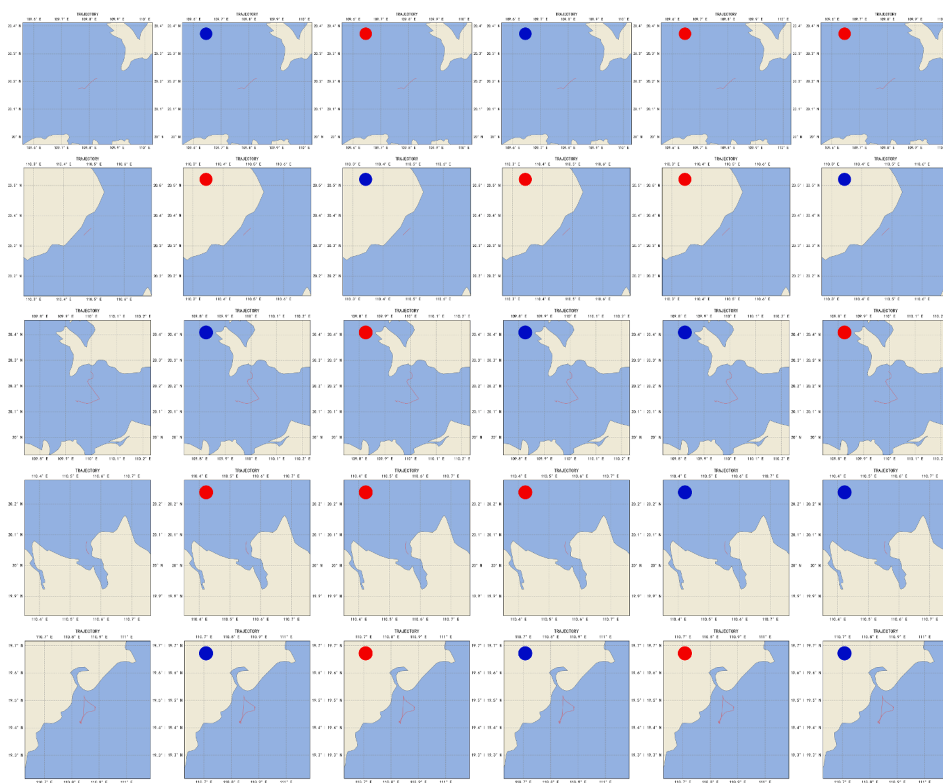
Beyond the default attention fusion, we evaluated: (i) early feature stacking before TCN; (ii) gated additive fusion; (iii) cross-attention between branches. Early stacking reduced robustness under weather noise (F1 drop ~0.9%), suggesting interference in shared filters. Gated fusion improved precision but slightly reduced recall (F1 drop ~0.3%). Cross-attention matched our Attention-LSTM but added latency. Our chosen design balances accuracy and efficiency for real-time screening.

4.4. Qualitative experiment

In order to better demonstrate the MSIF-SSTR network's ability to identify suspected smuggling tracks, we plotted three representative abnormal tracks and four normal tracks respectively from the data set, as

Table 3
Influence of different components on MSIF-SSTR model on HN_BF dataset (MEAN ± SD (%)).

Pos	Atten	KAN	P	R	F1	Parameters	Avg Time/sample
✗	✗	✗	79.4728 ± 8.229	79.9448 ± 5.9974	79.6595 ± 7.0634	**105506**	0.000061
	✗	✗	85.2764 ± 7.2238	83.6062 ± 8.2616	84.4139 ± 7.6713	110118	**0.000055**
✗		✗	89.7509 ± 8.501	89.6388 ± 8.4482	89.696 ± 8.4752	122146	0.000079
✗	✗		88.1051 ± 0.3206	87.8565 ± 0.4691	87.9805 ± 0.3829	106656	0.000139
		✗	92.5107 ± 0.262	92.4011 ± 0.3479	92.4558 ± 0.3043	126758	0.000091
	✗		92.0864 ± 0.271	92.0699 ± 0.279	92.0782 ± 0.27	111268	0.000137
✗			92.9595 ± 0.5279	92.8979 ± 0.5611	92.9287 ± 0.5422	123296	0.000232
			94.3008 ± 0.394	**94.2226 ± 0.4664**	**94.2616 ± 0.4276**	127908	0.000166



(a) Original (b) ResNet (c) LSTM (d) TCN (e) Dlinear (f) LightTS

Fig. 3. Visualization of the models. The blue circles indicate correct tests, and the red circles indicate incorrect tests.

