Feng, Y, Wang, X, Luan, J, Wang, H, Li, H, Li, H, Liu, Z and Yang, Z

A novel method for ship carbon emissions prediction under the influence of emergency events

Transportation Research Part C 165 (2024) 104749

Contents lists available at ScienceDirect

Transportation Research Part C

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

A novel method for ship carbon emissions prediction under the influence of emergency events

Yinwei Feng [a,b], Xinjian Wang [a,b,c,d,e,*], Jianlin Luan [f], Hua Wang [g], Haijiang Li [f,*], Huanhuan Li [d], Zhengjiang Liu [a,b], Zaili Yang [d,h]

[a] Navigation College, Dalian Maritime University, Dalian 116026, China
[b] Key Laboratory of Navigation Safety Guarantee of Liaoning Province, Dalian 116026, China
[c] Seafarers Research Institute, Dalian Maritime University, Dalian 116026, China
[d] Liverpool Logistics, Offshore and Marine (LOOM) Research Institute, Liverpool John Moores University, Liverpool L3 3AF, UK
[e] Key Laboratory of International Shipping Development and Property Digitization of Hainan Free Trade Port, Hainan Vocational University of Science and Technology, Haikou 570100, China
[f] School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China
[g] School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei 230009, China
[h] Transport Engineering College, Dalian Maritime University, Dalian 116026, China

ARTICLE INFO

Keywords:
Maritime transport
Carbon emissions
Emergency events
Time series forecasting
Navigation states identification
Deep learning

ABSTRACT

Accurate prediction of ship emissions aids to ensure maritime sustainability but encounters challenges, such as the absence of high-precision and high-resolution databases, complex nonlinear relationships, and vulnerability to emergency events. This study addresses these issues by developing novel solutions: a novel Spatiotemporal Trajectory Search Algorithm (STSA) based on Automatic Identification System (AIS) data; a rolling structure-based Seasonal-Trend decomposition based on the Loess technique (STL); a modular deep learning model based on Structured Components, stacked-Long short-term memory, Convolutional neural networks and Comprehensive forecasting module (SCLCC). Based on these solutions, a case study using pre and post-COVID-19 AIS data demonstrates model reliability and the pandemic's impact on ship emissions. Numerical experiments reveal that the STSA algorithm significantly outperforms the conventional identification standard in terms of accuracy of ship navigation state identification; the SCLCC model exhibits greater resistance against emergency events and excels in comprehensively capturing global information, thus yielding higher accurate prediction results. This study sheds light on the changing dynamics of maritime transport and its impacts on carbon emissions.

1. Introduction

As global maritime transport burgeons, ship emissions have become a major concern of many countries and international organizations around the world (Huang et al., 2020). As one of the most important sources of greenhouse gases and air pollutants (International Maritime Organization, 2021), annual CO_2 equivalent emissions from shipping have reached 10.76 megatonnes, 2.89 percent of total global emissions (International Maritime Organization, 2020). As reported by literatures (Eyring et al., 2010; Park

* Corresponding authors.
E-mail addresses: fyinwleo@dlmu.edu.cn (Y. Feng), X.Wang1@ljmu.ac.uk (X. Wang), jianlin.luan@dlmu.edu.cn (J. Luan), hwang191901@hfut.edu.cn (H. Wang), haijiang@dlmu.edu.cn (H. Li), H.Li2@ljmu.ac.uk (H. Li), liuzhengjiang@dlmu.edu.cn (Z. Liu), Z.Yang@ljmu.ac.uk (Z. Yang).

https://doi.org/10.1016/j.trc.2024.104749
Received 7 February 2024; Received in revised form 5 June 2024; Accepted 2 July 2024
Available online 13 July 2024

et al., 2023), approximately 70 percent of CO_2 emissions from ships occur within 400 km of land. These emissions have wide-ranging, multi-scale, and multi-faceted impacts on human lives and societal sustainability. Furthermore, ship emissions exhibit a distinct seasonal pattern, with more pronounced pollution during the summer months (Aliabadi et al., 2015). To facilitate effective time-diverse regulation of ship emissions and mitigate the impact of pollutants and greenhouse gases across varying time frames, there is an urgent need for accurate emissions calculation and effective Time Series Forecasting (TSF) within the jurisdiction.

In the existing body of research, numerous studies have focused on assessing changes in air quality within specific harbours (Park et al., 2023; Sim et al., 2022). However, few research has addressed the challenge of forecasting ship emissions using TSF methods. This gap is primarily attributed to the absence of high-resolution ship emission databases in port areas or at sea (Alfaseeh et al., 2020; Peng et al., 2020). While numerous studies have formulated ship emission estimation models utilizing Automatic Identification System (AIS) for calculating emissions in designated sea areas or ports (Chen et al., 2021; Hussain et al., 2022; Muhammad and Long, 2020), the majority of these models lack precise algorithms for discerning the ship navigation states. Consequently, these models are arguably incapable of capturing different emissions from varying ship operations under dynamic navigational conditions, resulting in inadequate accuracy in both measuring and projecting ship emissions.

In addition to the reliance on high-resolution databases, unforeseen emergency events such as COVID-19 pandemic, the Russia-Ukraine crisis, and the Red sea crisis can lead to Concept Drift (CD) of data-streaming, affecting the accuracy of TSF results. For example, the emergence of COVID-19 profoundly impacted the shipping industry in January 2020 (Li et al., 2024a), fundamentally altering the trajectory of carbon emissions (Wang et al., 2023), as depicted in the time series of global merchant fleet carbon dioxide emissions in Fig. 1. This significant change reflects that CD of data-streaming (Karimian and Beigy, 2023) can potentially diminish the performance of multiple TSF models (Gharib et al., 2021; Hsiao et al., 2021; Mroua and Lamine, 2023).

To address the aforementioned challenges, this study aims to develop a novel framework for ship emissions calculation. The framework combines Lloyd's Ship Technical Specification Database (STSD) with AIS data to establish an updated ship information database. Furthermore, a novel navigation state identification algorithm, rooted in the spatiotemporal ship trajectory, is developed to accurately identify a ship's navigation state, employing the five navigation states defined by the International Maritime Organization (IMO). In pursuit of TSF for ship emissions in a specified maritime area, this study proposes a deep learning model with superior predictive performance and resistance against CD of data-streaming. To illustrate the application of these methods, the study conducts a case study and examines the CO_2 emissions from ships within China's Bohai Bay waters during the years 2019–2020.

The remaining sections of this paper are structured as follows. Section 2 provides a literature review on ship emissions calculation and TSF and outlines the specific contributions made by this study. Section 3 details the methodology employed for constructing a database of ship emissions, presents a problem description for the TSF task, and introduces the proposed deep learning model. Section 4 focuses on establishing a ship emissions database and the training of the TSF model. Section 5 evaluates the predictive performance of the TSF model, assesses its resistance to CD, and offers an interpretability analysis of the proposed deep learning model. Section 6 provides the conclusion and limitations.

2. Literature review

2.1. Carbon emission calculations

Current research relies on two primary approaches for estimating ship emissions: the "top-down" method based on macro modelling and the "bottom-up" method based on micro modelling (Chen et al., 2021; Nocera et al., 2017). The "top-down" approach, using macro-statistics like ship fuel consumption and emission factors, provides accurate global estimates but tends to underestimate regional emissions due to the lack of ship specifics (Chen et al., 2021; Huang et al., 2020). Conversely, the widely used "bottom-up"

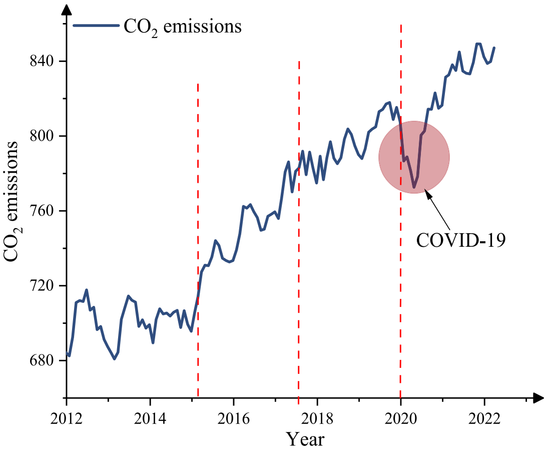


Fig. 1. Global merchant fleet CO_2 emissions from 2012 to 2022. Note: Data is from the United Nations Conference on Trade and Development (UNCTAD) (https://unctad.org/rmt2022).

approach calculates emissions based on detailed operational parameters, including engine type, power, and fuel specifics, making it more precise and dependable (Muhammad and Long, 2020; Murcia González, 2021). For the calculation of regional ship emissions, the "bottom-up" method based on micro modelling outperforms the "top-down" method based on macro modelling in both accuracy and temporal resolution (Fuentes García et al., 2021; Hickman et al., 2011; Yuan and Frey, 2021).

The performance of ships' engines, fuel consumption, and other emission indicators varies due to different navigation states (Muhammad and Long, 2020). Therefore, accurate identification of a ship's navigation state is paramount for the precision of the "bottom-up" approach in estimating carbon emissions (Li et al., 2023). For instance, in assessing emissions from ships along the China-Pakistan Economic Corridor, Muhammad and Long (2020) calculated CO_2 emissions based on average cruising speed, average maneuvering time, and berth time. Hussain et al. (2022) used ship-related data from the Pakistani port database to estimate carbon emissions during different operational phases. While these methods, to some extent, reflect the emissions of the study's subject, the inherent coarseness of the raw data limits their accuracy in capturing real emissions, resulting in a high margin of error.

