Ocean Engineering 303 (2024) 117711

Contents lists available at ScienceDirect

Ocean Engineering

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

Research paper

MFGTN: A multi-modal fast gated transformer for identifying single trawl marine fishing vessel

Yanming Gu [a,c], Zhuhua Hu [a,c,*], Yaochi Zhao [b], Jianglin Liao [b], Weidong Zhang [a]

[a] School of Information and Communication Engineering, Hainan University, Haikou, 570228, China
[b] School of Cyberspace Security, Hainan University, Haikou, 570228, China
[c] State Key Laboratory of Marine Resource Utilization in South China Sea, Hainan University, Haikou, 570228, China

ARTICLE INFO

Keywords:
Deep learning
Data fusion
Automatic identification system
Ship trajectory classification
Recurrence plot image

ABSTRACT

In order to achieve sustainable development of marine fishery resources, effective supervise of trawl fishing during forbidden fishing period is of great significance. This paper addresses the challenges of poor generalization and the lack of unstructured information in the precise identification of single trawler fishing behavior. We propose a Transformer network with multi-source information fusion processing (MFGTN), which accurately classifies fishing vessels as single trawl or non-single trawl vessels. Firstly, a private fishing dataset of single trawl behavior is constructed by integrating AIS data with radar data, named HaiNan_SingleTrawlVessel(HN_STV). Subsequently, as fused data lacks unstructured information, it undergoes transformation into trajectory point images and recurrence plot images to reveal the internal structure of the fused data. As such, a visual module is introduced to handle the trajectory point images and recurrence plot images as a branch. Simultaneously, the fused data are input into a Double-Tower Transformer with Dual-gate structures to extract information in different dimensions of the time series and feature space as two separate branches. The Fast Attention module replaces the traditional Attention module to improve network speed and reduce memory consumption. Ultimately, the output of the three branches are fused and controlled by a Dual-gate structure that can autonomously learn to determine the network output. Experimental results show that compared to the current best-performing methods, the method discussed herein on the HN_STV dataset has improved the accuracy, recall, precision, and F1-score performance indicators by 2.34%, 2.46%, 0.97%, and 1.39%, respectively. The AUC area on the ROC curve increased by 4%. In a public dataset including three fishing activities, the proposed method improved accuracy, recall, precision, and F1-score by 2.95%, 2.59%, 2.25%, and 2.70%, respectively, and the AUC area on the ROC curve increased by 3%. And in all experiments, our network incurs the lowest time cost. Therefore, the method proposed herein demonstrates its advanced performance.

1. Introduction

The marine fisheries are very valuable resource in the world, and in order to maintain their sustainability, Nations across the globe are increasingly intensifying the monitoring and enforcement of fisheries resources, aiming to combat illegal, unreported, and unregulated (lUU) fishing activities. This initiative is fundamentally geared towards ensuring that commercial fishing operations do not exceed the sustainable yield of marine resources (Tursi et al., 2015). China is one of the countries with the largest fishing population and fishing volume in the world, and the output of Marine fishing vessels and Marine fishing ranks among the top in the world all year round. Hainan Island, China's second largest island, has a curved coastline about 1823 kilometers long. Fishery is the mainstay of Hainan Island's economic and industrial development. According to statistics (Hu et al., 2021), there are a total of 29,537 unlicensed vessels along the coast of the province's 14 cities and counties. The large number of Unlicensed vessels makes supervision extremely difficult. Moreover, illegal activities such as illegal fishing often occur when criminals exploit these unlicensed vessels. According to reports, during the summer fishing moratorium in the South China Sea in 2023, the Coast Guard Bureau seized a total of 32 fishing cases, 93 illegal fishing vessels, and caught 4163.7 kilograms of fish, which greatly damaged the sustainability of Marine resources. At present, there are still issues to be addressed regarding the inability of certain methods to accurately identify illegal activities involving unlicensed vessels, as well as the simplicity of existing warning model structures, unscientific parameter configurations, and high

* Corresponding author at: State Key Laboratory of Marine Resource Utilization in South China Sea, Hainan University, Haikou, 570228, China.
E-mail address: eagler_hu@hainanu.edu.cn (Z. Hu).

https://doi.org/10.1016/j.oceaneng.2024.117711
Received 5 January 2024; Received in revised form 27 March 2024; Accepted 28 March 2024
Available online 9 April 2024

false alarm rates. Furthermore, the detection of abnormal behaviors currently relies mainly on manual identification, which fails to meet the requirements for real-time automatic recognition. However, in recent years, the emergence of deep learning technology, especially large-scale models, with their high parameter count and strong ability to mine potential information from data, has shown promise in addressing the aforementioned challenges. Consequently, using ship trajectory data and intelligently detecting single trawl operations or other anomalous behaviors of fishing vessels during forbidden fishing period or regulated periods has become a crucial issue. This issue has attracted researchers to apply various statistical and machine learning solutions (Riveiro et al., 2018).

Statistical techniques such as Extended Kalman Filtering and Particle Filtering have been employed for reconstructing ship trajectories (Perera et al., 2012). Bayesian networks have been utilized for handling missing values (anomalies) in datasets (Hruschka et al., 2007), aiding in prediction and classification. Zhang et al. (2023) proposes utilizing the minimum description length criterion to extract features from ship trajectory data provided by the automatic identification system (AIS). Additionally, the dynamic time warping trajectory similarity measurement algorithm is employed to optimize the density-based spatial clustering of applications with noise algorithm. Zhen et al. (2017) proposed an approach combining ship trajectory clustering with a Naive Bayesian classifier for detecting abnormal vessel behavior. On the other hand, machine learning (ML) algorithms, including Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Long Short-Term Memory (LSTM), generally outperform statistical methods in classification problems. Recently, Mazzarella et al. (2017) introduced an SVM-based anomaly detection framework that considers both AIS data and received signal strength for AIS Out-of-Service (OOS) anomalies. Sánchez Pedroche et al. (2020) proposed a trajectory-based AIS vessel classification architecture that treats vessels as binary classification into fishing and non-fishing based on ship trajectories. Feng et al. (2019) utilized Vessel Monitoring System (VMS) data from shrimp trawl vessels in the range of 25-35N, 120-130E as parameters for a neural network model to identify trawl fishing vessels, etc. For a more detailed research roadmap, please refer to the illustration provided in Section 2.

Thus, based on recent research trends, two key issues are summarized as follows:

- The statistical methods used to address the challenges associated with the classification of fishing vessel trajectories or the detection of anomalous behavior often require some artificial parameters, and different parameters cannot be well generalize to other datasets. Meanwhile, a detection method is typically tailored to a specific dataset, limiting its applicability across diverse datasets.
- Most existing networks overlook the unstructured information corresponding to the fused data, such as converting structured information like fused data into visual graphics to assist the network in classification.

To overcome these issues, we propose a fast gate-level Transformer for multi-modal fishing data, named MFGTN. The main contributions of this paper are as follows:

- The HN_STV dataset is constructed, which extracts ship trajectories from the surrounding sea area with Hainan Province as the center, and effectively integrates the ship's AIS data and radar data to create temporal fishing fusion sequences (TFFS) data.
- To address the first issue, a Double-Tower Transformer network is proposed to analyze and process the information in different dimensions of the time series and feature space of TFFS data. The traditional Attention module is modified to a Fast Attention module to reduce training time and memory consumption for large model.
- For the second issue, the Fusion former visual module is introduced to process the unstructured information of input data, including trajectory point images and recurrence plot images.
- A double-layer learnable gate structure is proposed to continuously optimize the weights of different information sources according to the training results of neural networks, so that the network can independently choose the weight ratio of different information sources.

2. Related work

Most existing studies primarily focused on proposing research methods for the targeted identification of ship anomaly behavior detection within their respective domains, yet research in the field of identifying single trawl fishing vessels is limited. With reference to the literature of Wang et al. (2023), this study classifies the research in the field of ship abnormal behavior detection into five categories in chronological order to elucidate the relevance and continuity of methods in this domain: statistical analysis methods, knowledge-based and data-driven integrated methods, traditional machine learning algorithms, deep learning methods, and multi-source data fusion methods. The temporal progression is illustrated in the timeline diagram presented in Fig. 1.

Statistical Analysis Methods: From 2008 to 2012, the predominant approach in ship anomaly behavior detection was the utilization of statistical analysis methods to detect ships. Statistical analysis methods rely on mathematical models based on the probability distribution of random variables. Kernel Density Estimation (KDE) is a nonparametric estimation method, which is one of the statistical analysis methods. Ristic et al. (2008) proposed a ship position and speed anomaly detection method based on KDE. They designed an adaptive KDE algorithm that automatically calculates the optimal bandwidth value of the Gaussian kernel, thereby improving the accuracy of detection. Laxhammar et al. (2009) constructed a ship behavior clustering model based on the Expectation–Maximization Gaussian Mixture Model (GMM), classifying ship trajectories outside the cluster as anomalous trajectories. Smith et al. (2012) developed a ship anomaly behavior recognition algorithm based on the Gaussian Process (GP) model, combining Gaussian processes and extreme value theory to identify anomalous behavior in ship flow data. Kowalska and Peel (2012) introduced active learning rules based on this algorithm, addressing issues such as the high computational complexity of the GP model. Vermard et al. (2010) introduced a Hierarchical Bayesian Model (HBM) based on Hidden Markov Models, utilizing basic information such as each fishing vessel's trajectory and speed to distinguish different behavioral states during fishing vessel operations.

Knowledge-Based and Data-Driven Integrated Methods: From 2012 to 2015, there emerged a shift towards methods that combine knowledge-based and data-driven approaches as relying solely on data to detect abnormal ship trajectories proved less accurate. Sheng et al. (2018) employs neural networks to initially detect anomalous data, followed by a secondary inspection combining expert knowledge to reduce false positives. Vandecasteele and Napoli (2012) designed a DASB expert system that associates trajectory data with a geographical spatial engine. Kazemi et al. (2013) proposed a hybrid anomaly detection framework based on open data and expert knowledge. The knowledge was derived from verified rules used by the Swedish Coast Guard to identify anomalous behavior, leading to the development of an open data anomaly detection system.