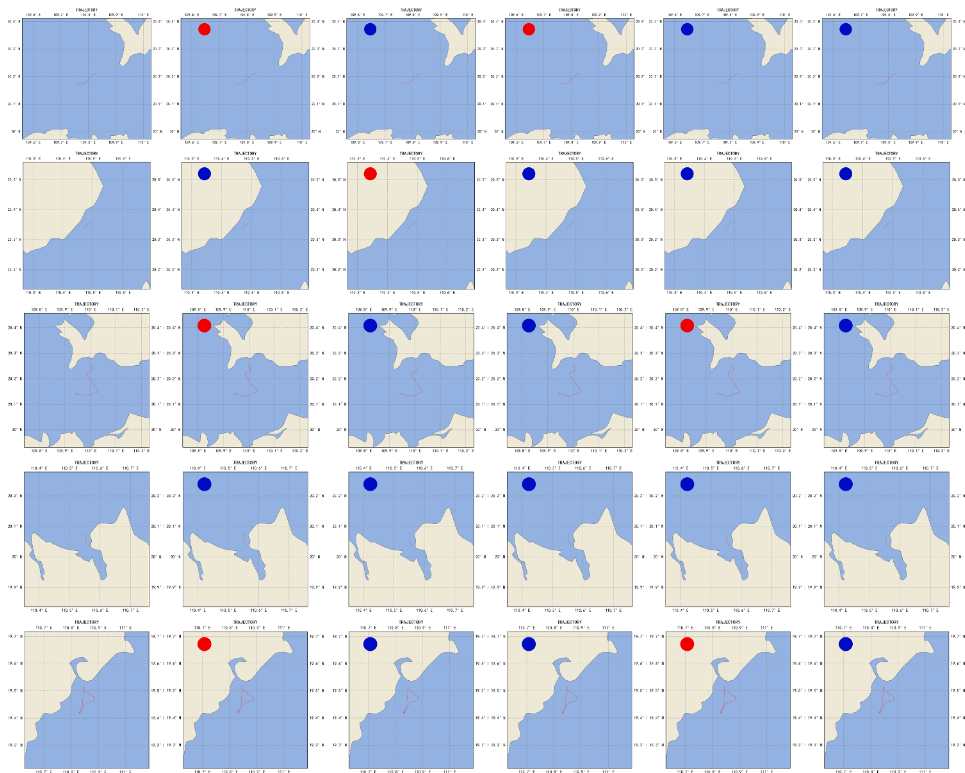
shown in Fig. 4(a). In these graphs, the first three represent abnormal trajectories and the last two represent normal trajectories. The trajectory of the fourth figure starts at the port, so it is a normal trajectory. The tracks of the fifth figure is detour many times and do not conform to the behavior pattern of smuggling tracks.

Subsequently, we input these five trajectories into each network for detection and replotted the detection results. The blue circles in the upper left corner of the figure indicate the correct detection results, while the red circles indicate the wrong detection results, as shown in Fig. 3(b)-4(f). From the detection results, the MSIF-SSTR network successfully identifies the trajectory of the suspected “Quick Smuggler” smuggler, while the other models have different degrees of false alarms and omissions.

In addition, we believe that the performance of a model in real-world applications is one of the most important criteria for evaluating its effectiveness. For this reason, we deploy the MSIF-SSTR model directly into real-world application scenarios to verify its practicality and reliability in real-world environments. Taking the trajectory data near Qiongzhou Strait on July 9, 2024 as an example, there were a total of 145 abnormal trajectories detected at night with high speed and no AIS equipment turned on based on the rules. After secondary screening by MSIF-SSTR, 41 abnormal tracks were found to be suspected of smuggling and 104 were found to be non-suspected. All tracks were then checked, and the total number of false positives from MSIF-SSTR was 21. Compared with the manual screening method, the MSIF-SSTR model greatly saves the consumption of human resources. As shown in the Fig. 5, the first three pictures are the abnormal trajectory segments suspected of smuggling discovered by our model, and the last three pictures visually demonstrate the practical effectiveness of this model.