AIS technology is crucial for accurately determining ships' navigational states and has gained prominence in maritime research (Chen et al., 2021; Huang et al., 2020; Xin et al., 2024). It continuously broadcasts ship information like position, speed, and load factor at short intervals, offering high temporal and spatial resolution (Huang et al., 2020). Additionally, the IMO mandates AIS for specific ship categories, leading to the widespread availability of AIS data and the construction of global shipping datasets (Chen et al., 2021; Huang et al., 2020). Consequently, the "bottom-up" approach, using AIS data, has gradually become the prevailing method for estimating ship emissions (Li et al., 2023; Weng et al., 2020).

The Ship Traffic Exhaust Assessment Model (STEAM), relying on AIS data, is a prominent example of the "bottom-up" approach (Jalkanen et al., 2009; Li et al., 2023). The STEAM model, drawing from extensive ship trajectory data obtained from AIS equipment and offering high spatial and temporal resolution emission measurements, has gained recognition from the IMO and numerous researchers (Fuentes García et al., 2021; Jalkanen et al., 2009; Li et al., 2023). Moreno-Gutiérrez et al. (2015) analysed nine methods for estimating carbon emissions by using AIS data to match engine power with shipping numbers in Lloyd's database. They found that the STEAM model to be the most reliable method. However, due to missing information in Lloyd's database, accurate assessment of total emissions within a sea area was hindered. Li et al. (2023) proposed a Random Forest-based iterative repair algorithm for missing Lloyd's data and used DBSCAN-based AIS trajectory repair for missing AIS points. They estimated CO_2 emissions within Domestic Emission Control Areas (DECA) using the repaired databases and IMO's navigation state criteria.

2.2. Time series forecasting

The TSF analysis is integral to modern data-driven research (Zeng et al., 2023). TSF methods, using historical data, are widely applied in forecasting epidemic spread (Istaiteh et al., 2020), traffic patterns (Bogaerts et al., 2020), financial investments (Khosravi and Ghazani, 2023), and air pollution levels (Alfaseeh et al., 2020). A compilation of recent TSF studies is presented in Table 1.

The evolution of TSF tasks spans three stages (Zeng et al., 2023): statistical models (Acharya and Singleton, 2022), machine learning (Ma et al., 2020; Zhang and Wang, 2018), and deep learning (Bogaerts et al., 2020; Kabir et al., 2023). Statistical models primarily use autoregressive time series data and aggregate information to predict future events (Acharya and Singleton, 2022). While machine learning and deep learning models vary their information extraction techniques based on the specific research area and dataset (Wu and Peng, 2022). Statistical models are proved easy to use and exhibit high accuracy when dealing with limited data, but their computational costs increase significantly, and prediction accuracy declines considerably when handling substantial datasets and extended forecasting horizons. In contrast, machine learning and deep learning models need separate training and test sets. In complex TSF tasks, machine learning and deep learning models often surpass statistical models (Kabir et al., 2023; Wu and Peng, 2022).

In machine learning, Support Vector Machines (SVM) (Zhang and Wang, 2018) and integrated tree-based learning models (Hu and Xiong, 2023; Tang et al., 2022) are commonly employed for simple TSF tasks. Complex TSF tasks frequently leverage various deep

Table 1
The recent research on TSF tasks.

Literature	Research area	Method type			Technique
		Statistical model	Machine learning	Deep learning	
Acharya and Singleton (2022)	Associations of air quality and weather with non-motorized counts	√			ARIMA
Ma et al. (2020)	Traffic forecasting	√	√		ARIMA, NN
Zhang and Wang (2018)	Electricity load forecasting		√		SVM
Bogaerts et al. (2020)	Traffic forecasting			√	CNN + LSTM
Kabir et al. (2023)	Fuel consumption forecasting			√	LSTM
Khosravi and Ghazani (2023)	Cryptocurrency market forecasting		√	√	RF, XGBoost, LSTM
Hu and Xiong (2023)	High-dimensional population inflow forecasting			√	Transformer
Park et al. (2023)	PM 2.5 and PM 10 forecasting			√	LSTM
Zhu and Li (2021)	Flight delay time forecasting			√	LSTM
Alfaseeh et al. (2020)	Greenhouse gas emission forecasting			√	LSTM

Note: Autoregressive Integrated Moving Average Model (ARIMA), Neural Network (NN), Support Vector Machine (SVM), Convolutional Neural Networks (CNN), Long short-term memory (LSTM), Random Forest (RF), eXtreme Gradient Boosting (XGBoost).

learning models (Park et al., 2023; Zeng et al., 2023). Recurrent Neural Networks (RNN) represent a specialized deep learning model for TSF, characterized by its recurrent structure retaining essential information in time series data (Hellton et al., 2022; Servizi et al., 2020). The Long Short-Term Memory (LSTM) model, an extension of RNN with a three-gate structure (memory gate, forgetting gate, and output gate), effectively addresses RNN issues like gradient vanishing and explosion while retaining valuable information (Khosravi and Ghazani, 2023; Park et al., 2023; Zhu and Li, 2021). For intricate time series data, several studies indicate that Convolutional Neural Networks (CNN) perform exceptionally well in extracting complex information (Bogaerts et al., 2020; Hou et al., 2022; Liu et al., 2022). In addition to the aforementioned deep learning models, the Transformer framework is considered one of the most successful sequence modelling approaches available (Hu and Xiong, 2023). It exhibits exceptional performance in Natural Language Processing (NLP), Computer Vision (CV), and TSF tasks (Hu and Xiong, 2023). Nevertheless, there is evidence suggesting that the effectiveness of Transformer in long-term time series tasks remains to be substantiated. Zeng et al. (2023) introduced a straightforward deep learning model, DLinear, and experimental results demonstrated that DLinear outperformed Transformer in TSF, all while offering significantly lower computational complexity.

Despite the various advantages associated with different TSF models, a significant challenge emerges in their practical applications, referred to as CD of data-streaming (Gharib et al., 2021; Hsiao et al., 2021). CD of data-streaming represents the alteration in the distribution or relationship of data in a time series over time (Karimian and Beigy, 2023). When a model encounters an unknown data transformation, its predictive accuracy declines (Mroua and Lamine, 2023). For instance, during global emergency events like the COVID-19 pandemic, disruptions in global finance, population movements, and transport patterns invalidate many predictive models (Gharib et al., 2021; Hsiao et al., 2021; Liu et al., 2021), highlighting the need for CD-resistance TSF models in practical engineering applications.

2.3. Research gaps and contributions

Despite the fact that extensive research contributions exist in ship emissions and TSF domains, notable research gaps persist. Firstly, a significant gap exists in the realm of accurately identifying a ship's navigation state, which consequently impairs the accuracy and resolution of models used for estimating ship emissions. Second, there is a paucity of research dedicated to the time series aspect of ship emissions, despite the numerous attempts to estimate ship emissions. This gap results in a certain delay in the implementation of time-differentiated measures by stakeholders. Finally, the practical application of TSF tasks has seen limited efforts to address the challenge of CD of data-streaming, rendering many models imperfect for real-world applications. In this study, the corresponding contributions to the above research gaps are made as follows:

1) A novel ship navigation state identification algorithm based on temporal and spatial changes in AIS trajectories is proposed in this study. This algorithm demonstrates high accuracy in identifying ship navigation states, thereby enhancing the precision of ship emission estimation models and the resolution of the emission database.

2) TSF is introduced to investigate ship carbon emissions, build a forecasting framework, and lay a foundation for solving the influence of emergency events. The result of TSF can also provide stakeholders with advanced, accurate information on emissions, enabling them to make well-informed decisions.

3) A rolling structured time series decomposition module for ship emission is developed. This module effectively circumvents potential data leakage issues during time series decomposition and exhibits substantial resistance to CD while achieving remarkable accuracy in TSF tasks.

4) An in-depth examination of the deep learning model using interpretability techniques is provided in this study. This highlights the model's strong interpretability and unveils the underlying factors contributing to its resistance against CD and high prediction accuracy.

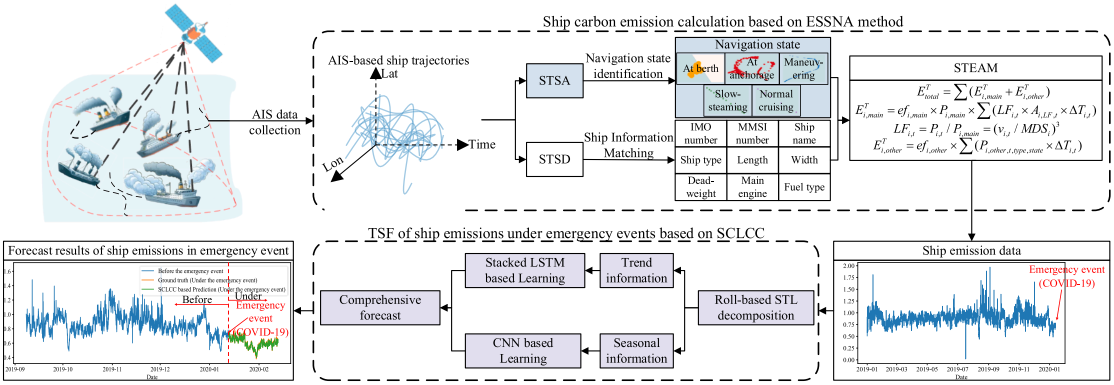

Ship carbon emission calculation based on ESSNA method
AIS data collection
AIS-based ship trajectories
Lat
Lon
Time
STSA
STSD
Navigation state identification
Ship Information Matching
Navigation state
At berth
At anchorage
Maneuv-ering
Slow-steaming
Normal cruising
IMO number
MMSI number
Ship name
Ship type
Length
Width
Dead-weight
Main engine
Fuel type
STEAM
$E^T_{total} = \sum (E^T_{i,main} + E^T_{i,other})$
$E^T_{i,main} = ef_{i,main} \times P_{i,main} \times \sum (LF_{i,t} \times A_{i,LF,t} \times \Delta T_{i,t})$
$LF_{i,t} = P_{i,t} / P_{i,main} = (v_{i,t} / MDS_i)^3$
$E^T_{i,other} = ef_{i,other} \times \sum (P_{i,other,t,type,state} \times \Delta T_{i,t})$
Ship emission data
Emergency event (COVID-19)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
2019-01
2019-03
2019-05
2019-07
2019-09
2019-11
2020-01
Date
TSF of ship emissions under emergency events based on SCLCC
Roll-based STL decomposition
Trend information
Seasonal information
Stacked LSTM based Learning
CNN based Learning
Comprehensive forecast
Forecast results of ship emissions in emergency event
Before the emergency event
Ground truth (Under the emergency event)
SCLCC based Prediction (Under the emergency event)
Before
Under
Emergency event (COVID-19)
1.6
1.4
1.2
1.0
0.8
0.6
0.4
2019-09
2019-10
2019-11
2019-12
2020-01
2020-02
Date

Fig. 2. The research framework of this study.