Traditional Machine Learning Methods: From 2015 to 2020, after the foundation laid by statistical analysis methods, the use of traditional machine learning algorithms became a promising direction. Traditional machine learning algorithms are further divided into unsupervised and supervised algorithms. Unsupervised algorithms are typically employed for extracting daily maritime vessel behavior patterns. Li et al. (2017) designed a constrained adaptive dynamic time warping algorithm to

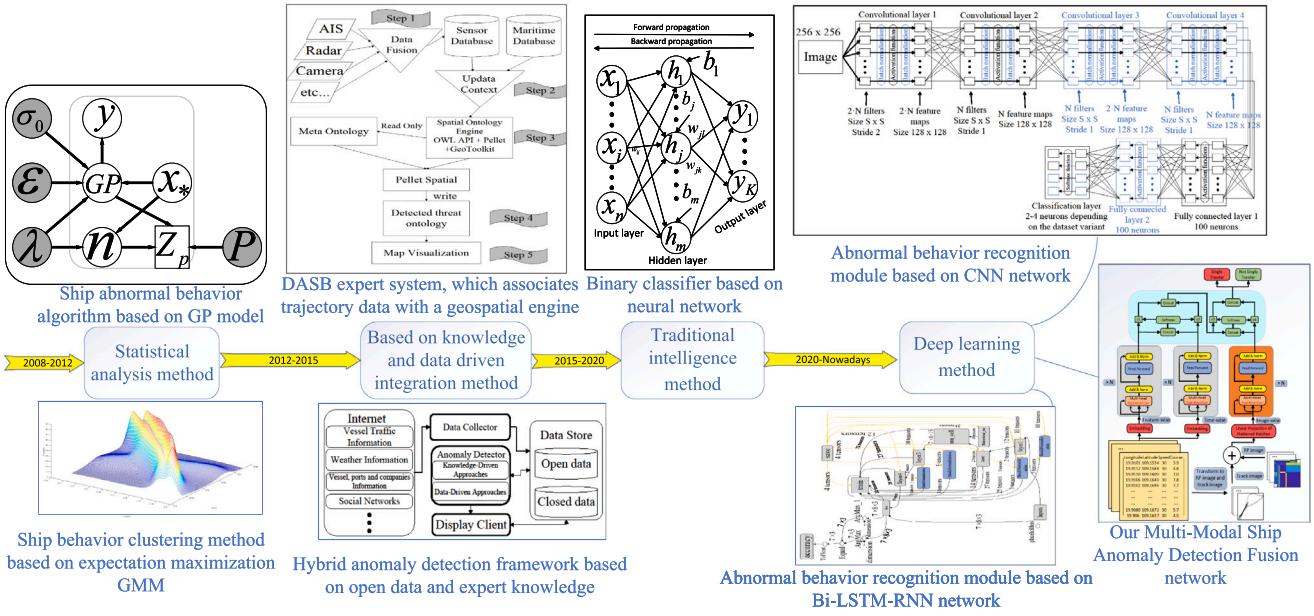


Fig. 1. The roadmap of ship behavior anomaly detection.

calculate distances between different trajectories. They used an improved center clustering algorithm and DBSCAN algorithm designed by Ester et al. (1996) to cluster ship trajectory data. Liu et al. (2019) developed a novel square-shaped neighborhood DBSCAN algorithm, combining DP and DTW algorithms with adaptive parameters to better detect anomalous ship behavior. Supervised methods are usually applied when pre-classified ship anomaly and normal data are available. Singh and Heymann (2020) proposed a neural network-based AIS detection method, training a binary classifier with historical AIS data to determine whether ship trajectory data indicates anomalous behavior. Elwakdy et al. (2015) segmented each ship trajectory into multiple segments and used polynomial functions to extract each sub-trajectory as input features. Subsequently, they built a ship classifier based on an Adaptive Neuro-Fuzzy Inference System (ANFIS) to classify tankers and fishing vessels, utilizing ship motion shapes to construct the classifier. Sheng et al. (2018) employed a logistic regression model, constructing a ship classifier using features directly extracted from ship AIS trajectories. This method established three logistic regression models for classification and is effective in identifying different motions and performs well on large datasets.

Deep Learning Methods: From 2015 to the present, deep learning has demonstrated advantages in feature extraction from large-scale data compared to traditional intelligent algorithms and has become an important pillar in various fields. In comparison to traditional methods, deep learning models exhibit continuous improvement in accuracy and performance, especially in these days of massive data. Among them, Nguyen et al. (2021) investigated trajectory prediction methods based on Recurrent Neural Network (RNN) models. Gao et al. (2018) developed a real-time ship behavior prediction model based on Bidirectional Long Short-Term Memory (LSTM). Czaplewski and Dzwonkowski (2022) plotted ship routes on grayscale images and constructed 1458 different CNN models for ship anomaly behavior recognition. Kroodsma et al. (2018) utilized fishing vessel AIS data and Convolutional Neural Networks (CNN) for fishing vessel operation type recognition, categorizing fishing vessels into seven classes. Fu et al. (2021) employed a vector encoding scheme based on trajectory sequences, using text vectors to train a word2vec model and calculating embedding features for each position. They proposed a recognition method based on a hierarchical ensemble framework to improve the accuracy of fishing vessel operation type recognition, which resulted in a significant improvement in recognition accuracy compared to a single model.

Multi-Source Data Fusion Methods: Multi-source data fusion methods represent a category within deep learning approaches. The acquisition of ship behavior data relies on various sensors, as data obtained from a single sensor alone is insufficient to meet the low false-positive requirements for ship behavior anomaly detection. Multi-source heterogeneous data effectively addresses this deficiency by incorporating static information from AIS data, accurately detecting ship targets using satellite imagery, detecting and tracking ship targets through video, and employing radar data to detect ship anomalies based on clustering methods, among other techniques. Utilizing video data to assist in identifying anomalous ship behavior is a viable option, but to do so in nighttime or adverse weather conditions, Lu et al. (2021) designed an effective deep neural network for image enhancement of low-quality video data, and Guo et al. (2021) designed a heterogeneous dual-dehazing network. Additionally, radar equipment demonstrates good identification accuracy and high coverage in coastal waters, unaffected by adverse weather conditions. Therefore, Lin et al. (2020) used an improved Euclidean clustering method to capture ship targets from point cloud data sensed by lidar, followed by continuous correction of ship position and heading deviations using AIS data.

Combining the various methods mentioned above, this paper employs the approach of transforming the TFFS data into corresponding trajectory point images and recurrence plot images to achieve multi-source fusion data. This method assists in discriminating whether fishing vessels exhibit illicit behavior, thereby enhancing recognition accuracy of our network.

3. Proposed methods

3.1. Problem formulation

In this section, the data used is represented by certain symbols. Due to the similarity between the structure of TFFS data and multivariate time series (MTS), symbols from MTS notation are adopted. The TFFS data used are denoted as $X \in R^{V \times T}$ where V is the number of features in the TFFS data, and T is the sequence length for each feature. Specifically, as expressed in Eq. (1):

$$X = \{X^1, X^2, \ldots, X^V\} \tag{1}$$

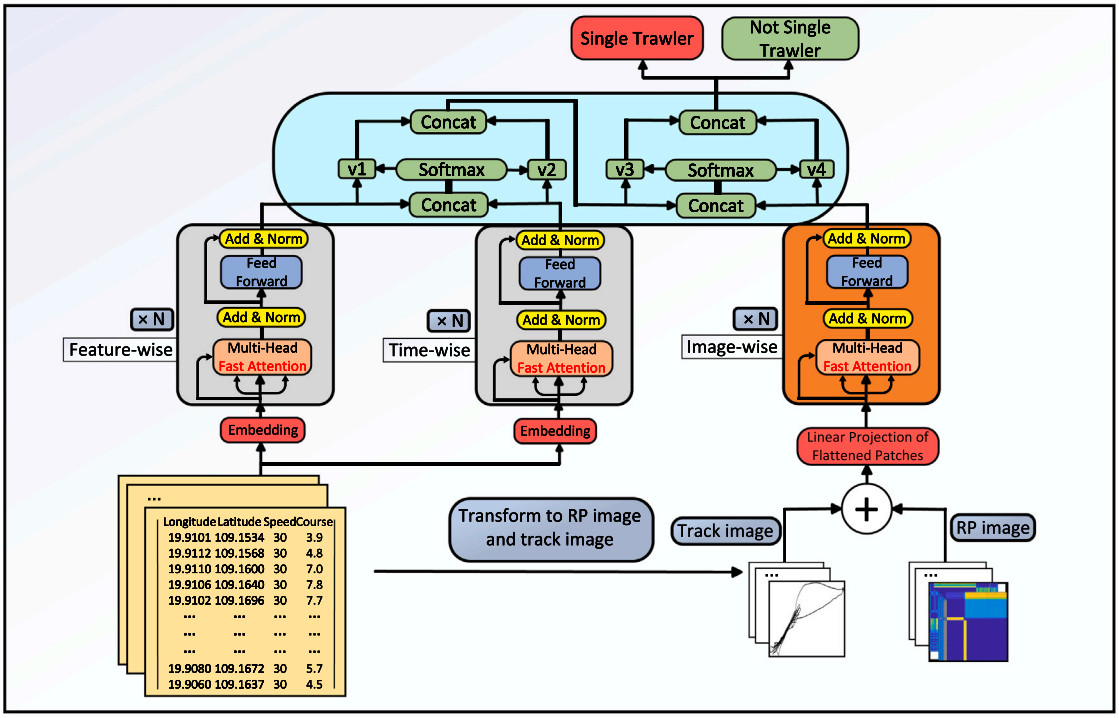


Fig. 2. MFGTN network structure.

where X^V represents the time series of the v-th variable, specified as follows in Eq. (2):

$$X^v = (X_1^v, X_2^v, \ldots, X_T^v) \tag{2}$$

where X_t^v represents the value at the t-th timestamp for the v-th variable.