5. Discussions

Sections 4.2 to 4.4 of this paper detail the advantages of the MSIF-SSTR model over some of the most recent and typical networks, as well as analysing the roles and contributions of the individual MSIF-SSTR modules in the overall framework. In the comparison experiment section, MSIF-SSTR achieves the best evaluation metrics, with about 0.6% improvement in evaluation metrics compared to TimesNet and ResNet, which have the second and third best evaluation metrics, but had great advantage in terms of the number of training parameters and the time re-



(a) Original (b) PatchTST (c) TimesNet (d) LSTM-FCN (e) MFGTN (f) **Ours**

Fig. 4. Visualization of the models (Continuation of Fig. 3). The blue circles indicate correct tests, and the red circles indicate incorrect tests.

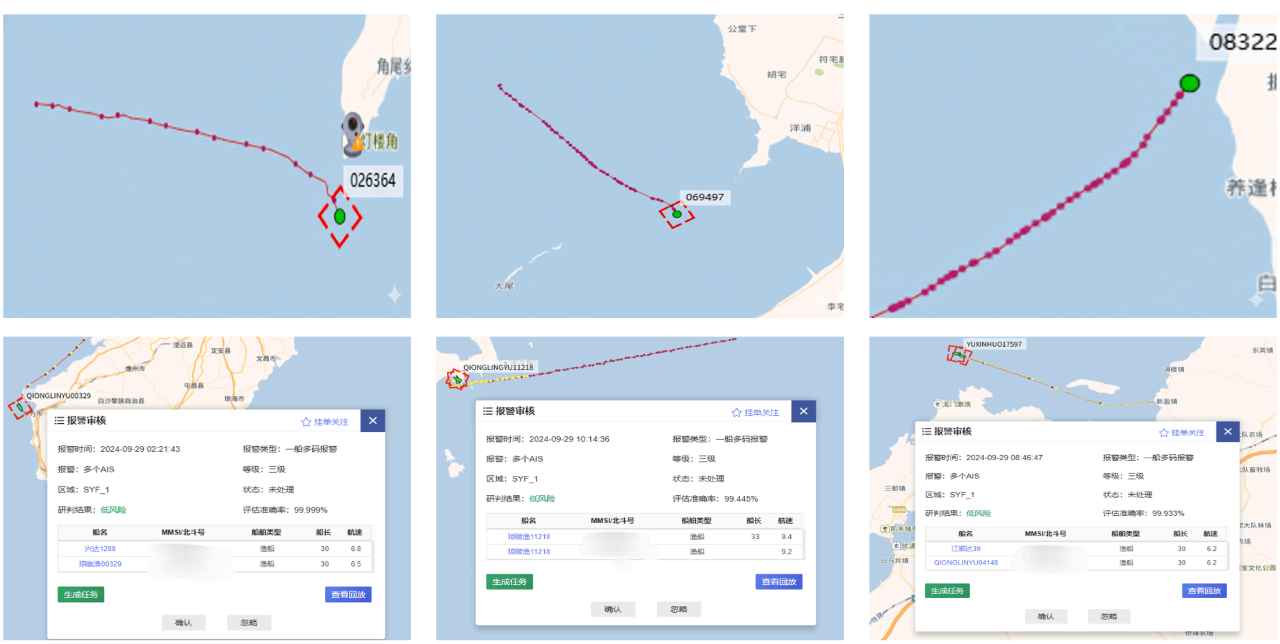


Fig. 5. Suspected smuggling trajectories detected by the MSIF-SSTR model in real scenarios and its practical combat effectiveness.

quired for testing. This means that MSIF-SSTR is able to accomplish the task at a much lower computational cost while ensuring high precision. This ability to balance performance and computational cost fully demonstrates the overall superiority of MSIF-SSTR in the suspected smuggling track identification task.