3. Datasets and methods

This study focuses on two main parts: establishing a ship emission database and developing a TSF model, as shown in Fig. 2. In Section 3.1, an algorithm for ship navigation state identification is developed to construct a ship carbon emission database. In Section 3.2, a novel deep learning model for TSF is developed to conduct the prediction and interpretation of ship emission time series, particularly under the influence of CD of data-streaming.

3.1. Carbon emission dataset construction based on navigation states identification

Analysing ship carbon emissions in a specific sea area through TSF necessitates a dataset containing high-resolution ship carbon emission data points (Alfaseeh et al., 2020; Luan et al., 2023). However, existing studies (Peng et al., 2020; Weng et al., 2020) emphasize the absence of an accurate dataset that can provide high-resolution carbon emission reports from ships operating in a particular sea region. Drawing insights from related studies (Alfaseeh et al., 2020; Wu et al., 2023), leveraging reliable models to generate relevant emission data appears to be a viable approach for training and testing deep learning models. Therefore, this study introduces an improved method for estimating Emission of Ships based on Ship Navigation Activity (ESSNA), as depicted in Fig. 3. As a micro modelling technique, ESSNA is tailored for estimating ship emissions at specific times and scenarios. It achieves this by estimating individual ship emissions using ship-specific information, including navigation state, ship size, and main engine power. Consequently, it offers highly accurate and time-resolved calculations of ship emission data, which are well-suited for specific studies.

Initially, the Lloyd's Ship Technical Specification Database (STSD), which was updated in a previous study (Li et al., 2023), is linked with the ship identification numbers in the AIS data to enhance the fundamental ship information. Then, a new Spatiotemporal Trajectory Search Algorithm (STSA) is proposed to identify the ship's navigational state, and the STEAM model (Peng et al., 2020) is used to calculate ship emission intensity, enabling the construction of a high-temporal-resolution Ship Carbon Emissions (SCE) database in the area. (Note: Specific details of the STEAM model are detailed in Appendix B).

3.1.1. The framework of navigation states identification

Ship emissions include main engine emissions and those from auxiliary engines or boilers, and these emissions rely on real-time power, with main engine power linked to the vessel's real-time speed. However, the parameters for auxiliary engine and boiler power remain lacking (Peng et al., 2020). To estimate the power of auxiliary engines and boilers, the IMO recommends the utilization of specific parameters, such as the ship's type, gross tonnage, and navigation state. The IMO classification standard classifies a ship's navigation state into five distinct categories: berthing, anchoring, maneuvering, slow-steaming, and normal cruising (International Maritime Organization, 2020; Li et al., 2023; Weng et al., 2020). The standard determines ship navigation states based on ship speed and main engine power. However, this method has two major drawbacks: it lacks robustness, as it is vulnerable to irregular data fluctuations, and it struggles to accurately classify a wide range of vessel types and sizes using predefined thresholds. To address the aforementioned issues, this research proposes the use of STSA based on AIS to determine ship's navigation states. The ship's trajectory is represented as $nt = \{p_n = \{l_n, t_n\} | n \in [1, N]\}$, where p_n denotes the spatiotemporal coordinates, encompassing temporal and spatial details. N represents the number of these spatiotemporal coordinates, arranged chronologically. Each spatiotemporal coordinate consists of $l_n = (la_n, lo_n)$, with lo_n and la_n representing longitude and latitude, and t_n signifying the corresponding timestamp. $snt = \{p_m | m \in [i, i+1, \ldots, j-1, j]; i \leqslant j, i, j \in [1, N]\}$ refers to a sub-trajectory of the overall trajectory. The STSA method analyses the ship's navigation state through continuous spatiotemporal sub-trajectories. Displayed in Fig. 3 and Table A1 in Appendix A, STSA classifies states into three groups: Mooring (berth and anchoring), Sailing (maneuvering and cruising), and Cruising (slow-steaming and normal cruising), based on varying motion characteristics. First, it identifies mooring sub-trajectories to differentiate between berthing and anchoring. Next, it discerns the sailing state from the remaining sub-trajectories to distinguish between maneuvering and cruising. Lastly, it differentiates slow-steaming from normal cruising within the identified cruising sub-trajectory.

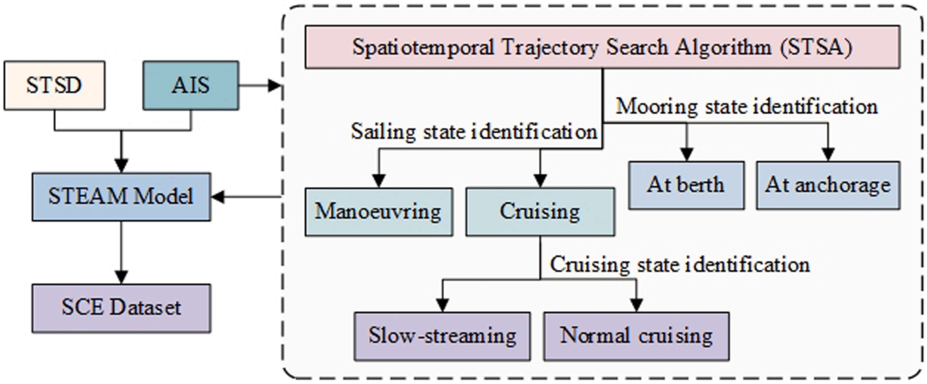

STSD
AIS
STEAM Model
SCE Dataset
Spatiotemporal Trajectory Search Algorithm (STSA)
Mooring state identification
Sailing state identification
Manoeuvring
Cruising
At berth
At anchorage
Cruising state identification
Slow-streaming
Normal cruising

Fig. 3. The procedure of SCE dataset construction.

3.1.2. The sub-algorithm of mooring state identification

Even if a ship is moored, external factors like wind, currents, or onboard equipment can induce slight positional shifts, causing variations in latitude and longitude data within a specific range. This study employs the ship trajectory's temporal duration and spatial extent to accurately identify the mooring state of the ship. By separately examining the displacement range of a ship in both anchor and berth states, a clear distinction between these two states is established. Ships in the anchor state exhibit a significantly larger spatial migration range compared to those in the berth state. In contrast, in the berth state, the duration of spatial migration of a similar scale is typically much longer than that in the anchor state. Hence, a Mooring State Identification Sub-Algorithm (MSISA) is introduced in this study to determine whether a ship trajectory represents an anchor or berth state based on specific temporal and spatial thresholds. Table A2 presents the pseudo-code for the MSISA algorithm, while Eqs. (1) to (3) outline the principles for distinguishing between berth and anchor states.

$$dis(p_i, p_m) = 2R \cdot \arcsin\sqrt{\sin^2\left(\frac{la_m - la_i}{2}\right) + \cos(la_m) \cdot \cos(la_i) \cdot \sin^2\left(\frac{lo_m - lo_i}{2}\right)} \tag{1}$$

$$dis(p_i, p_m) \leqslant \xi_{dis}, \forall m \in [i, j], \xi_{dis} = \{\xi_a, \xi_b\} \tag{2}$$

$$\Delta t(p_i, p_m) = t_m - t_i \geqslant \tau_{dis}, \forall m \in [i, j], \tau_{dis} = \{\tau_a, \tau_b\} \tag{3}$$

where $dis(p_i, p_m)$ represents the spatial distance between the ship's spatiotemporal coordinates p_i and p_m, R is the radius of the Earth. This study utilises Haversine's formula (Andreou et al., 2023) to accurately compute the ship's drift distance based on latitude and longitude coordinates. ξ_{dis} denotes the spatial distance threshold for mooring state identification, whereas ξ_a and ξ_b represent the spatial distance thresholds for anchor and berth, respectively. $\Delta t(p_i, p_m)$ represents the period between the spatiotemporal coordinates p_i and p_m of the ship. τ_{dis} is the threshold for identifying the mooring state, and τ_a, τ_b represent the threshold time intervals for anchor and berth identification, respectively.