Simultaneously, each X corresponds to a Y. The HN_STV dataset, as illustrated in the bottom left corner of Fig. 2, consists of N instances, and each instance has a corresponding class label, as shown in Eq. (3):

$$[X_1, Y_1], [X_2, Y_2], \ldots, [X_N, Y_N] \tag{3}$$

Given the HN_STV dataset is a binary classification dataset, Y is set to 1 for single-trawl fishing vessels and 0 for non-single-trawl fishing vessels. Therefore, the problem can be formalized as follows in Eq. (4):

$$X \overset{input}{\rightarrow} Network \overset{Predict}{\rightarrow} \begin{cases} Y = 1 \\ Y = 0 \end{cases} \tag{4}$$

Additionally, for the public dataset including three fishing activities, Y is assigned as follows: 0 for seine fishing data, 1 for trawler fishing data, and 2 for gillnet fishing data. Similarly, the problem for the public dataset including three fishing activities can be formalized in the following formula (5):

$$X \overset{input}{\rightarrow} Network \overset{Predict}{\rightarrow} \begin{cases} Y = 2 \\ Y = 1 \\ Y = 0 \end{cases} \tag{5}$$

3.2. MultiModal fast gated transformer

The proposed MFGTN network is illustrated in Fig. 2, where the functionalities of each module and the working principle of the network are as follows:

Module introduction: The yellow part in the bottom left corner of the image illustrates basic form of TFFS data we utilized, where each trajectory is represented as a list (depicted by a yellow box). Each row in the list corresponds to information at a timestamp, with two minutes interval between rows. Furthermore, each trajectory has four features, corresponding to the four columns in the list: longitude, latitude, speed, and course. The gray module are two encoder networks that process structured information of TFFS data, named Double-Tower Transformer, as shown in Section 3.7. The orange module is a encoder network that processes unstructured information of TFFS data, named Fusion former, as shown in Section 3.5. The portion of the red font in each encoder network represents changing Attention mechanism from a traditional encoder module to a more responsive Fast Attention mechanism, as shown in Section 3.6. The blue module illustrates the gate structure of the fused output across the three networks, called the Dual-gate structure, as shown in Section 3.8. Finally, the image in the bottom right corner depicts the unstructured information of TFFS data, which is trajectory point images and the recurrence plot images respectively.

Network working principle: First, the organized TFFS data, after undergoing embedding. In the Time-wise Encoder, each value of longitude/latitude is treated as a unique number and utilize a parameter matrix with a d_model value of 1024 to reduce its dimensionality, thus avoiding excessive resource consumption. In the Feature-wise Encoder, a parameter matrix with a d_model value of 512 is used to increase its dimensionality, amplifying the four features in the dataset to facilitate better exploration of the correlations among them by the network. Furthermore, all embeddings in the model share weights. Then, the TFFS data adding positional encoding and fed into Double-Tower Transformer module. The feature-wise encoder simultaneously takes multiple time series features to compute attention weights among different time series at all time steps. This is achieved by processing attention weights of feature values between multiple time series at the same time step to focus on different functionalities. The Time-wise encoder separately takes input from individual time series features to compute attention weights within each time series across all time steps. Meanwhile, the TFFS data transformed into images is sliced into patches and positional encoding is added. These are then inputted into the Fusion former module for feature extraction. Subsequently, the features extracted by N layers of Double-Tower Transformer and Fusion former module respectively are passed through a Dual-gate structure to determine the

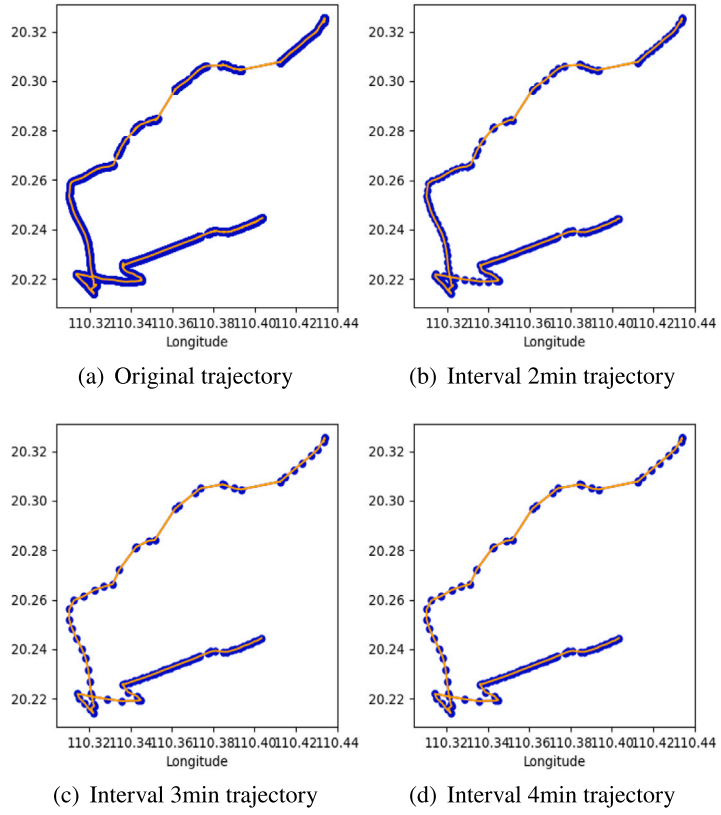


Fig. 3. Resampling of initial ship trajectory data. The horizontal coordinate in the image represents longitude and the vertical coordinate represents latitude.

weight ratio assigned to the outputs among the three encoders. Finally, a softmax function is applied to determine whether the input TFFS data corresponds to single trawler operation type.

3.3. HN_STV dataset

The single-trawl dataset within the scope of Hainan Province, referred to as HaiNan_SingleTrawlVessel, is abbreviated as HN_STV. The dataset utilized in this study is derived from ship trajectories in the vicinity of Hainan Province from July to September 2023. Initially, AIS data and radar data for each trajectory during this period were obtained from the database. Through predefined rules, the data from these two sources were effectively integrated to form the initial ship trajectories. The preliminary extraction resulted in 10,252 ship trajectories, encompassing those identified as single-trawl fishing vessels and other non-single-trawl fishing vessel trajectories. Subsequently, due to issues such as missing data, abnormal MMSI values, and abnormal time values in the obtained initial ship trajectories, further screening was performed. This involved filtering out duplicate and erroneous data, correcting abnormal values, resulting in 1256 positive samples and negative samples. Simultaneously, for the fused initial ship trajectory data, redundancy existed in the temporal dimension, leading to occurrences of multiple duplicate trajectory points within a timestamp. Therefore, the selected positive and negative samples underwent resampling to eliminate redundant data, with the resampling conducted at different time intervals, as illustrated in Fig. 3.

As observed in Fig. 3(b), the resampling effect at a 2-minute interval is optimal, with these trajectories exhibiting a uniform distribution of trajectory points while retaining the relevant features of the original trajectories. Therefore, for the fused initial ship trajectory data, the resampling interval of 2 min is chosen. The resampled trajectory data constitute our current HN_STV dataset, with an example sample presented in Table 1.

3.4. Recurrence plot images and trajectory point images

The recurrence plot (RP), initially conceived by Eckmann et al. (1995) as a visualization tool to reveal the recurrence of sequences in dynamic systems, is illustrated in Fig. 4. RP serves as a crucial method for analyzing the periodicity, chaos, and non-stationarity of time series. It unveils the internal structure of time series, providing prior knowledge about similarity, information content, and predictability. In recent years, RP has become indispensable for exposing hidden patterns and amplifying discriminative regions in time series data due to its graphical characteristics. The specific process of encoding time series into recurrence plot images is as follows:

Firstly, to ensure that different features have a consistent scale, the time series data undergoes linear transformation using Min-Max Normalization, wherein the original data is scaled to map its resulting values within the range of [0–1], as shown in Eq. (6).

$$x^* = \frac{x - Min(x)}{Max(x) - Min(x)} \tag{6}$$

where x represents the original time series data, and $Min(x)$ and $Max(x)$ represent the minimum and maximum values in that series.

Subsequently, according to Eq. (7), The linearly transformed sequence x^* is partitioned into multiple subsequences and mapped onto a multidimensional phase space, where each subsequence corresponds to a state within this space.

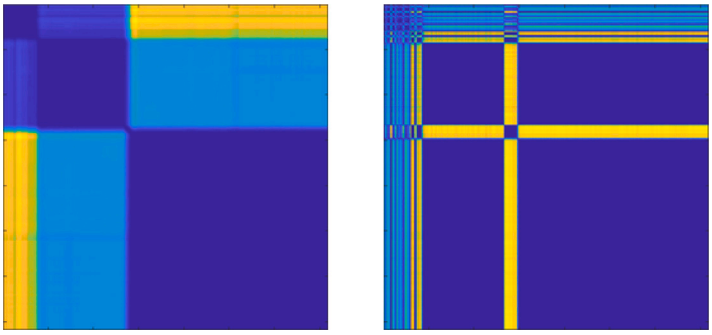
$$\vec{S} = [x^*(1 : len(x^*) - 1), x^*(2 : len(x^*))] \tag{7}$$

where $\vec{S}$ represents the set of each state in the phase space, $len(x^*)$ represents the number of elements in the sequence.

Finally, by employing Eq. (8), the distances between states in the phase space are computed, yielding a distance matrix. Based on this matrix, a color map can be generated to assign colors to each pixel,

Table 1
A sample example in the HN_STV dataset.

TargetId	Mmsi	Heading	Latitude	Longitude	Length	Speed	Course	State	RadarId	Timestamp
74200000	100899627	63	20.32378	110.4335	30	3.9	63.1	1	None	2023/7/2 17:49
74200000	100899627	94	20.32538	110.4338	30	3.9	94.7	1	HaiNan403	2023/7/2 17:51
74200000	100899627	163	20.32520	110.4340	30	3.8	163.2	1	HaiNan403	2023/7/2 17:53
74200000	100899627	221	20.32067	110.4306	30	4.1	221	1	HaiNan403	2023/7/2 17:55
...	...	...	...	...	...	...	...	...	...	...
74200000	100899627	241	20.31827	110.4274	30	4.2	241.7	1	None	2023/7/2 19:57



(a) Feature1 : Longitude (b) Feature2 : Latitude

Fig. 4. Recurrence plot images example.

thereby generating the recurrence plot image. Eq. (8) formally defines the recurrence plot.

$$RP_{i,j}(\varepsilon) = \theta(\varepsilon - \| \vec{S}(i) - \vec{S}(j)\|), \vec{S}(\cdot) \in \Re^m, i, j = 1, \ldots, N \tag{8}$$

where $RP_{i,j}(\varepsilon)$ represents a pixel in the RP image, $\vec{S}(i)$ denotes the i-th state in the phase space, and a subsequence sampled at the i-th position in the time series. The symbol $\|\cdot\|$ indicates a norm operation. θ is the Heaviside step function, utilized to binarize the distance matrix through a threshold ε, m signifies the number of points for each state in the phase space. N stands for the total number of states, determining the size of the recurrence plot.