In the ablation study, after introducing position coding to the TCN network, we unexpectedly found that the testing speed of the model was significantly accelerated, and even the training speed was accelerated during the training process. This effect was not foreseen at the beginning of our design, and through analysis, this may be due to the fact that position coding enhances the TCN's ability to capture temporal information, which makes it easier for the TCN to understand the relationship between the time points in the trajectory sequence, and thus improves the efficiency of feature extraction. It is also possible that the addition of positional coding reduces the computational burden on the TCN in inferring the temporal relationships between the preceding and following texts of the trajectory, allowing the training to converge more quickly and reducing the need for complex computation.

Still, MSIF-SSTR has some limitations and areas that can be improved, and we will discuss them in two parts, a model-based discussion, and a reality-level based discussion.

At the model level, this study has the following limitations. First, the tandem structure of TCN and LSTM may cause delay in information transfer, especially when the sequence length is long, which may affect the classification performance. Secondly, although the splicing activity of weather features and motion features can integrate multidimensional information, this simple splicing may lead to interference between features, making it difficult for the model to effectively distinguish the importance of different features. Therefore the following aspects deserve further research: firstly, exploring and adopting more efficient network architectures to reduce computational complexity and improve information transfer efficiency. Second, explore the use of more sophisticated fusion methods, such as weighted fusion or attention mechanisms, to better capture the relationships between different features and improve the representation of information. Pure attention layers attenuate simple splicing interference but cannot by themselves resolve modality-specific noise profiles and asynchronous cues. The parallel-then-fuse pipeline ensures modality-specialized encoding prior to interaction, while attention provides context-dependent weighting at sequence segments where coupling becomes informative. Future work may explore hierarchical fusion-coarse weather gating at low temporal resolution and fine motion attention-to further enhance robustness without inflating runtime. At the level of real-world applications, the current work in this paper has the following limitations. Identifying smuggling activities is a difficult and complex process, from the discovery of suspicious ships to the final confirmation of the smuggling process can be roughly divided into five steps: through the rules of the discovery of suspicious high-speed ship trajectory; manually screen whether the suspected smuggling behavior; continuous tracking, and by means of photoelectric cameras and other means of observing whether there are abnormal connections, mooring and other behaviors; continuous tracking of high-speed ships and connecting ships; dispatch of marine police to conduct seizures of smuggling vessels. The overall process is long, during which the continuous tracking time needs to last for hours or even months. The research in this paper is currently focusing on the optimization of the second step, intelligent judgment under the premise of rule-based determination, to a certain extent, to reduce the dependence on human experience, while enhancing the early warning of the fight against smuggling. Therefore, the following aspects deserve further study: First, the linkage mechanism of photoelectric camera identification is added to the “Quick Smuggler” discrimination model, so that the camera can be called after detecting the suspected “Quick Smuggler” trajectory for subsequent judgment of abnormal behavior of connecting. Secondly, correlating the ship trajectory with the length of the ship captured by the radar can more accurately capture the characteristics of the smuggling ship, but in this study, this method could not be realized because the accuracy of the radar itself is difficult to support this work. Based on the above outlook, we will continue to work on the recognition of abnormal ship behavior by fusing different modal information.

6. Conclusions

In this study, a deep learning model MSIF-SSTR based on multi-source information fusion is proposed, and the “Quick Smuggler” identification dataset HN_BF is constructed using real radar data and meteorological data near the Qiongzhou Strait in China. The MSIF-SSTR model uses independent TCN networks to extract motion features and weather features from the trajectory sequence separately. Among them, we propose a new TCN network based on additive position encoding for extracting motion features. These features are then fused and characterized by an LSTM network with a multi-head attention mechanism, while the KAN module is introduced to classify the complex trajectory sequences. The evaluation indexes on the HN_BF dataset are as high as 94.2% or more, and there is also a greater advantage in the training parameters and the time required for the test. The excellent experimental results and practical effects demonstrate that the structural framework of the MSIF-SSTR model can adequately extract the depth features of smuggled ship trajectories.

However, it should be objectively noted that the current model still has limitations: the daKan module, as the core nonlinear modeling component of MSIF-SSTR, relies on parameter learning based on B-spline functions, which requires a large number of data iterations to achieve stable convergence. In online learning scenarios, new samples are often input in the form of 'small batches and fragmented data,' with the amount of incremental training data in a single batch being only 1/50 of the offline dataset. This small-batch data input prevents the B-spline control point parameters of the KAN module from being fully optimized, resulting in a phenomenon of 'parameter oscillation'.