3.1.3. The sub-algorithm of sailing state identification

Analysis of a ship's motion state reveals that during maneuvering process, it undergoes more frequent steering operations. In contrast, when the ship is cruising, its heading typically remains stable. Consequently, the Sailing State Identification Sub-Algorithm (SSISA) primarily employs changes in the ship's heading in the trajectory to identify maneuvering voyages (refer to Table A3 for the pseudo-code). The equations for calculating heading changes are presented in Eqs. (4) to (8):

$$C(p_i, p_m) = (x_m - x_i, y_m - y_i, z_m - z_i) \tag{4}$$

$$x_i = R \times \cos(lo_i) \times \cos(la_i) \tag{5}$$

$$y_i = R \times \cos(lo_i) \times \sin(la_i) \tag{6}$$

$$z_i = R \times \sin(la_i) \tag{7}$$

$$\Delta C\left(p_i, p_j, p_m\right) = \arccos\left(\frac{C\left(p_i, p_j\right) \cdot C(p_i, p_m)}{\left|C\left(p_i, p_j\right)\right| \times |C(p_i, p_m)|}\right) \tag{8}$$

where $C(p_i, p_m)$ represents the heading between the ship's spatiotemporal coordinates p_i and p_m; $\Delta C\left(p_i, p_j, p_m\right)$ represents the change angle of heading between heading $C\left(p_i, p_j\right)$ and $C(p_i, p_m)$; $C\left(p_i, p_j\right)$ is the initial heading, p_i is the initial spatiotemporal coordinate of the sub-trajectory, and p_j represents the first spatiotemporal coordinate where the initial heading exists.

In this study, the endpoint (p_e) of the sub-trajectory (*snt*) is determined as the spatiotemporal coordinate with the most significant change in heading angle from the initial heading, as represented in Eq. (9). If the sub-trajectory *snt* satisfies Eq. (10), it is classified as a maneuvering trajectory.

$$\Delta C\left(p_i, p_j, p_e\right) \geqslant \Delta C\left(p_i, p_j, p_m\right), \forall m \in [i, e] \tag{9}$$

$$\Delta C\left(p_i, p_j, p_e\right) \geqslant \eta \tag{10}$$

where η is the threshold of heading change angle.

In Eq. (10), p_e is the sub-trajectory endpoint and η is the threshold. To effectively determine the value of p_e, this study introduces the spatiotemporal coordinate search number threshold (λ). The algorithm begins by designating the initial spatiotemporal coordinate with a heading difference as a temporary endpoint. It then continues searching for larger heading difference angles in subsequent coordinates. If a greater angle is found within λ coordinate points, the corresponding coordinate becomes the new temporary endpoint. However, if no larger angle is discovered and the heading difference angle satisfies Eq. (10), then the sub-trajectory $snt(p_i, p_e)$ is classified as a maneuvering navigation state; otherwise, it is categorized as a cruising state.

3.1.4. The sub-algorithm of cruising state identification

The main distinction between slow-steaming and normal cruising lies in the ship's speed (Gan et al., 2022; Weng et al., 2020). A ship is identified in the slow-steaming state when its speed is 12 knots or less, as per IMO standards (International Maritime Organization, 2020). Conversely, it is classified as normal cruising when exceeding this threshold (Gan et al., 2022). Considering the continuous nature of normal cruising, a change in sailing state isn't triggered merely by the speed dipping below the threshold at specific spatiotemporal positions. Therefore, the Cruising State Identification Sub-Algorithm (CSISA) relies on the sub-trajectory's average speed for identification. Sub-trajectories that meet the conditions described in Eqs. (11) and (12) are identified as normal cruising states, while those that fail to meet these criteria are recognized as slow-steaming states.

$$v\left(p_i, p_j\right) = \frac{dis\left(p_i, p_j\right)}{\Delta t\left(p_i, p_j\right)} \tag{11}$$

$$\overline{v}(snt(p_i, p_k)) = \frac{\sum_{h=i}^{k-1} v(p_i, p_{i+1})}{k-1} > 12 \tag{12}$$

where $v\left(p_i, p_j\right)$ denotes the speed of the ship between spatiotemporal coordinates p_i and p_m; $\Delta t\left(p_i, p_j\right)$ denotes time difference of the ship between spatiotemporal coordinates; $\overline{v}(snt(p_i, p_k))$ denotes the average speed of the sub-trajectory $snt(p_i, p_k)$; p_k denotes the endpoint of the sub-trajectory.

Eq. (12) emphasizes the significance of determining p_k value as it enhances the average speed's resilience against minimal abnormal values, avoiding potential misjudgments caused by sporadic anomalies. However, during the initial phase of a trajectory's transition from normal cruising to slow-steaming, the average speed might misjudge. To address this, this study introduces a sequential threshold ϑ, which tracks speed anomalies. Specifically, when the algorithm is searching for the spatiotemporal coordinate p_h and detects more than ϑ consecutive spatiotemporal coordinates with speeds less than or equal to 12, it will backtrack to the ϑ ship spatiotemporal coordinates and designate $p_{h-\vartheta}$ as the concluding point of the sub-trajectory. The pseudocode for CSISA can be found in Table A4.

3.2. Modular deep learning algorithm based on decomposition

3.2.1. Problem statement

Let the time series data for carbon emissions from ships in a specific sea area be denoted as $X = \left[x_1, x_2, x_3, ..., x_{d_T}\right]^T \in \mathbb{R}^{d_T}$, where x_t represents the carbon emissions from ships at time t, and d_T represents the length of the carbon emissions time series. $\widehat{X}$ is forecasted by the development of a predictive function using historical information.

Since the raw data, denoted as X, consists of changes in carbon emission data over an extended time frame for a specific marine area, which cannot be directly assimilated by the model. Therefore, a sliding time window approach is employed to transform the raw data into a set of learnable examples. In the rolling time window setting, the encoder length (i.e., look-back window) is denoted as d_{LB}, and the decoder length (i.e., forecasting horizons) is denoted as d_{FH}.

To unveil the shared nonlinear mapping relationship, represented as a nonlinear function, between each pair of encoder and decoder, Eq. (13) is given as follows:

$$\widehat{x}_{(t+1):(t+d_{FH})} = \wp\left(x_{(t-d_{LB}+1):t}\right) \tag{13}$$

where t, $\forall t \in [d_{LB}, d_{LB} + 1, ..., d_T - d_{FH}]$, indicates the initiation of the rolling time window; $x_{(t-d_{LB}+1):t}$ represents the past observed carbon emissions; $\widehat{x}_{(t+1):(t+d_{FH})}$ denotes the anticipated future carbon emissions; and $\wp()$ signifies the shared nonlinear map between

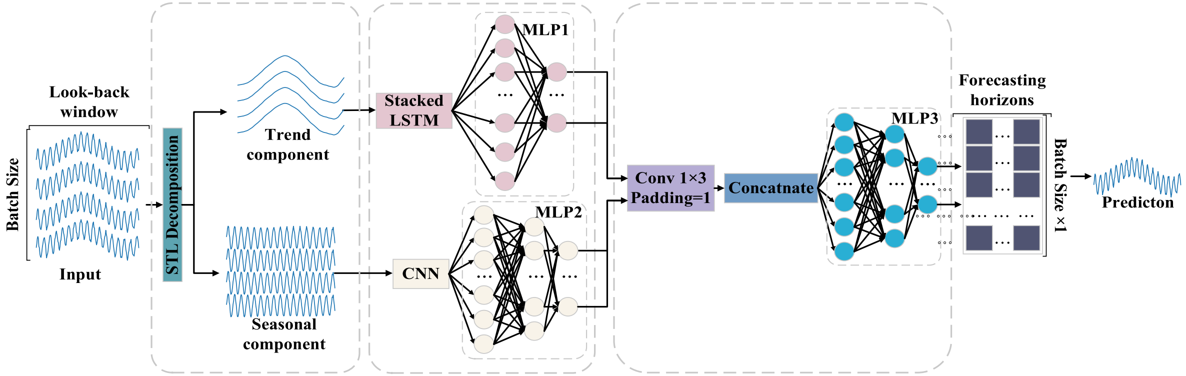


Fig. 4. The architecture of the proposed SCLCC model.

each encoder and decoder pair.

3.2.2. Model structure

In this research, a comprehensive prediction model based on Structured Components, stacked-Long short-term memory, Convolutional neural networks and Comprehensive forecasting module (SCLCC) is proposed to predict carbon emissions in an investigated sea region before and after the emergency events. SCLCC is inspired by various sources, including time series decomposition (Trull et al., 2022; Zeng et al., 2023), Stacked-LSTM (Kabir et al., 2023; Park et al., 2023), and CNN methods (Bogaerts et al., 2020). Fig. 4 illustrates the top-level design of SCLCC, which encompasses the following key components:

(1) Roll-based structural Seasonal-Trend decomposition using Loess (STL)

This study employs a structured decomposition module to enhance time series data by segregating it into trend, seasonal, and residual components. This enhanced data aids in improving the model's learning process (Zeng et al., 2023).

Time series decomposition simplifies complex patterns within the data. For instance, trend analysis facilitates long-term change predictions (Trull et al., 2022). Popular decomposition techniques, such as X11, X12, X13 algorithms based on ARIMA models, and the STL algorithm, are used for this purpose (Qian et al., 2019). Compared to other methods, the STL algorithm is capable of processing a wide range of time series data and exhibits robustness to outliers. It is known for its reliability and strong recognition of seasonal patterns (Tebong et al., 2023). Consequently, the STL algorithm is utilized in this study to decompose the time series data.

The STL method includes an inner and an outer loop, as detailed shown in Table A5. The inner loop separates seasonal and trend components, while the outer loop computes robustness weights, aiding in a robust decomposition of time series data into seasonal, trend, and residual constituents.