Moreover, to avoid the loss of texture information, the threshold in Eq. (8) is often omitted. Therefore, Eq. (8) can be simplified to Eq. (9):

$$RP_{i,j}(\varepsilon) = \| \vec{S}(i) - \vec{S}(j)\|, \vec{S}(\cdot) \in \Re^m, i, j = 1, \ldots, N \tag{9}$$

Due to the multi-feature nature of the multivariate time series, a single recurrence plot is insufficient to represent the features of the entire time series. Therefore, the channel fusion recurrence plot is proposed, which overlaps the recurrence plots of each feature to form a multivariate time series superimposed recurrence plot, as shown in Eq. (10):

$$MRP(\varepsilon) = \left\{RP^1(\varepsilon),\ RP^2(\varepsilon), \ldots, RP^V(\varepsilon)\right\} \tag{10}$$

where RP^i represents the value in Eq. (9), and V is the number of features in the multivariate time series.

Simultaneously, due to the variability in the number of features in different multivariate time series, some multivariate time series may have a higher number of features. Therefore, multiple recurrence plots need to be superimposed. This imposes a significant burden on the training and testing load of the network, leading to large training time expenses and substantial GPU memory requirements. To address the structural symmetry of the recurrence plot's main diagonal, the recurrence plots of two features in a multivariate time series are concatenated along the main diagonal to reduce the number of required recurrence plots. This converts Eq. (10) to (11) and the concatenated image can be observed as depicted in Fig. 5(a) and (b).

$$\begin{aligned} MSRP(\varepsilon) &= \left\{MSRP^1, MSRP^2, \ldots, MSRP^{V/2}\right\} \\ &= \{[RP^1(\varepsilon), RP^2(\varepsilon)], [RP^3(\varepsilon), RP^4(\varepsilon)], \ldots, \\ &\quad [RP^{V-1}(\varepsilon), RP^V(\varepsilon)]\} \end{aligned} \tag{11}$$

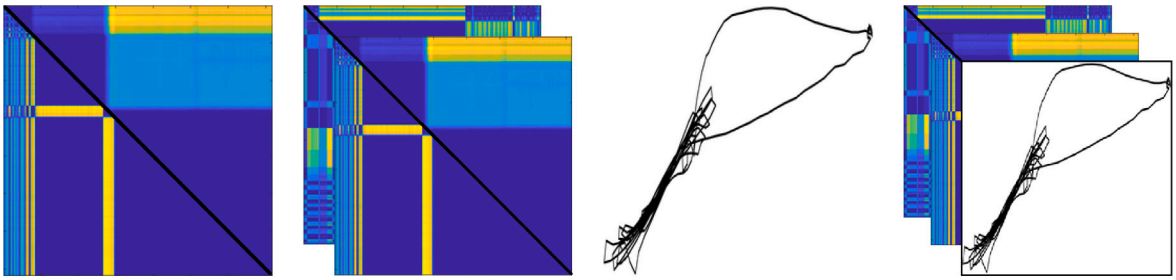
Besides, trajectory point images are also crucial for detection. Trajectory point images constitute vital characteristics of fishing vessel activities. Acknowledging the potent capability of Transformer structures in capturing trajectory features, certain unstructured information present in these images still plays a significant role in identifying single-trawl fishing operations. Latitude and longitude in the TFFS data are then converted into trajectory point images. These trajectory point images are subsequently concatenated with the recurrence plot images at the channel level for ease of input processing by the network. The specific process is illustrated in Fig. 5(c) and (d).

3.5. Fusion former module

In recent years, with the increasing research on Transformers, various Transformer variants have emerged for image analysis (Lin et al., 2022). Meanwhile, utilizing TFFS data alone for extracting information on fishing vessel operations is not comprehensive enough, as TFFS data lacks non-structural data such as images. However, relying solely on TFFS data to extract information on fishing vessel operations may not be comprehensive enough, as TFFS data lacks non-structural data such as images. Therefore, the TFFS data are transformed into multivariate recursive trajectory diagrams as unstructured data to compensate for the limitations of a single data type. Proposed as a derivative of the recently introduced Vision Transformer (ViT) module by Dosovitskiy et al. (2020), the Fusion Former network aims to learn and extract unstructured features. The Fusion former replace the original Attention mechanism with the Fast Attention mechanism to improve operational efficiency. Additionally, for the structured prior information of TFFS data, which includes ship longitude, latitude, speed, and course, the Double-Tower Transformer is used to extract structured features.

The workflow of the Fusion former module is roughly described as follows. The input Multivariate recursive trajectory diagram are divided into multiple patches. Each patch is then projected into a fixed-length vector, with a special token added to the projected vector to correspond to the final output category prediction. The process is represented by the following Eq. (12):

$$Z_0 = [x_{class}; x_p^1 E; x_p^2 E; \ldots; x_p^N E] + E_{pos}$$



(a) Feature 3 fused with feature 4 (b) Fusion feature superposition (c) Trajectory point image (d) Fusion image

Fig. 5. Multivariate fusion superposition image.

$$E \in R^{(P^2\times C)\times D}, E_{pos} \in R^{(N+1)\times D}, x_p \in R^{N\times(P^2\times C)} \tag{12}$$

where, Z_0 is the output after preprocessing, x_{class} is the special symbol token, x_p^i is a patch, E is the linear projection vector, N is the number of patches, P is the size of each patch, E_{pos} is the position code, D is the dimensional size of a linear projection and C is the number of channels.

The processed features are fed into the Fast Attention module, as detailed in Section 3.6. The output obtained after the Fast Attention module is used for predicting the final output.

3.6. Fast attention module

Due to the quadratic computational complexity of the Transformer regarding input sequence length, resulting in inefficient computation, the network model of Fastformer designed by Wu et al. (2021) is referred. The Fast Attention module is embedded into MFGTN, replacing the original multiplicative attention module to additivity attention, thus substituting the original attention module to accelerate the training speed and reduce the computational resource overhead of the network model. Similar to the original attention module, it starts by transforming the input matrix into query, key, and value matrices $Q, K, V \in R^{N\times d}$ using three linear layers with non-shared parameters, where N is the length of the sequence and d is the dimension of the channel, which can also be denoted as a combination of a series of vectors $Q = [q_1, q_2, \ldots, q_N], K = [k_1, k_2, \ldots, k_N], V = [v_1, v_2, \ldots, v_N]$. The Q matrix is then summarized into a global vector $q \in R^d$ using additive attention. It compresses the global context information of query, and the attention weight α_i of the ith query vector is calculated as following Eq. (13):

$$\alpha_i = \frac{exp\left(w_q^T q_i \Big/ \sqrt{d}\right)}{\sum_{j=1}^{N} exp\left(w_q^T q_i \Big/ \sqrt{d}\right)} \tag{13}$$

where, $w_q \in R^d$ is a learnable parameter variable and d is the dimension of the channel. The global query vector is calculated as following Eq. (14):

$$q = \sum_{i=1}^{N} \alpha_i q_i \tag{14}$$

Subsequently, the interaction between the global query vector and each key vector is modeled through element-wise multiplication of these two matrices. They are then combined to form a globally context-aware key matrix. The i-th vector in this matrix, denoted as p_i, is represented as $p_i = q * k_i$. Similarly, additive attention is used to aggregate the key matrix with global context awareness, where the additive attention weight calculation for the i-th vector is given by the following Eq. (15):

$$\beta_i = \frac{exp\left(w_k^T p_i \Big/ \sqrt{d}\right)}{\sum_{j=1}^{N} exp\left(w_k^T p_i \Big/ \sqrt{d}\right)} \tag{15}$$

where, $w_k \in R^d$ is the attention parameter vector and d is the dimension of the channel. The global key vector is calculated as following Eq. (16):

$$k = \sum_{i=1}^{N} \beta_i p_i \tag{16}$$

Finally, the interaction between the value matrix and the global key vector is modeled, the product of elements is performed between the global key and each value vector, and the key–value interaction vector u_i is calculated, expressed as $u_i = k * v_i$. Similar to Transformer, the authors apply a linear transformation layer to each key–value interaction vector to learn its hidden representation. The output matrix of this layer is expressed as $R = [r_1, r_2, \ldots, r_N] \in R^{N\times d}$. This matrix and query matrix are further added to form the final output of Fast Attention, as shown in the following Eq. (17):

$$output = R + Q = [r_1, r_2, \ldots, r_N] + [q_1, q_2, \ldots, q_N] \tag{17}$$

3.7. Double-tower transformer

Traditional Transformer modules focus only on extracting contextual features along the temporal dimension of sequential data (Vaswani et al., 2017). However, for multi-dimensional time series data, there may exist hidden correlations among different features. Losing the inter-feature correlations can easily lead to incomplete learning, resulting in a loss of accuracy. Therefore, for the fusion data of multi-dimensional time series, unlike traditional Transformer modules, the Double-Tower Transformer employs two Encoder modules to perform feature extraction along both the temporal and feature dimensions. In each tower, the encoder captures hierarchical and inter-level correlations explicitly through attention and masking, as illustrated in the gray module in Fig. 2.

Time-wise Encoder: The input consists of features from a single time series. Additionally, attention weights are computed across all time steps within each time series using a Fast Attention mechanism. Simultaneously, consistent with the original encoder structure, fully connected feedforward layers are stacked on each multi-head attention layer to enhance feature extraction. Residual connections around the two sub-layers are also retained to guide information and gradient flow, followed by layer normalization.

Feature-wise Encoder: Similarly, the Feature-wise Encoder receives features from multiple time series and computes attention weights among different time series across all time steps, specifically focusing on the attention weights of feature values between multiple time series at the same time step. This is done to attend to the intrinsic information interactions among different series. For instance, longitude and latitude occur simultaneously; hence, losing the intrinsic correlation information between these two features at the same time step could significantly impact the network's recognition performance.

3.8. Dual-gate structure

To merge the output features from the Time-wise, Feature-wise, and Fusion former modules, a straightforward combination of all features might degrade the performance of the Double-Tower network. Therefore, a Dual-gate control structure is proposed to effectively control the weights of each network, as illustrated in the blue module in Fig. 2. In the Double-Tower architecture, Time-wise and Feature-wise correspond to different types of features. Following non-linear activation modules, these two features are treated as T and F, respectively, forming a gate structure. By concatenating them into a vector and subjecting them to linear projection, the concatenated vector h_1 is obtained as shown in the following Eq. (18):

$$h_1 = W_1 \cdot Concat(F, T) + b_1 \tag{18}$$

where, h_1 represents the result of the fusion of Time-wise and Feature-wise, T represents the feature of Time-wise and F represents the feature of Feature-wise.