In future work, we intend to conduct further research on the issues mentioned in Section 4.5, especially based on the real-world application level. In addition, considering the dynamic nature of smuggling behavior, subsequent research will consider how to introduce an online learning mechanism so that the model can continuously learn the data distribution and movement patterns of new trajectories. Our ultimate goal is to develop a more flexible, efficient and reliable smuggling trajectory recognition system to provide stronger support for maritime law enforcement.

CRediT authorship contribution statement

Zhuhua Hu: Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization; **Yifeng Sun:** Writing – review & editing, Software, Investigation, Formal analysis, Data curation; **Yaochi Zhao:** Supervision, Resources, Investigation, Formal analysis, Conceptualization; **Wei Wu:** Supervision, Investigation, Visualization, Data curation; **Lingkai Kong:** Supervision, Resources, Conceptualization; **Keli Chen:** Investigation, Software, Data curation.

Data availability

The authors do not have permission to share data.

Declaration of competing interest

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

Acknowledgements

This research was supported by the National Natural Science Foundation of China (62361024 and 62161010), the Key Research and Development Project of Hainan Province (ZDYF2022GXJS348 and ZDYF2024GXJS021), the National Key Research and Development Program of China (2022YFD2400504), and the Hainan University Postgraduate Innovative Talent Cultivation Project (HDJXAL2024Y0007, HDZBJC2024Y0008).

References

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271.