The trend component signifies long-term data pattern changes, while the seasonal component captures cyclic fluctuations. The residual component, mostly white noise, remains unpredictable (Qian et al., 2019). In view of the analysis of pronounced trend and seasonality aids in better predictions. Hence, this study focuses on trend and seasonality, disregarding the residual term. Furthermore, it employs a rolling decomposition based on STL during data learning to prevent data leakage. Specifically, the encoder stage in Fig. 5 employs this approach on the look-back time series.

(2) Stacked-LSTM-based Trend Item Learning Module

This module aids the model in acquiring trend-related in-depth information. In this study, it focuses on extracting trend component time series data, simplifying the information (Qian et al., 2019). This approach effectively tackles challenges like long-term dependency and gradient vanishing encountered in TSF (Bogaerts et al., 2020; Kabir et al., 2023). The utilization of Stacked-LSTM (shown in Fig. 6) enhances the model's capability to capture high-dimensional nonlinear data compared to a single LSTM, achieved by stacking multiple LSTM layers (Li *et al.*, 2021a).

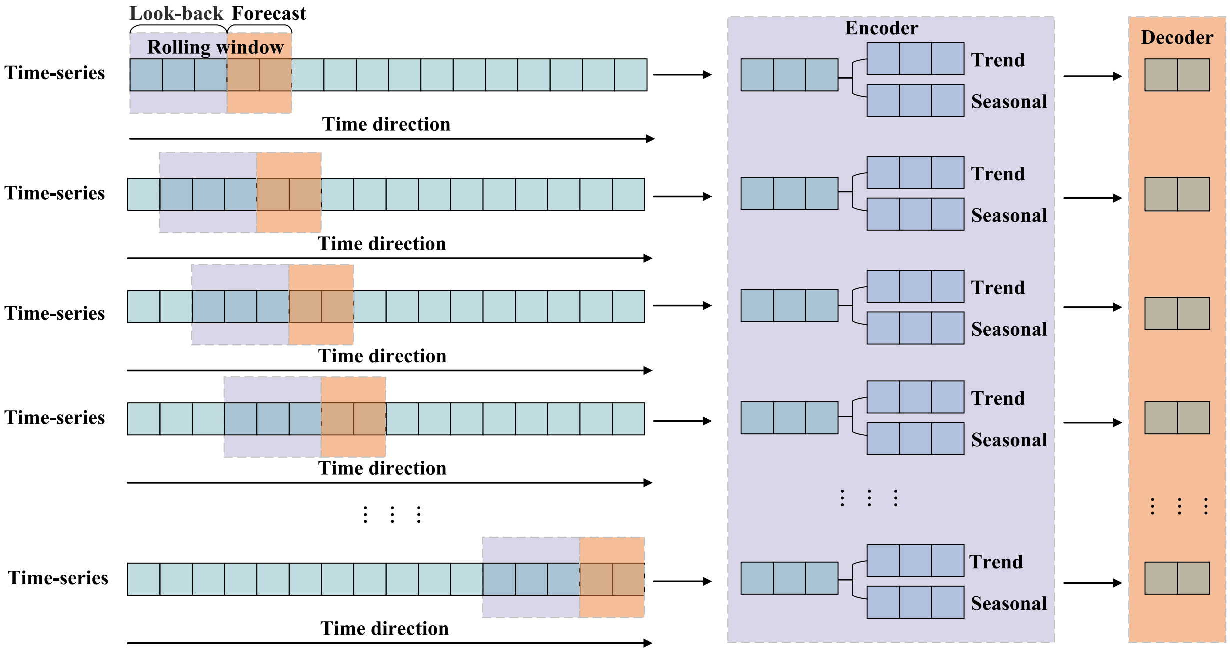


Fig. 5. The roll-based structural STL decomposition.

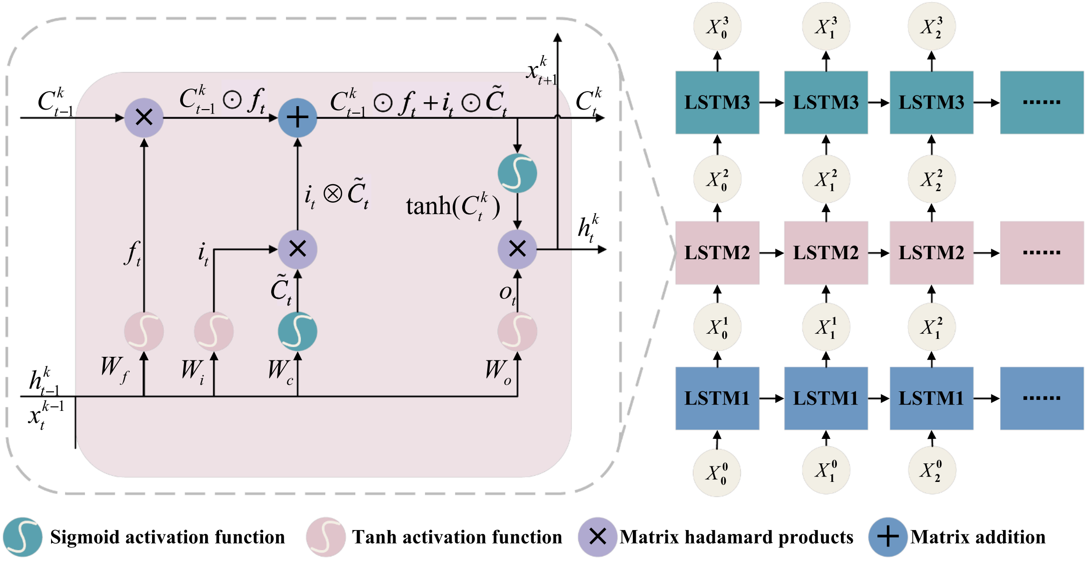


Fig. 6. The Stacked-LSTM architecture.

(3) CNN-Based Seasonal Item Learning Module

This module focuses on understanding seasonal characteristics. CNN, known for capturing spatial dependencies, also demonstrates prowess in extracting intricate temporal dependencies (Hou et al., 2022; Liu et al., 2022). Comprising convolutional and pooling layers, among others (Bogaerts et al., 2020), CNN extracts features from time-series data, where the depth of the convolutional layer correlates with feature abstraction. Meanwhile, the pooling layer selects location-invariant information, broadening the perceptual field of view for subsequent features. In addition, it blurs detailed data, filtering out noise.

Since substantial white noise is removed through STL decomposition in this study, employing pooling during learning could potentially diminish valuable seasonal information, reducing prediction accuracy at the same training cost. Refer to Table C2 in Appendix C for experimental results on pooling's impact on the model. Hence, in this study, CNN without a pooling layer is used to extract seasonal information.

(4) Comprehensive Forecasting Module

Comprehensive Forecasting (CF) module is a holistic unit comprising convolutional, concatenation, and fully connected layers. Its primary role is to merge trend and seasonal data for the final forecast. Employing one-dimensional (1D) convolutional and concatenation layers, it assigns distinct weights to trend and seasonal information, illustrated in Eqs. (14) to (16). Following the acquisition of comprehensive time series data, a fully connected feed-forward neural network is utilized to extrapolate learned insights and predict future carbon emissions, as depicted in Eq. (17).

$$\widetilde{Trend} = \widehat{\mathrm{Trend}} * \mathrm{W}_{Trend} \tag{14}$$

$$\widetilde{Seasonal} = \widehat{Seasonal} * \mathrm{W}_{Seasonal} \tag{15}$$

$$\widetilde{Timeseries} = concat\left[\widetilde{Trend}, \widetilde{Seasonal}\right] \tag{16}$$

$$\widehat{x}_{(t+1):(t+d_{FH})} = \left(\delta\left(\delta\left(\delta\left(\widetilde{Timeseries} \cdot W_{FC1} + b_{FC1}\right) \cdot W_{FC2} + b_{FC2}\right) \cdots\right)\right) \cdot W_{FCn} + b_{FCn} \tag{17}$$

where $\widehat{Treand}$ and $\widehat{Seasonal}$ represent the learned trend and seasonal term information respectively; $\widetilde{Timeseries}$ denotes the integrated time series information; W_{Trend} and $W_{Seasonal}$ indicate the weights of the convolutional layers; $*$ denotes the convolution operation; and $concat[]$ is the concatenate operation; $(W_{FC1}, \cdots, W_{FCn})$ denotes the weight of the fully connected layer; $(b_{FC1}, \cdots, b_{FCn})$ is the bias of the fully connected layer; and $\delta()$ denotes the activation function.

3.2.3. Model measurements and interpretation

This study evaluates the model's performance using multiple metrics: mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and Adjusted R^2 (Alfaseeh et al., 2020; Bogaerts et al., 2020; Kabir et al., 2023). MSE gauges the average of squared deviations between predicted and ground truth values but is sensitive to outliers. MAE calculates the mean of absolute errors and is less affected by outliers but might underestimate error size. MAPE assesses relative error by averaging the ratio between absolute difference and true values. Smaller values of MSE, MAE, and MAPE are preferred. Adjusted R^2 measures model fit to time series data while accounting for the test set size; a value closer to 1 signifies a better fit (Alfaseeh et al., 2020).