The gating weights v_1, v_2 of Time-wise and Feature-wise module can be obtained respectively, as shown in the following Eq. (19):

$$v_1, v_2 = Softmax(h_1) \tag{19}$$

Then the output vector y_1 of Double-Tower Transformer is calculated according to the gating weight, as shown in the following Eq. (20):

$$y_1 = Concat(F \cdot v_1,\ T \cdot v_2) \tag{20}$$

Since the vision module and Double-Tower module belong to different datasets and have different features extracted, the output of the two nonlinear activation modules is followed by the fully connected layer as V and y_1. Similar to the above, the splicing vector h_2 is obtained, as shown in the following Eq. (21):

$$h_2 = W_2 \cdot Concat(y_1, V) + b_2 \tag{21}$$

where, h_2 represents the result of the fusion of vision module and Double-Tower module and V represents the feature of vision module.

Thus, the gating weights v_3, v_4 of vision module and Double-Tower module are obtained, as shown in the following Eq. (22):

$$v_3, v_4 = Softmax(h_2) \tag{22}$$

Finally, the output of the duel-gate structure is obtained as following Eq. (23):

$$Gate_{Output} = Concat(y_1 \cdot v_3, V \cdot v_4) \tag{23}$$

4. Experimental results and analysis

In this section, the performance of MFGTN will be assessed.. Firstly, the experimental setup is introduced in Section 4.1, covering the environment, dataset, parameter setting, and evaluation metrics. Next, baseline methods compared in Section 4.2 are presented. Section 4.3 reports the comparison with State-of-the-art methods, and Section 4.4 delves into ablation experiments. Finally, the experimental results are discussed in Section 4.5.

4.1. Experimental setup

Environment: All experiments were conducted on a machine equipped with an Intel(R) Xeon(R) Gold 6330 CPU @ 2.00 GHz and an NVIDIA RTX 3090. Python 3.8 and PyTorch 2.1.0 were used for building and training our models.

Datasets: (1) HN_STV dataset. (2) Public dataset including three fishing activities. The latter consists of trajectory data from ships in the East China Sea (anonymized). It represents real historical maritime vessel trajectory data, covering multiple dimensions of information. Each trajectory includes details such as vessel ID, latitude, longitude, speed, course, time, and operational mode (trawling, encircling, and gillnetting). The latter dataset can be found on the https://aistudio.baidu.com/datasetdetail/146541. The latter dataset comprises 14,656 training samples and 3664 testing samples, with an equal distribution of positive and negative samples.

Table 2
Confusion matrix.

Confusion		Predition	
		Positive	Negative
Reference	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Parameter Setting: In the MFGTN network, some key hyperparameters used have the following values: the embedding dimension d_{model} is 1024 and 512, the input length d_{input} for time series is 1391, the feature dimension $d_{channel}$ for time series is 4, the output dimension d_{output} for the classifier is 2, the number of layers for the encoder is 8, the number of heads for multi-head attention is 8, the learning rate is 1e-5 and the optimizer selected is Adagrad.

Evaluation Metrics: Classification accuracy, Recall, Precision, F1-score, ROC curve and Time/Sample are chosen as evaluation metrics. Wherein, Time/Sample denotes the time taken to test each individual sample, measured in seconds. Meanwhile, before introducing other metric, it is essential to understand the confusion matrix, as shown in Table 2:

True Positive and True Negative are correctly predicted data, while False Positive and False Negative are incorrectly predicted data, with the following meanings:

TP (True Positive): Instances correctly predicted as positive. In other words, the actual value is positive, and the prediction is also positive.

TN (True Negative): Instances correctly predicted as negative. This refers to situations where the actual value is negative, and the prediction is also negative.

FP (False Positive): Instances incorrectly predicted as positive. This occurs when the actual value is negative, but it is incorrectly predicted as positive.

FN (False Negative): Instances incorrectly predicted as negative. This happens when the actual value is positive, but it is incorrectly predicted as negative.

From these definitions, the definitions of Accuracy, Precision, Recall, and F1-Score can be derived.

Accuracy represents the proportion of correctly classified samples to the total number of samples. It is defined as follows Eq. (24):

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{24}$$

Precision represents the proportion of samples predicted as positive among those that are actually positive. It is defined as follows Eq. (25):

$$Precision = \frac{TP}{TP + FP} \tag{25}$$

Recall represents the proportion of actual positive samples among those predicted as positive. It is defined as follows Eq. (26):

$$Recall = \frac{TP}{TP + FN} \tag{26}$$

F1-Score is a weighted average of precision and recall, defined as follows Eq. (27):

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{27}$$

Precision reflects the model's ability to differentiate negative samples, with higher precision indicating stronger discrimination against negative samples. Recall reflects the model's ability to identify positive samples, with higher recall indicating stronger identification of positive

samples. F1-Score combines both, and a higher F1-Score suggests a more robust model.

ROC Curve is a graphical tool used to represent the performance of a classification model. It depicts the performance of the classifier at different thresholds by plotting the true positive rate (TPR) and false positive rate (FPR) on the y-axis and x-axis, respectively.

True Positive Rate (TPR): also known as Recall, measures the classifier's ability to correctly identify positive instances. TPR can be understood as the rate of detection within all positive instance, so a higher TPR is better. Its calculation formula is the same as recall, as shown in Eq. (28):

$$TPR = \frac{TP}{TP + FN} \tag{28}$$

False Positive Rate (FPR): FPR indicates the proportion of negative instances that the model incorrectly predicts as positive. It can be understood as the rate of false positives within all actual negative instances (false alarm rate), so a lower FPR is better. Its calculation formula is as follows Eq. (29):

$$FPR = \frac{FP}{FP + TN} \tag{29}$$

AUC (Area Under the Curve): AUC is the area under the ROC curve and is used to measure the classifier's performance. A higher AUC value indicates better classifier performance, while a lower AUC value suggests poorer performance.

Additionally macro and micro evaluation metrics are included in the ROC curve graph, as shown in Figs. 6 and 7. The macro metric calculates TPR/FPR/AUC separately for each class and then computes the arithmetic average. The micro metric performs a global computation by summing up all confusion matrices and then calculating TPR/FPR/AUC. These two metrics, as with AUC in the ROC curve, are represented by the area under the curve, where a larger area indicates superior recognition performance of the classifier.

4.2. Baselines

MFGTN was compared with four traditional deep learning baseline methods and three recent networks, providing a brief introduction to each method. Interested readers can refer to the original papers for more details.

Fully Convolutional Network (FCN) and Residual Network (ResNet) (Wang et al., 2017): These are reported as among the best deep learning models for multi-dimensional time series classification tasks (Ismail Fawaz et al., 2019). Multi-Layer Perceptron (MLP) is also included in our comparison as a simple baseline. For these three baseline networks, reference was made to the original papers for the proposed model structures and the network parameters were adjusted to be of the same order of magnitude.

General Neural Network Encoder (Serra et al., 2018): This structure is based on the Transformer's Encoder.

DA-Net (Chen et al., 2022): A model for multi-dimensional time series classification that employs a novel dual-attention-based network to explore local–global features in multi-dimensional time series classification.

MF-Net (Du et al., 2023): A model for multi-dimensional time series classification that utilizes global-local blocks to capture local features through an attention-based mechanism and spatial local blocks to capture spatial dependency features. It further incorporates a sparse self-attention mechanism to capture global features. Finally, combining local and global features, it utilizes spatial local blocks to capture spatial dependency features.

TimesNet (Wu et al., 2022): It achieves this by transforming one-dimensional time series into a set of two-dimensional tensors based on multiple periods, extending the analysis of temporal changes into two-dimensional space. It employs the proposed TimesBlock to adaptively discover the multi-periodicity adaptively and extract the complex temporal variations from transformed 2D tensors by a parameter-efficient inception block.

Dlinear (Zeng et al., 2023): Through investigation, it was found that employing positional encoding and using tokens to embed subseries in Transformer facilitate preserving some ordering information. However, the nature of the permutation-invariant self-attention mechanism inevitably results in temporal information loss. Therefore, a simple one-layer linear models named LSTF-Linear was proposed, and comprehensive empirical research was conducted, demonstrating the significant advantages of this model.

SVP-T (Zuo et al., 2023): A model for multi-dimensional time series classification that initially takes time series subsequences as input, originating from different variables and positions. It introduces a variable position encoding layer (vp layer) to leverage variable and positional information for each shape. Lastly, it incorporates a new vp (variable position)-based self-attention mechanism to enhance attention weights for overlapping shapes.

4.3. Comparison with state-of-the-art methods

Due to the limitations of the experimental data, the accuracy of all baseline experimental results is provided by the baseline models available in open-source networks under the same experimental conditions, utilizing the same dataset for testing. For reproducibility, a fixed random seed was set, and for consistency in presentation, two decimal places were retained. The results are presented in Table 3. To better showcase the advanced capabilities of our network as a classifier, ROC curves for all networks are plotted, as depicted in Fig. 6.

Additionally, to verify the generalization of our network, the public dataset including three fishing activities available online was also utilized to conduct comparative experimental tests. The experimental results and corresponding ROC curves are shown in Table 3 and Fig. 7 below.

4.4. Ablation study

In order to clearly illustrate the performance gains of each module in MFGTN, a comprehensive study is conducted as shown in Table 4. It can be seen from the last row of the table that by integrating all modules, experimental results such as accuracy rate, recall rate and memory-usage degree all reflect the effectiveness of each module in improving classification network. Meanwhile, the following conclusions can be drawn from Table 4:

1. The addition of the visual module, making full use of both structured and unstructured information in fusion data, is superior to utilizing only structured information in fusion data. As seen in Table 4, in the third row, the accuracy is increased by 2%, and the Recall, Precision, and F1-score indicators are improved to varying degrees.

2. Considering the time complexity of the Attention mechanism in Transformers, replacing the original module with the Fast Attention module improves network memory consumption and precision. As shown in Table 4, in the four rows with the addition of the Fast Attention module, under the same batch_size conditions, not only does the network's training memory consumption significantly decrease by half, but there is also improvement in accuracy and various indicators.