Dan, H., Xin, M., Shuai, L., & Lining, X. (2022). Application of a parallel lstm-fcn model in ship track prediction [j/OL]. *Control and Decision*, (pp. 1–7).
Foumani, N. M., Tan, C. W., Webb, G. I., & Salehi, M. (2024). Improving position encoding of transformers for multivariate time series classification. *Data Mining and Knowledge Discovery*, *38*(1), 22–48.
Gu, Y., Hu, Z., Zhao, Y., Liao, J., & Zhang, W. (2024). Mfgtn: A multi-modal fast gated transformer for identifying single trawl marine fishing vessel. *Ocean Engineering*, *303*, 117711.
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
Hochreiter, S. (1997). Long short-term memory. *Neural Computation MIT-Press*.
Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: A review. *Data Mining and Knowledge Discovery*, *33*(4), 917–963.
Junfei Chen, W. W. Y. Z., Hu, Z., & Huang, B. (2024). Lkpf-yolo: A small target ship detection method for marine wide-area remote sensing images. *Ocean Engineering*.
Kazemi, S., Abghari, S., Lavesson, N., Johnson, H., & Ryman, P. (2013). Open data for anomaly detection in maritime surveillance. *Expert Systems with Applications*, *40*(14), 5719–5729.
Kowalska, K., & Peel, L. (2012). Maritime anomaly detection using gaussian process active learning. In *2012 15th International Conference on Information Fusion* (pp. 1164–1171). IEEE.
Laxhammar, R. (2008). Anomaly detection for sea surveillance. In *2008 11th International Conference on Information Fusion* (pp. 1–8). IEEE.
Lei, B., & Mingchao, D. (2018). A distance-based trajectory outlier detection method on maritime traffic data. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)* (pp. 340–343). IEEE.
Liang, Y., & Zhang, H. (2020). Ship track prediction based on AIS data and PSO optimized LSTM network. *Int. Core Journal Engineering*, *6*(5), 23–33.
Liao, J., Zhao, Y., Xia, J., Gu, Y., Hu, Z., & Wu, W. (2024). Dynamic-equalized-loss based learning framework for identifying the behavior of pair-trawlers. In *International Conference on Intelligent Computing* (pp. 337–349). Springer.
Lihang Chen, J. C., Hu, Z., & Sun, Y. (2025). Sviadf: Small vessel identification and anomaly detection based on wide-area remote sensing imagery and ais data fusion. *Remote sensing*.
Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2024). Kan: Kolmogorov-arnold networks. arXiv:2404.19756.
Martineau, E., & Roy, J. (2011). Maritime anomaly detection: Domain introduction and review of selected literature.
Nie, Y., Nam, H. N., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*.
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.
Portnoy, L. (2000). Intrusion detection with unlabeled data using clustering. Ph.D. thesis. Columbia University.
Ren, Y., Zhao, J., Liu, W., Wang, S., & Wei, Y. (2019). Ship navigation behavior prediction based on AIS data and LSTM network. *J. Shanghai Marit. Univ*, *40*, 32–37.
Rong, H., Teixeira, A. P., & Soares, C. G. (2020). Data mining approach to shipping route characterization and anomaly detection based on AIS data. *Ocean Engineering*, *198*, 106936.
Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., & Pei, D. (2019). Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2828–2837).
Sun, Y., Zhao, Y., Hu, Z., Wu, W., Xia, J., & Wang, Y. (2024). Ssrlm: A self-supervised representation learning method for identifying one ship with multi-mmsi codes. *Ocean Engineering*, *312*, 119186.
Tang, H., Yin, Y., & Shen, H. (2022). A model for vessel trajectory prediction based on long short-term memory neural network. *Journal of Marine Engineering & Technology*, *21*(3), 136–145.
Terroso-Saenz, F., Valdes-Vela, M., & Skarmeta-Gomez, A. F. (2016). A complex event processing approach to detect abnormal behaviours in the marine environment. *Information Systems Frontiers*, *18*, 765–780.
Tianjiao Wei, Y. Z., Hu, Z., & Fan, X. (2025). Aildp: A research on ship number recognition technology for complex scenarios. *Complex & Intelligent Systems*.
U. N. C. on Trade, (2023). Development, review of maritime transport 2023. *Review of Maritime Transport*.
Vespe, M., Visentini, I., Bryan, K., & Braca, P. (2012). Unsupervised learning of maritime traffic patterns for anomaly detection.
Wang, Z. R., Zhao, K. Y., Cai, C., Ding, M. Z., & Wang, P. (2021). Analysis of abnormal ship behavior based on DBSCAN and iforest algorithm. *Ship Electron. Eng*, *41*, 89–94.
Wen, C.-H., Hsu, P.-Y., & Cheng, M.-S. (2017). Applying intelligent methods in detecting maritime smuggling. *Maritime Economics & Logistics*, *19*, 573–599.
Wen, C.-H., Hsu, P.-Y., Wang, C.-y., & Wu, T.-L. (2012). Identifying smuggling vessels with artificial neural network and logistics regression in criminal intelligence using vessels smuggling case data. In *Intelligent Information and Database Systems: 4th Asian Conference, ACIIDS 2012, Kaohsiung, Taiwan, March 19–21, 2012, Proceedings, part II 4* (pp. 539–548). Springer.
Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., & Long, M. (2023a). Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*.
Wu, W., Li, X., Hu, Z., & Liu, X. (2023b). Ship detection and recognition based on improved YOLOv7. *Comput. Mater. Contin*, *76*(1), 489–498.
Xiao, F., Ligteringen, H., Van Gulijk, C., & Ale, B. (2015). Comparison study on AIS data of ship traffic behavior. *Ocean Engineering*, *95*, 84–93.
Xiyu Fan, X., Hu, Z., & Huang, Z. (2024). Csdp-yolo: Class-imbalance small target ship detection algorithm for remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.
Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 11121–11128). (*vol. 37*).
Zhang, B., Hirayama, K., Ren, H., Wang, D., & Li, H. (2023). Ship anomalous behavior detection using clustering and deep recurrent neural network. *Journal of Marine Science and Engineering*, *11*(4), 763.
Zhang, S., & Tang, Q. (2015). Abnormal vessel behavior detection based on AIS data. *Artificial Intelligence and Robotics Research*, *4*(04), 23–31.
Zhang, T., Zhang, Y., Cao, W., Bian, J., Yi, X., Zheng, S., & Li, J. (2022). Less is more: Fast multivariate time series forecasting with light sampling-oriented MLP structures. arXiv:2207.01186.
Zhao, L., & Shi, G. (2019). Maritime anomaly detection using density-based clustering and recurrent neural network. *The Journal of Navigation*, *72*(4), 894–916.
Zhen, R., Jin, Y., Hu, Q., Shao, Z., & Nikitakos, N. (2017). Maritime anomaly detection within coastal waters based on vessel trajectory clustering and naïve bayes classifier. *The Journal of Navigation*, *70*(3), 648–670.