To illustrate the model's significant interpretability, this study tends to explain the proposed model from two perspectives. Firstly, analysing the changes in carbon emissions data before and after data enhancement with STL technique, specifically examining the alterations in data correlation and the similarity of time series data using the Spearman correlation test (Han et al., 2018) and the Derivative Dynamic Time Warping (DDTW) algorithm (Górecki and Łuczak, 2015; Li et al., 2024b), respectively. Secondly, interpreting the relative contributions of trend, seasonal terms, and look-back window points to the overall model, represented as ψ_{TS}, ψ_{LB}. The study maintains a relative contribution range between 0–100 % and primarily conducts global sensitivity experiments to evaluate the impact of randomly shuffling training data with trend or seasonal terms on the model's performance (Hu and Xiong, 2023). To comprehensively assess the extent of this impact, the study calculates the average changes in the previously mentioned evaluation metrics to determine the absolute contribution, i.e., ω_{TS}, ω_{LB}. This value is subsequently converted into the relative contribution using Eq. (18).

$$\psi_{TS}^{k} = \frac{\omega_{TS}^{k}}{\sum \omega_{TS}^{k}}, \psi_{LB}^{k} = \frac{\omega_{TS}^{k}}{\sum \omega_{LB}^{k}} \tag{18}$$

where ψ_{TS}^{k} represents the relative contribution of the trend or seasonal component to the model; ω_{TS}^{k} is the absolute contribution of the trend or seasonal component to the model; ψ_{LB}^{k} represents the relative contribution of the k^{th} time point in the look-back time window to the model; and ω_{LB}^{k} denotes the absolute contribution of the k^{th} time point in the look-back time window to the model.

4. Experiments

This section covers the construction and description of the SCE database and the training of the TSF model using the constructed SCE database. In Section 4.1, the effectiveness of the STSA algorithm in identifying ship navigation states is verified, and the created SCE database in Bohai Bay is detailed and described. In Section 4.2, the training process of the time series model, encompassing baseline model selection, hyper-parameter tuning, and model validation, is outlined.

4.1. Carbon emission data

The Bohai Bay, part of China's crucial economic bay areas and situated near the North China Plain, hosts one of the nation's three prominent port clusters (Wan et al., 2020). However, rapid economic growth in this region has led to significant environmental challenges, notably concerning ship emissions, impacting regional air quality and human health (Wan et al., 2023). This study collected ship AIS data in Bohai Bay's waters from January 1, 2019, to December 31, 2020, using the ESSNA method to establish the SCE database. Notably, the IMO Fourth Greenhouse Gas Study (International Maritime Organization, 2020) highlights that ship carbon emissions encompass both CO_2 and Black Carbon (BC). However, CO_2 emissions significantly outweigh BC emissions, making CO_2 the primary greenhouse gas emitted by ships (International Maritime Organization, 2021; Weng et al., 2020). Consequently, this study focuses on constructing a database exclusively for CO_2 emissions from ships.

4.1.1. SCE dataset construction

When constructing the dataset using the ESSNA method, accurate identification of the ship's navigational state is critical. To evaluate the proposed STSA algorithm's effectiveness in this regard, ten ship trajectories, with a total of 213,572 points, were randomly selected from AIS data. Seven trajectories were used for adjusting the fundamental parameters, while three served as test cases. In this study, accuracy and F1-score were used as evaluation metrics (Herrema et al., 2019).

The optimal fundamental parameters for the STSA algorithm were determined through multiple experiments, as shown in Table 2, and the comparative effectiveness result of the STSA algorithm and the International Maritime Organization (2020) approach using the optimal parameters is illustrated in Table 3. It is evident from Table 3 that the STSA algorithm outperforms the International Maritime Organization (2020) method in identifying the navigational state of ships.

To further compare the effectiveness of the two methods in identifying ship navigational states, this study visualizes the results in each of the five scenarios (i.e., ship navigation states), as depicted in Fig. 7. In Scenario 1, Fig. 7 (a1) and (a2) show that while the ship is in a state of maneuvering navigation, the STSA algorithm effectively recognizes the maneuvering navigation. In contrast, International Maritime Organization (2020) incorrectly identifies most trajectories as the anchoring state, which could be due to International Maritime Organization (2020) stereotypically defining maneuvering as speeds greater than 3 knots, overlooking low-speed

Table 2
STSA optimum parameters.

Parameters	Value	Parameters	Value	Parameters	Value
$\xi_{dis} = (\xi_a, \xi_b)$	$(300m, 50m)$	η	10°	ϑ	5
$\tau_s = (\tau_a, \tau_b)$	$(1h, 12h)$	λ	15		

Table 3
Comparison of the effectiveness of navigation state identification models.

Method	Accuracy	F1-score
STSA	0.9128	0.9036
International Maritime Organization (2020)	0.6834	0.6529

maneuvering states.

Scenarios 2 and 3 illustrate the ship's departure from the berth and its anchoring, respectively. These scenarios reveal that the distance covered by the ship in the berthing state is significantly less than the distance covered during the anchoring state. Moreover, Fig. 7 b1) and b2) demonstrate that during departure from the port, the ship's speed is lower, and it must continually adjust its heading. Therefore, this phase should be identified as maneuvering navigation. The International Maritime Organization (2020) erroneously identifies a section of the trajectory as anchoring. At the same time, from Fig. 7 c1) and c2), it is clear that the STSA algorithm accurately identifies the anchoring state. In contrast, the International Maritime Organization (2020) fails to identify the majority of the trajectory states in the anchorage area as berthing states. It is evident that the STSA algorithm can better differentiate between anchoring and berthing using the duration and geographical range of the ship's trajectory, than the International Maritime Organization (2020) standard, which only considers speed for distinguishing between these two states.

Scenarios 4 and 5 depict the identification of navigational states by the two methods during the slow-steaming phase and the normal cruising phase, respectively. Fig. 7 d1) and d2) reveal that when the ship is in the slow-steaming phase, International Maritime Organization (2020) incorrectly identifies the trajectory of the cruising state as a maneuvering sailing state, whereas the STSA can effectively identify the cruising state. Meanwhile, since there are several consecutive trajectory points with a speed greater than 12 in this scenario, the STSA algorithm determines these consecutive trajectory points as normal cruising. Fig. 7 e1) and e2) show that in the normal cruising phase, the International Maritime Organization (2020) simply and roughly identifies some of the trajectory points with speeds less than or equal to 12 knots as slow-steaming, without considering that these may be a few abnormal speed points occurring in the normal cruising state. Furthermore, the continuity of the cruising state is not taken into account. In contrast, the STSA algorithm can ignore a few abnormal points and identify a continuous normal cruising state trajectory.

Following the identification of navigational states, this study measured the carbon emissions of each ship with a time granularity of 1 h based on the STEAM model, ultimately resulting in the creation of the SCE database.

4.1.2. Dataset description and analysis

To better discern variations in carbon emissions in Bohai before and after the COVID-19 outbreaks, this study used January 12, 2020—the date of the World Health Organization's declaration of the virus—as the dividing point. The SCE database, with hourly data points spanning 2019–2020, encompassed a total of 17,544 points. Fig. 8 illustrates carbon emissions from ships in Bohai and shows emission changes following STL decomposition. The outbreak led to a notable decline in emissions due to reduced ship activity. As shown in Table 4, post-outbreak emissions were lower but more variable, suggesting higher ship activity before the outbreak (Wang et al., 2023). For assessing CD in ship emissions, Spearman's correlation test and the Kolmogorov-Smirnov (KS) test were conducted. Table 5 depicts the correlation and distribution changes before and after the epidemic. The results indicate a significant 99 % reliability level. The Spearman test suggests a weak temporal correlation between emissions before and after the epidemic. The KS test confirms a distributional difference, suggesting the epidemic-induced CD in ship emissions (Karimian and Beigy, 2023). To highlight the SCLCC model's superiority, pre-epidemic emissions were used as training data (covering the period from January 1, 2019, at 00:00 to January 12, 2020, at 23:00), the first month after the outbreak as test data (encompassing January 13, 2020, at 00:00 to February 12, 2020, at 23:00), and the second month as validation data (spanning from February 13, 2020, at 00:00 to March 12, 2020, at 23:00).