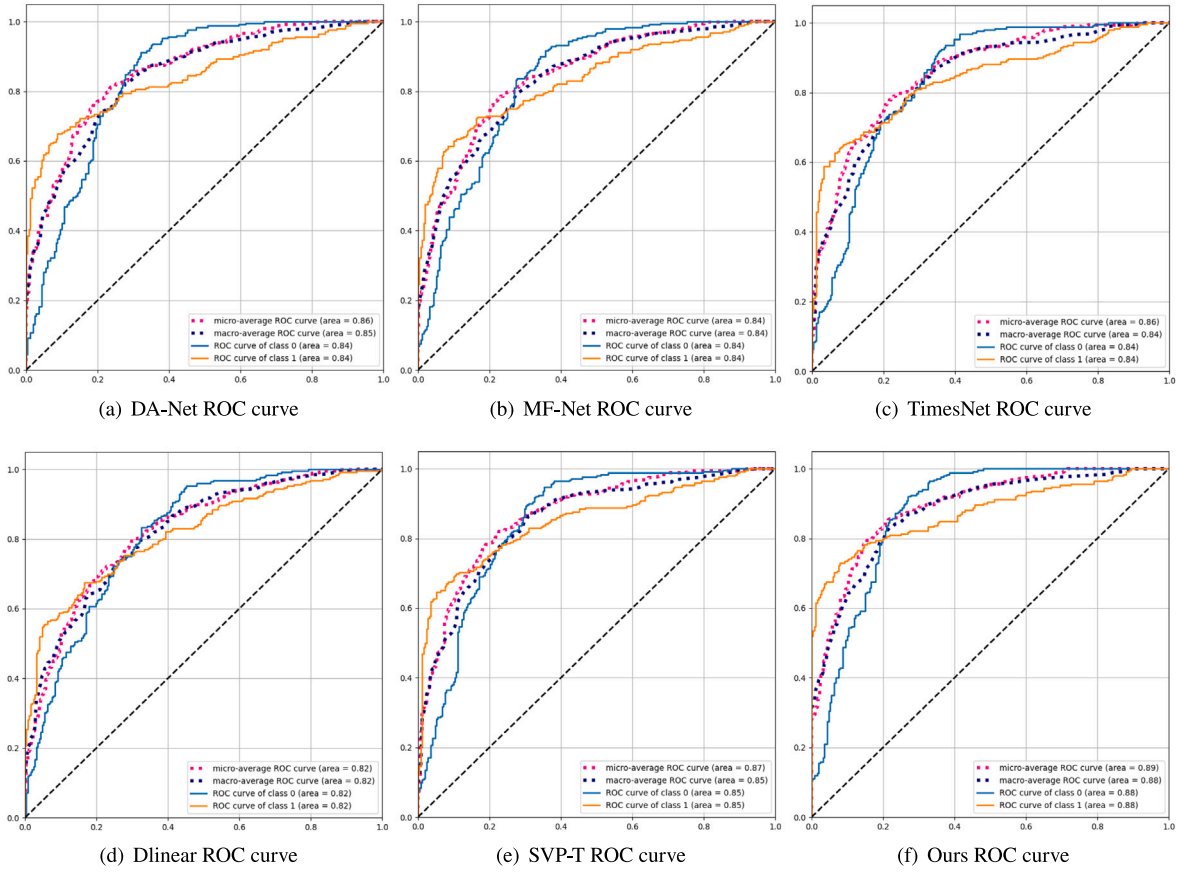
3. For the Double-Tower network, a proposed improved gate structure allocates different weights to outputs for time series and image data, addressing them separately. As indicated in Table 4, in the four rows with the addition of the Gate module, this gate structure proves to be effective. Analyzing the weights assigned by this gate structure when the network runs up to 2/3 of the epoch, extracted from the blue module in Fig. 2, the weight ratio is Feature-wise: Time-wise = 0.7103: 0.2887. This suggests that, concerning fishing vessel fusion information, the network places more emphasis on structural information at the feature level, as features include the important vessel

Table 3
The results of HN_STV dataset and public dataset including three fishing activities for each network.

	HN_STV				Public data					
	Accuracy	Recall	Precision	F1-score	Time/Sample	Accuracy	Recall	Precision	F1-score	Time/Sample
FCN (Wang et al., 2017)	68.18	67.27	69.53	68.38	0.0026	63.56	63.21	63.89	63.54	0.0031
ResNet (Wang et al., 2017)	70.17	70.29	70.93	70.61	0.0152	65.36	65.29	66.03	65.66	0.0143
MLP (Wang et al., 2017)	67.76	67.67	68.07	67.87	0.0130	62.95	62.68	62.05	62.36	0.0127
Encoder (Serra et al., 2018)	74.18	74.27	73.53	73.89	0.0132	77.97	77.92	78.55	77.98	0.0138
DA-Net (Chen et al., 2022)	81.55	81.34	83.87	81.15	0.0122	74.32	74.32	74.09	73.60	0.0125
MF-Net (Du et al., 2023)	79.76	79.52	82.64	79.21	0.0120	75.03	75.08	75.01	74.99	0.0115
TimesNet (Wu et al., 2022)	82.38	82.12	80.56	81.33	0.0156	78.52	77.12	78.04	77.58	0.0167
Dlinear (Zeng et al., 2023)	73.21	73.12	72.93	73.02	**0.0019**	70.32	70.04	71.64	70.83	**0.0023**
SVP-T (Zuo et al., 2023)	82.59	82.36	84.56	83.44	0.0102	79.66	79.64	79.67	79.37	0.0121
Ours	**84.92**	**84.82**	**85.53**	**84.83**	0.0078	**82.61**	**82.23**	**81.92**	**82.07**	0.0081

Table 4
The results of Ablation Study on the HN_STV dataset.

Double-Tower	Vision	Fast Attention	Gate	Accuracy	Recall	Precision	F1-score	Memory-Usage
✓				80.74%	80.72%	80.96%	80.84%	16894MiB
✓	✓			82.56%	82.42%	84.30%	83.35%	23715MiB
✓		✓		82.41%	82.03%	83.98%	82.99%	8694MiB
✓			✓	81.23%	81.05%	81.78%	81.41%	17219MiB
✓	✓	✓		84.13%	84.01%	83.56%	83.78%	11024MiB
✓	✓		✓	83.12%	83.02%	83.82%	83.41%	23867MiB
✓		✓	✓	82.89%	82.87%	83.09%	82.97%	8934MiB
✓	✓	✓	✓	**84.92%**	**84.82%**	**85.53%**	**84.83%**	**11058MiB**



(a) DA-Net ROC curve (b) MF-Net ROC curve (c) TimesNet ROC curve

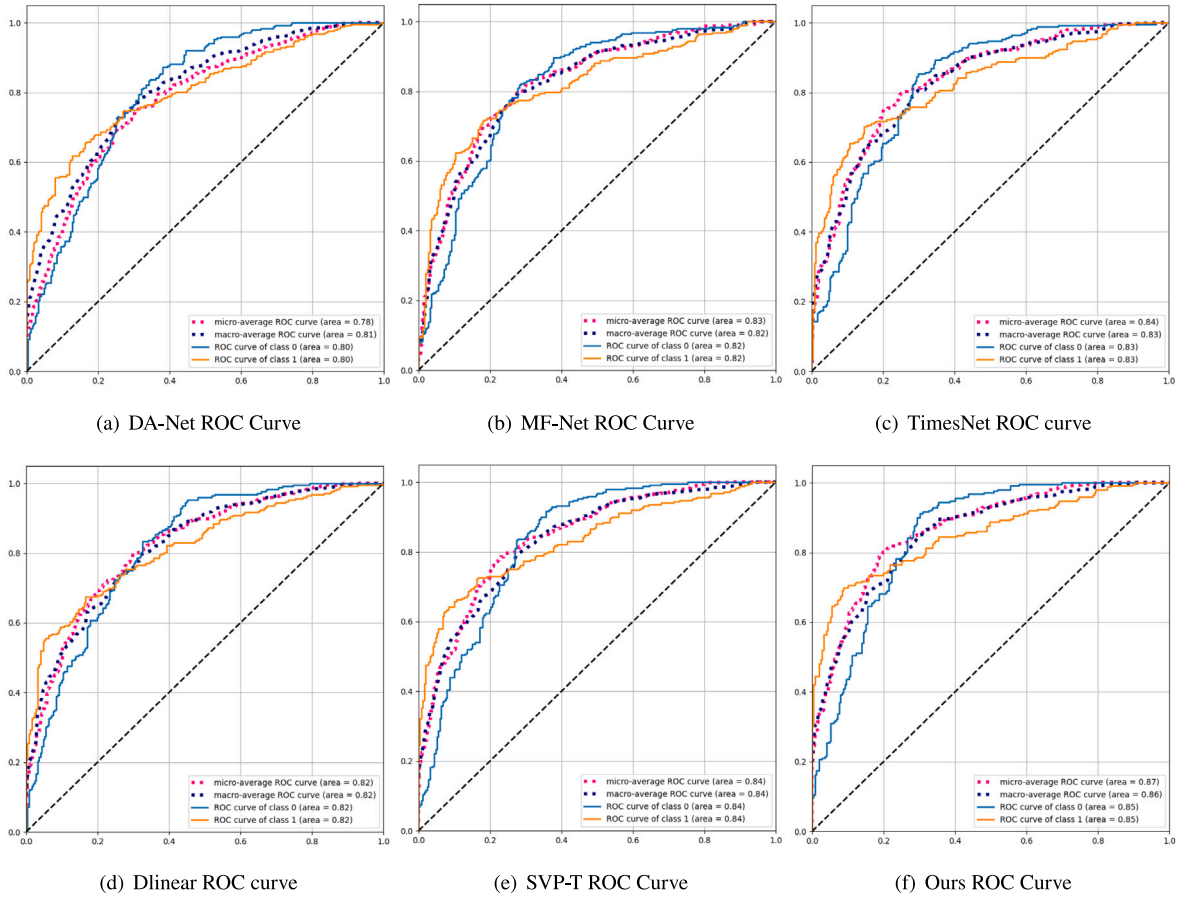
(d) Dlinear ROC curve (e) SVP-T ROC curve (f) Ours ROC curve

Fig. 6. ROC curves of each network in HN_STV dataset. The horizontal coordinate indicates False Positive Rate, and the vertical coordinate indicates True Positive Rate. The magenta dotted line indicates the micro-average ROC curve, the blue dotted line indicates the macro-average ROC curve, the cyan solid line indicates the ROC curve of class 0, and the yellow solid line indicates the ROC curve of class 1.

attributes of longitude, latitude, speed, and direction. Additionally, the weight ratio of Time Series data: Vision data is 0.6823: 0.3177. This indicates that the network still places considerable emphasis on the unstructured information in fusion data and effectively utilizes visual data to improve network classification accuracy.