4.2. Model comparison experimental design

4.2.1. Forecasting comparison

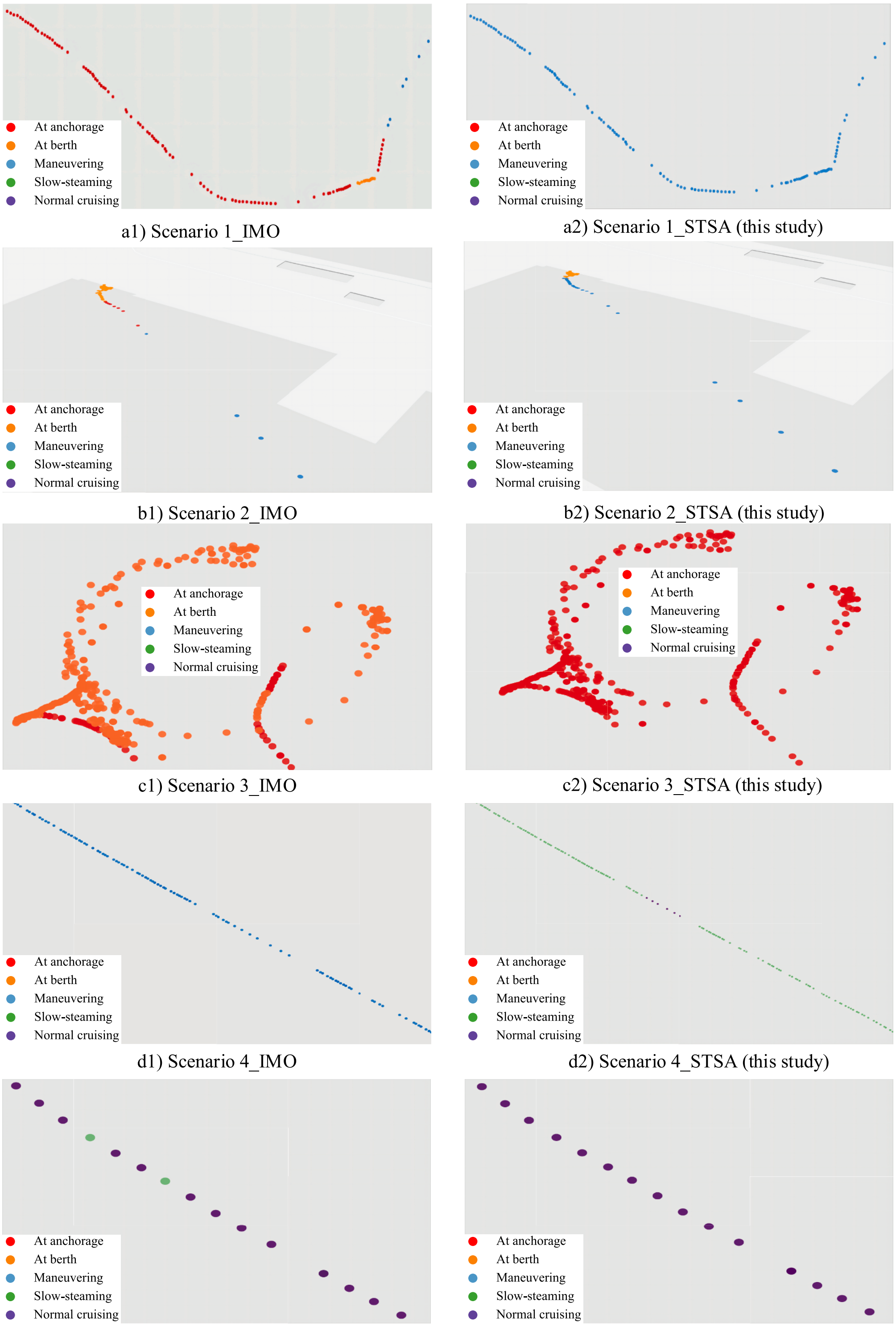
To assess the superior effectiveness of SCLCC, this study conducts a comparative analysis of its predictive accuracy with several advanced TSF models. These baseline models encompass a decision tree-based machine learning model and several advanced deep learning models, as shown in Table 6.

To address this limitation of XGBoost's multi-step prediction, this study employs a recursive approach (Hu and Xiong, 2023), iteratively generating single-step predictions, with each prediction serving as input for the next step's forecasting.

4.2.2. Hyper-parameter tuning

This study rigorously tuned all models' hyper-parameters within a fixed training budget of 100 epochs, considering the significant sensitivity of deep learning models to these parameters (Bogaerts et al., 2020; Hu and Xiong, 2023). Seven look-back windows (4, 8, 12, 18, 24, 36, and 48 h) were examined for each model, while maintaining a consistent one-hour forecasting horizon. The predictive capabilities across seven forecasting horizons (1, 2, 4, 6, 8, 10, and 12 h) were evaluated, with a fixed 12-hour look-back window. A grid search algorithm systematically explored the hyper-parameter space to determine the optimal parameters for each model.

Fig. 9 depicts a portion of the hyper-parameter tuning process for the SCLCC model using the grid search algorithm. Different colors represent distinct training processes with varied hyper-parameter configurations. Solid lines represent training set errors, while dashed lines signify validation set errors. Details on the hyper-parameter search range and optimal configurations for different look-back



a1) Scenario 1_IMO

a2) Scenario 1_STSA (this study)

b1) Scenario 2_IMO

b2) Scenario 2_STSA (this study)

c1) Scenario 3_IMO

c2) Scenario 3_STSA (this study)

d1) Scenario 4_IMO

d2) Scenario 4_STSA (this study)

e1) Scenario 5_IMO

e2) Scenario 5_STSA (this study)

Fig. 7. The comparative results of ship navigation state detection in five scenarios.

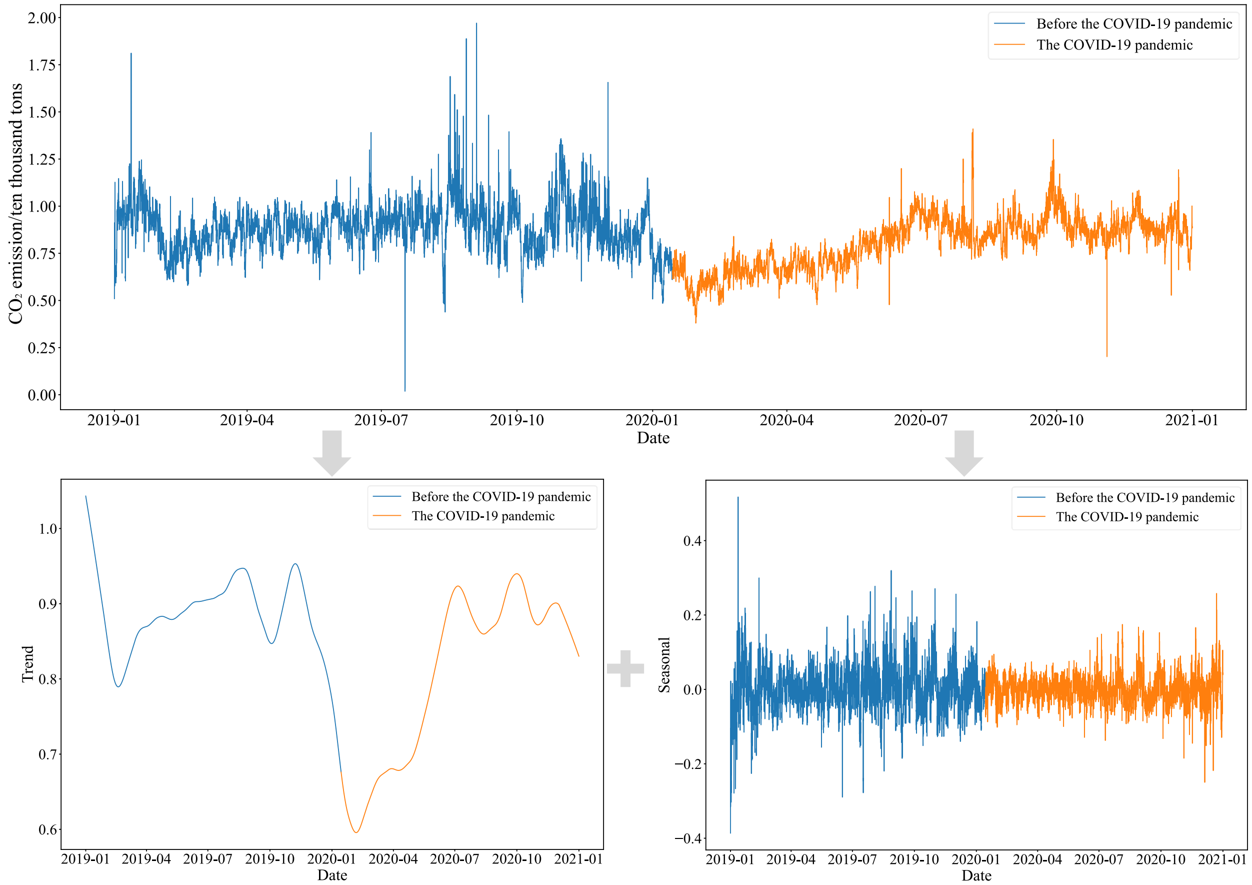


Fig. 8. Changes in carbon emissions in the Bohai Sea before and after the outbreak of the COVID-19 pandemic. *Note: The raw data are mainly decomposed into Trend items and Seasonal items.

Table 4
Basic description of data.

Data	All	Before the COVID-19	After the COVID-19
Count	17,544	9084	8460
Mean	0.8441	0.8812	0.8041
Standard deviation	0.1343	0.1226	0.1348
Maximum	1.9709	1.9709	1.4097
Minimum	0.0186	0.0186	0.2018
25 %	0.7584	0.8018	0.6982
50 %	0.8572	0.8846	0.8274
75 %	0.9307	0.9562	0.8978
Skewness	−0.0468	0.2046	−0.0372
Kurtosis	1.4633	3.6327	0.3552

Table 5
Statistical analysis of carbon emissions data before and after the COVID-19 epidemic.

Test	Spearman correlation coefficient	P-value	–	KS	
				KS statistics	P-value
Before-After	0.1172	2.95E-27		0.2476	1.07E-228

Note: Both are 99% significant.

Table 6
Baseline models.

Model	Features	Fitness	Advantages	Disadvantages	References
XGBoost	Boosting framework	High-dimensional multivariate time series data	Highly interpretable	Limited to single-step predictions	(Luo et al., 2023; Yan et al., 2023; Feng et al., 2024)
MLP	Forward-structured ANN	Relatively simple time series tasks	Highly non-linear global action	Learning may be inadequate	(Abdelaty et al., 2021; Damman and Steen, 2021)
RNN	Recurrent units	Short-term dependent time series tasks	Short-term memory	Gradient disappearance or explosion	(Park et al., 2023; Servizi et al., 2020)
Stack-LSTM	Memory gate, forgetting gate, and output gate	High-dimensional time series data	Long-term memory	Long training time	(Kabir et al., 2023; Li et al., 2021; Park et al., 2023; Zhu and Li, 2021)
CNN	Convolutional operations	Extracting intricate temporal dependencies from time series data	Local feature extraction	Pooling layer loses some valuable information	(Bogaerts et al., 2020; Hou et al., 2022; Liu et al., 2022)
Transformer	Self-attention	Long sequence modeling	Strong modeling capabilities and versatility	Local information acquisition is not as strong as RNN and CNN	(Hu and Xiong, 2023; Zeng et al., 2023)

Note: Multilayer Perceptron (MLP).