4.5. Quantitative comparison

From Table 3, it can be observed that our method achieves an overall accuracy superior to all relevant methods in HN_STV dataset, with a 2.34% improvement compared to the second-ranking network,



(a) DA-Net ROC Curve (b) MF-Net ROC Curve (c) TimesNet ROC curve

(d) Dlinear ROC curve (e) SVP-T ROC Curve (f) Ours ROC Curve

Fig. 7. ROC curves of each network in public dataset including three fishing activities. The horizontal coordinate indicates False Positive Rate, and the vertical coordinate indicates True Positive Rate. The magenta dotted line indicates the micro-average ROC curve, the blue dotted line indicates the macro-average ROC curve, the cyan solid line indicates the ROC curve of class 0, and the yellow solid line indicates the ROC curve of class 1.

SVP-T. The recall, precision, and F1-score values are also superior to other methods, with increases of 2.46%, 0.97%, and 1.39%, respectively, compared to the second-ranking network, SVP-T. Meanwhile, as indicated by the "Time/sample" column in Table 3, it is observed that the our method to test each sample is 0.0078 s, second only to two networks that use the fully connected layer as a classifier. This shows that our approach has significant advantages in terms of efficiency across all complex networks and validates the effectiveness of the fast attention mechanism integrated into MFGTN networks.

The primary reason for the generally low experimental accuracy of each network in Table 3 is that, to simulate the diversity of vessel types present in real oceans, some non-single-trawl fishing vessel data are randomly extracted from actual vessel trajectories. These data do not represent a single vessel operation type; for instance, they may include cruise ships, container ships, passenger ships, etc. Therefore, the sample distribution in non-single-trawl data is not regular but rather chaotic.

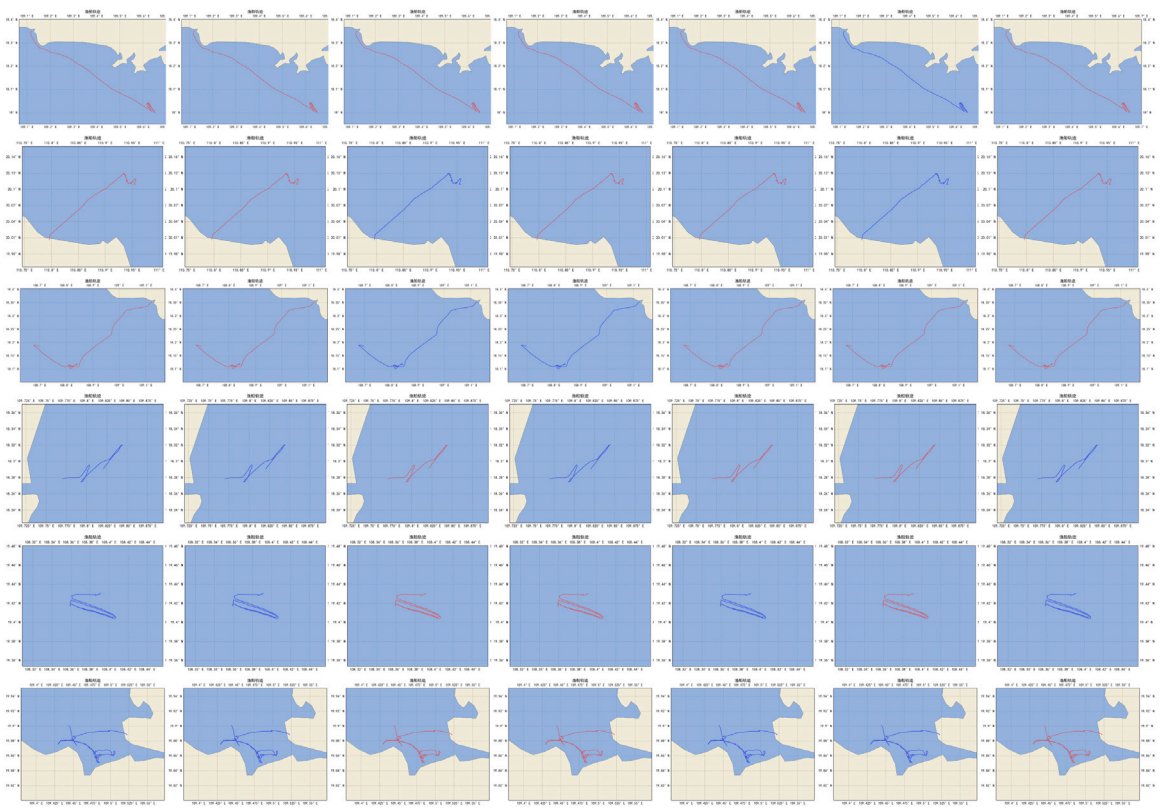
At the same time, as shown in Fig. 6, all curves of our network are located in the upper-left quadrant (0.2, 0.8) of the ROC curve, while the ROC curves of other networks such as DA-Net, MF-Net, and SVP-T are situated in the lower-right quadrant (0.2, 0.8). Consequently, the AUC area of our network is greater than that of other networks, as well illustrated in these eight comparative graphs, reflecting the superiority of our network.

Results from the public dataset including three fishing activities also reveal that our method's overall accuracy surpasses all relevant methods, with a 2.95% improvement compared to the second-ranking network, SVP-T. The recall, precision, and F1-score values are also superior to other methods, with increases of 2.59%, 2.25%, and 2.70%, respectively, compared to the second-ranking network, SVP-T. Moreover, it can be observed from the "Time/sample" column that our network still ranks third in time consumed per sample in this dataset, with the lowest time cost of all complex networks at 0.0081 s. This substantial lead in all complex networks still exists. Meanwhile, the performance of our network in the ROC curve is also enhanced, although not significantly, indicating that our network's classification ability on datasets with large data volumes still needs improvement.

4.6. Qualitative analysis

To better demonstrate the recognition capability of the MFGTN network, three trajectories of single-trawl fishing operations and three trajectories of non-single-trawl fishing operations were plotted separately. In these plots, trajectories highlighted in red represent single-trawl fishing operations, as shown in column (a) of Fig. 8. According to the prior knowledge of maritime personnel, single-trawl fishing vessels typically operate extensively in offshore waters, engaging in linear fishing activities, and exhibit characteristics such as departing from the shore. From the first three plots in Fig. 8(a), it is evident that these three trajectories correspond to single-trawl fishing operations. The fourth plot in column (a) indicates a non-single-trawl fishing operation trajectory, which did not depart from the shore and had a short fishing distance. The fifth plot clearly represents a non-single-trawl fishing operation trajectory. Although the trajectory in the sixth plot departed from the shore, its main activity range was not in offshore waters.

Subsequently, each of these six trajectories was inputted into each network for detection, and the detection results were then re-labeled and plotted, as shown in Fig. 8(b)–(g). From the obtained detection results, it can be seen that the MFGTN network accurately identified the trajectories of single-trawl fishing vessels and highlighted them in red.



(a) Images (b) **Ours** (c) DA-Net (d) MF-Net (e) Timesnet (f) Dlinear (g) SVP-T

Fig. 8. The operation result of MFGTN network in real scenario. The red trajectory represents a single-trawl vessel.

Meanwhile, DANet, MFNet, and Dlinear exhibited varying degrees of false alarm and miss rate, while TimesNet and SVP-T showed different degrees of false alarm. The experimental results above collectively demonstrate the superiority of our network.

4.7. Discussions

While Sections 4.4 to 4.6 adequately demonstrate the advantages of our model over some of the latest and typical networks, there are still certain limitations:

1. Although our model has significantly reduced the GPU memory consumption during training, the memory capacity on some micro-terminals is still insufficient. This limitation restricts the use of the model to testing on micro-terminals and prevents retraining the model when it becomes corrupted or when receiving large batches of new data.

2. Our model is designed to adapt to complete and clear trajectory data at ocean. However, when the received data contain errors or are blurry or fragmented, our model may easily lead to false alarm or omissions.

3. As shown in Table 3, while our model has outperformed some complex models in terms of data processing speed, there is still room for improvement while simpler models are more efficiently.

In response to these three points of discussion, we will continue to refine and optimize our model in the future.

5. Conclusion and prospect

In this paper, a Fast Gate-level Transformer network for multi-source information processing is proposed, named MFGTN. We leverage both the structured and unstructured information in TFFS data to enhance the performance of the network classifier. This is achieved by transforming TFFS data into recurrence plot images and trajectory point images, which are then input into the Fast former visual module. The features obtained from the time series module are combined with the visual module's output using an improved gate structure to achieve accurate data classification. To address challenges related to the large volume of TFFS data and the high memory consumption and training time of the network, we introduce the Fast Attention module to improve the traditional Attention mechanism in Transformers, reducing memory consumption by half. Experimental results demonstrate that MFGTN achieves the highest accuracy and outperforms existing methods in various metrics.

For future work, we intend to further improve the accuracy of MFGTN by focusing on enhancing features at the feature step level, as observed in Section 4.5. Additionally, we plan to propose more effective methods for improvement, especially tailored to datasets with large data volumes, to further enhance the classifier's classification capabilities.

CRediT authorship contribution statement

Yanming Gu: Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Zhuhua Hu:** Software, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Yaochi Zhao:** Conceptualization. **Jianglin Liao:** Validation, Software, Investigation, Data curation, Conceptualization. **Weidong Zhang:** Supervision, Software, Resources, Data curation, Conceptualization.

Declaration of competing interest

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

Data availability

The data that has been used is confidential.

Acknowledgments

This research was supported by the Key Research and Development Project of Hainan Province, China (Grant No. ZDYF2022GXJS348 and Grant No. ZDYF2022SHFZ039), the National Natural Science Foundation of China (Grant No. 62361024 and Grant No. 62161010), and the Hainan Province Natural Science Foundation (623RC446). The authors would like to thank the referees for their constructive suggestions.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.oceaneng.2024.117711.

References

Chen, R., Yan, X., Wang, S., Xiao, G., 2022. DA-Net: Dual-attention network for multivariate time series classification. Inform. Sci. 610, 472–487. http://dx.doi.org/10.1016/j.ins.2022.07.178.

Czaplewski, B., Dzwonkowski, M., 2022. A novel approach exploiting properties of convolutional neural networks for vessel movement anomaly detection and classification. ISA Trans. 119, 1–16. http://dx.doi.org/10.1016/j.isatra.2021.02.030.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16 × 16 words: Transformers for image recognition at scale. http://dx.doi.org/10.48550/arXiv.2010.11929, arXiv preprint arXiv:2010.11929.

Du, M., Wei, Y., Zheng, X., Ji, C., 2023. Multi-feature based network for multivariate time series classification. Inform. Sci. 639, 119009. http://dx.doi.org/10.1016/j.ins.2023.119009.

Eckmann, J.-P., Kamphorst, S.O., Ruelle, D., et al., 1995. Recurrence plots of dynamical systems. World Sci. Ser. Nonlinear Sci. Ser. A Monogr. Treatises 16, 441–446. http://dx.doi.org/10.1209/0295-5075/4/9/004.

Elwakdy, M., El-Bendary, M., Eltokhy, M., 2015. A novel trajectories classification approach for different types of ships using a polynomial function and ANFIS. In: Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition. IPCV, The Steering Committee of The World Congress in Computer Science, Computer . . . , p. 387.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96, pp. 226–231.

Feng, Y., Zhao, X., Han, M., Sun, T., Li, C., 2019. The study of identification of fishing vessel behavior based on VMS data. In: Proceedings of the 3rd International Conference on Telecommunications and Communication Engineering. pp. 63–68. http://dx.doi.org/10.1145/3369555.3369574.

Fu, H., Gao, S., Peng, Y., Zhao, N., 2021. Prediction of fishing vessel operation mode based on stacking model fusion. In: Journal of Physics: Conference Series, vol. 1792. IOP Publishing, 012030. http://dx.doi.org/10.1088/1742-6596/1792/1/012030.

Gao, M., Shi, G., Li, S., 2018. Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network. Sensors 18, 4211. http://dx.doi.org/10.3390/s18124211.

Guo, Y., Lu, Y., Liu, R.W., Wang, L., Zhu, F., 2021. Heterogeneous twin dehazing network for visibility enhancement in maritime video surveillance. In: 2021 IEEE International Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 2875–2880. http://dx.doi.org/10.1109/itsc48978.2021.9564887.

Hruschka, E.R., Hruschka, E.R., Ebecken, N.F.F., 2007. Bayesian networks for imputation in classification problems. J. Intell. Inf. Syst. 29, 231–252. http://dx.doi.org/10.1007/s10844-006-0016-x.

Hu, F., Zhong, H., Wu, C., Wang, S., Guo, Z., Tao, M., Zhang, C., Gong, D., Gao, X., Tang, C., et al., 2021. Development of fisheries in China. Reprod. Breed. 1, 64–79. http://dx.doi.org/10.1016/j.repbre.2021.03.003.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A., 2019. Deep learning for time series classification: A review. Data Min. Knowl. Discov. 33, 917–963. http://dx.doi.org/10.1007/s10618-019-00619-1.

Kazemi, S., Abghari, S., Lavesson, N., Johnson, H., Ryman, P., 2013. Open data for anomaly detection in maritime surveillance. Expert Syst. Appl. 40, 5719–5729. http://dx.doi.org/10.1016/j.eswa.2013.04.029.

Kowalska, K., Peel, L., 2012. Maritime anomaly detection using Gaussian process active learning. In: 2012 15th International Conference on Information Fusion. IEEE, pp. 1164–1171.

Kroodsma, D.A., Mayorga, J., Hochberg, T., Miller, N.A., Boerder, K., Ferretti, F., Wilson, A., Bergman, B., White, T.D., Block, B.A., et al., 2018. Tracking the global footprint of fisheries. Science 359, 904–908. http://dx.doi.org/10.1126/science.aao5646.

Laxhammar, R., Falkman, G., Sviestins, E., 2009. Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. In: 2009 12th International Conference on Information Fusion. IEEE, pp. 756–763.

Li, H., Liu, J., Liu, R.W., Xiong, N., Wu, K., Kim, T.-h., 2017. A dimensionality reduction-based multi-step clustering method for robust vessel trajectory analysis. Sensors 17, http://dx.doi.org/10.3390/s17081792.

Lin, J., Koch, L., Kurowski, M., Gehrt, J.-J., Abel, D., Zweigel, R., 2020. Environment perception and object tracking for autonomous vehicles in a harbor scenario. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1–6. http://dx.doi.org/10.1109/itsc45102.2020.9294618.

Lin, T., Wang, Y., Liu, X., Qiu, X., 2022. A survey of transformers. AI Open http://dx.doi.org/10.1016/j.aiopen.2022.10.001.

Liu, J., Li, H., Yang, Z., Wu, K., Liu, Y., Liu, R.W., 2019. Adaptive douglas-peucker algorithm with automatic thresholding for AIS-based vessel trajectory compression. IEEE Access 7, 150677–150692. http://dx.doi.org/10.1109/access.2019.2947111.

Lu, Y., Guo, Y., Zhu, F., Liu, R.W., 2021. Towards low-visibility enhancement in maritime video surveillance: An efficient and effective multi-deep neural network. In: 2021 IEEE International Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 2869–2874. http://dx.doi.org/10.1109/itsc48978.2021.9564669.

Mazzarella, F., Vespe, M., Alessandrini, A., Tarchi, D., Aulicino, G., Vollero, A., 2017. A novel anomaly detection approach to identify intentional AIS on-off switching. Expert Syst. Appl. 78, 110–123. http://dx.doi.org/10.1016/j.eswa.2017.02.011.

Nguyen, D., Vadaine, R., Hajduch, G., Garello, R., Fablet, R., 2021. GeoTrackNet—A maritime anomaly detector using probabilistic neural network representation of AIS tracks and a contrario detection. IEEE Trans. Intell. Transp. Syst. 23, 5655–5667. http://dx.doi.org/10.1109/tits.2021.3055614.

Perera, L.P., Oliveira, P., Soares, C.G., 2012. Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction. IEEE Trans. Intell. Transp. Syst. 13, 1188–1200. http://dx.doi.org/10.1109/tits.2012.2187282.

Ristic, B., La Scala, B., Morelande, M., Gordon, N., 2008. Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction. In: 2008 11th International Conference on Information Fusion. IEEE, pp. 1–7.

Riveiro, M., Pallotta, G., Vespe, M., 2018. Maritime anomaly detection: A review. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 8, e1266. http://dx.doi.org/10.1002/widm.1266.

Sánchez Pedroche, D., Amigo, D., García, J., Molina, J.M., 2020. Architecture for trajectory-based fishing ship classification with AIS data. Sensors 20, 3782. http://dx.doi.org/10.3390/s20133782.

Serra, J., Pascual, S., Karatzoglou, A., 2018. Towards a universal neural network encoder for time series. In: CCIA. pp. 120–129. http://dx.doi.org/10.48550/arXiv.1805.03908.

Sheng, K., Liu, Z., Zhou, D., He, A., Feng, C., 2018. Research on ship classification based on trajectory features. J. Navig. 71, 100–116. http://dx.doi.org/10.1017/s0373463317000546.

Singh, S.K., Heymann, F., 2020. Machine learning-assisted anomaly detection in maritime navigation using AIS data. In: 2020 IEEE/ION Position, Location and Navigation Symposium. PLANS, IEEE, pp. 832–838. http://dx.doi.org/10.1109/plans46316.2020.9109806.

Smith, M., Reece, S., Roberts, S., Rezek, I., 2012. Online maritime abnormality detection using gaussian processes and extreme value theory. In: 2012 IEEE 12th International Conference on Data Mining. IEEE, pp. 645–654.

Tursi, A., Maiorano, P., Sion, L., D'Onghia, G., 2015. Fishery resources: Between ecology and economy. Rendiconti Lincei 26, 73–79. http://dx.doi.org/10.1007/s12210-014-0372-3.

Vandecasteele, A., Napoli, A., 2012. Spatial ontologies for detecting abnormal maritime behaviour. In: 2012 Oceans-Yeosu. IEEE, pp. 1–7. http://dx.doi.org/10.1109/oceans-yeosu.2012.6263532.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Vermard, Y., Rivot, E., Mahévas, S., Marchal, P., Gascuel, D., 2010. Identifying fishing trip behaviour and estimating fishing effort from VMS data using Bayesian hidden Markov models. Ecol. Model. 221, 1757–1769. http://dx.doi.org/10.1016/j.ecolmodel.2010.04.005.

Wang, Y., Liu, J., Liu, R.W., Liu, Y., Yuan, Z., 2023. Data-driven methods for detection of abnormal ship behavior: Progress and trends. Ocean Eng. 271, 113673. http://dx.doi.org/10.1016/j.oceaneng.2023.113673.

Wang, Z., Yan, W., Oates, T., 2017. Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1578–1585. http://dx.doi.org/10.1109/ijcnn.2017.7966039.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M., 2022. Timesnet: Temporal 2D-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186.

Wu, C., Wu, F., Qi, T., Huang, Y., Xie, X., 2021. Fastformer: Additive attention can be all you need. http://dx.doi.org/10.48550/arXiv.2108.09084, arXiv preprint arXiv:2108.09084.

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, vol. 37. pp. 11121–11128. http://dx.doi.org/10.1609/aaai.v37i9.26317.

Zhang, C., Liu, S., Guo, M., Liu, Y., 2023. A novel ship trajectory clustering analysis and anomaly detection method based on AIS data. Ocean Eng. 288, 116082. http://dx.doi.org/10.1016/j.oceaneng.2023.116082.

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. J. Navig. 70, 648–670. http://dx.doi.org/10.1017/s0373463316000850.

Zuo, R., Li, G., Choi, B., Bhowmick, S.S., Mah, D.N.-y., Wong, G.L.H., 2023. SVP-T: A shape-level variable-position transformer for multivariate time series classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37. pp. 11497–11505. http://dx.doi.org/10.1609/aaai.v37i9.26359.