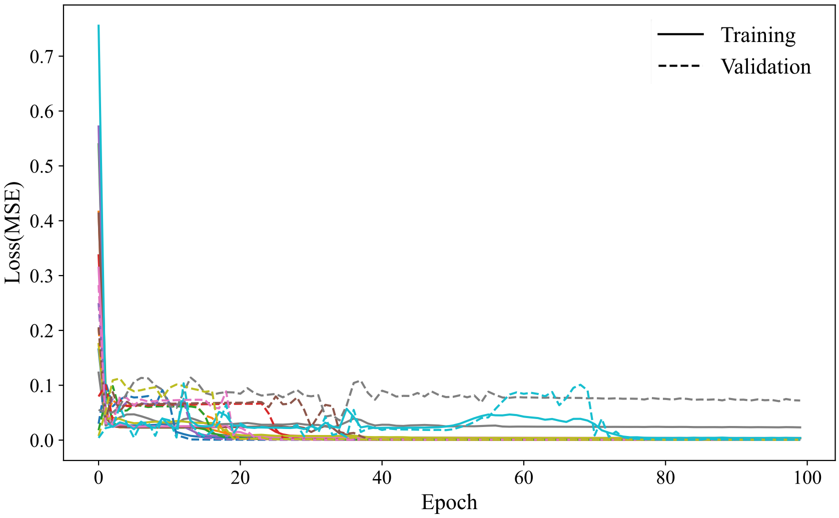


Fig. 9. Hyper-parameter tuning process using Grid-Search. Model = SCLCC, number of epochs = 100, batch size = 720, d_{FH} = 1, d_{LB} = 12. One epoch means completing the model parameter update using all training data.

windows and forecasting horizons of the SCLCC model can be found in Table 7.

4.2.3. *Training, validation, and testing implement*

As presented in Section 4.2.2, this study leverages the Adam algorithm with the objective of minimizing model loss, quantified as MSE. All the models, including SCLCC, Stacked-LSTM, MLP, RNN, CNN, and Transformer, were developed within the *Python* environment using *PyTorch* (Hu and Xiong, 2023). Additionally, XGBoost was implemented utilizing *Scikit-learn* (Luo et al., 2023; Yan et al., 2023).

Fig. 10 depicts the learning curve for SCLCC during training with the optimal parameters. Both the training and validation set losses show reduction and gradual smoothing over 1300 training steps. The minimal discrepancy between the losses of the training and validation sets indicates that SCLCC adeptly captures the nonlinear mapping relationship between the encoder and decoder. Learning curves are essential for identifying overfitting or underfitting (Park et al., 2023). The close resemblance between the training set loss and validation set loss curves (Fig. 10) suggests that the SCLCC model is well-fitted, avoiding overfitting or underfitting during training.

Table 7
Best hyper-parameter configuration.

Parameter		Hidden layers size	Stacked-LSTM layers size	Learning rate	Optimizer
Search Range		32, 64, 128, 256, 512	2, 3, 4, 5	0.001, 0.002, 0.003	SGD, Adam, Adamax, NAdam
Best Value					
$d_{LB} = 4$	$d_{FH} = 1$	256	2	0.002	Adam
$d_{LB} = 8$	$d_{FH} = 1$	64	2	0.001	Adam
$d_{LB} = 12$	$d_{FH} = 1$	128	2	0.001	Adam
$d_{LB} = 18$	$d_{FH} = 1$	512	3	0.003	Adam
$d_{LB} = 24$	$d_{FH} = 1$	128	3	0.003	Adam
$d_{LB} = 36$	$d_{FH} = 1$	128	4	0.001	Adam
$d_{LB} = 48$	$d_{FH} = 1$	256	3	0.001	Adam
$d_{LB} = 12$	$d_{FH} = 2$	128	3	0.002	Adam
$d_{LB} = 12$	$d_{FH} = 4$	128	3	0.001	Adam
$d_{LB} = 12$	$d_{FH} = 6$	128	4	0.001	Adam
$d_{LB} = 12$	$d_{FH} = 8$	256	3	0.001	Adam
$d_{LB} = 12$	$d_{FH} = 10$	128	2	0.001	Adam
$d_{LB} = 12$	$d_{FH} = 12$	128	2	0.002	Adam

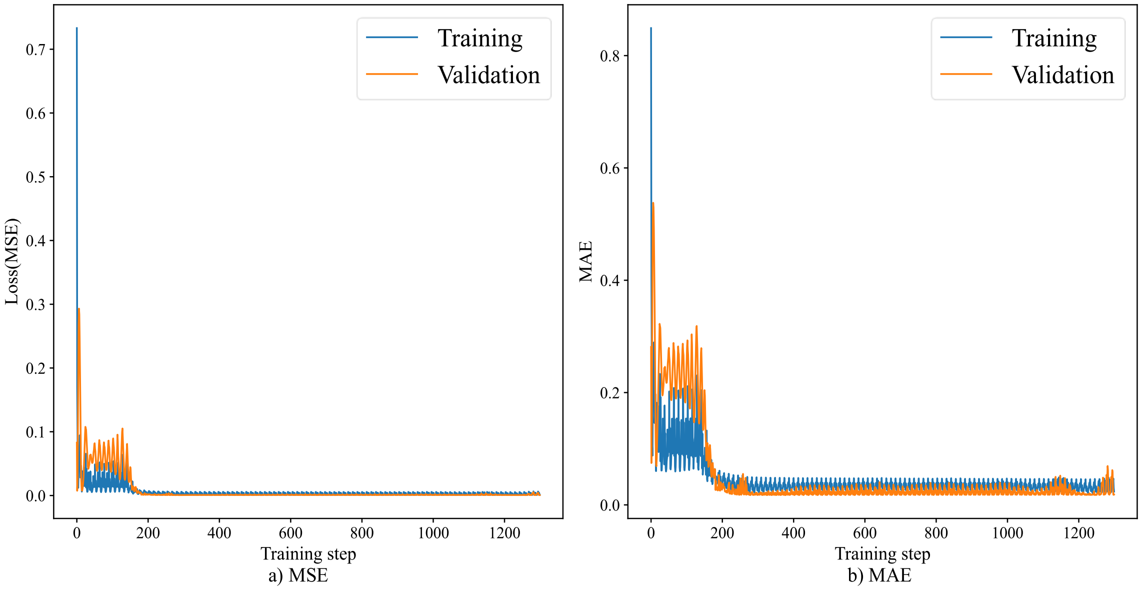


Fig. 10. Learning curve using the best configuration. Model = SCLCC, number of epochs = 100, batch size = 720, $d_{FH} = 1$, $d_{LB} = 12$. One training step means completing the model parameters update using one batch of training data.

5. Result analysis and discussion

5.1. Forecasting performance

5.1.1. Performances across models

Due to the extensive comparative experimental data, the main text focuses on the predictive results for a 12-hour look-back window and a 1-hour forecasting horizon, as indicated in Table 8. Additional comparative experiment results are available in Appendix D. Table 8 demonstrates SCLCC's remarkable ability to predict changes in ship carbon emissions following the epidemic outbreak. In this

Table 8
Model performances comparison if $d_{LB} = 12$ and $d_{HF} = 1$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00299 (406.78 %)	0.0382 (104.28 %)	5.8646 (88.86 %)	0.3864 (55.60 %)
MLP	0.00172 (191.53 %)	0.0345 (84.49 %)	5.5195 (77.75 %)	0.5172 (40.56 %)
RNN	0.00131 (122.03 %)	0.0362 (93.58 %)	4.8061 (54.78 %)	0.6660 (23.47 %)
Stacked-LSTM	0.00177 (200.00 %)	0.0358 (91.44 %)	5.7007 (83.59 %)	0.5421 (37.30 %)
CNN	0.00124 (110.17 %)	0.0352 (88.24 %)	4.3510 (40.12 %)	0.7029 (19.23 %)
Transformer	0.00147 (149.15 %)	0.0308 (64.71 %)	4.9617 (59.79 %)	0.6335 (27.20 %)
SCLCC	**0.00059**	**0.0187**	**3.1052**	**0.8702**
Improvement	**110.17 %**	**88.24 %**	**40.12 %**	**19.23 %**

Note: The percentages in brackets represent changes in metrics relative to SCLCC. The cell marked with an underline indicates the best baseline method based on the metric in that column. “Improvement” refers to the percentage increase in performance of SCLCC compared to the best baseline model.

setting, SCLCC achieves an impressive Adjusted R^2 value of 87.02 %. Additionally, a comprehensive analysis of Table 8 and the results in Appendix D reveals that SCLCC outperforms all the baseline models. When compared to the optimal baseline model, SCLCC achieves a significant 8.39 % to 110.17 % reduction in MSE, a 3.90 % to 88.24 % decrease in MAE, a 3.10 % to 40.12 % decrease in MAPE, and a 6.00 % to 90.30 % improvement in Adjusted R^2. These findings sufficiently prove that SCLCC is a robust predictor for this type of data.

When comparing all baseline models, it is clear that CNN and Transformer models consistently exhibit the highest predictive accuracy. This higher accuracy stems from their ability to capture more information from time series data, facilitated by the convolutional operation and the multi-head attention mechanism (Liu et al., 2022). In contrast, the RNN and Stacked-LSTM models, notwithstanding being designed for TSF, display moderate performance. The MLP model shows relatively good prediction accuracy for single-step predictions but faces challenges in multi-step predictions due to its limitations in capturing comprehensive time series information. With XGBoost, its performance excels with short-duration look-back windows and forecasting horizons but deteriorates as

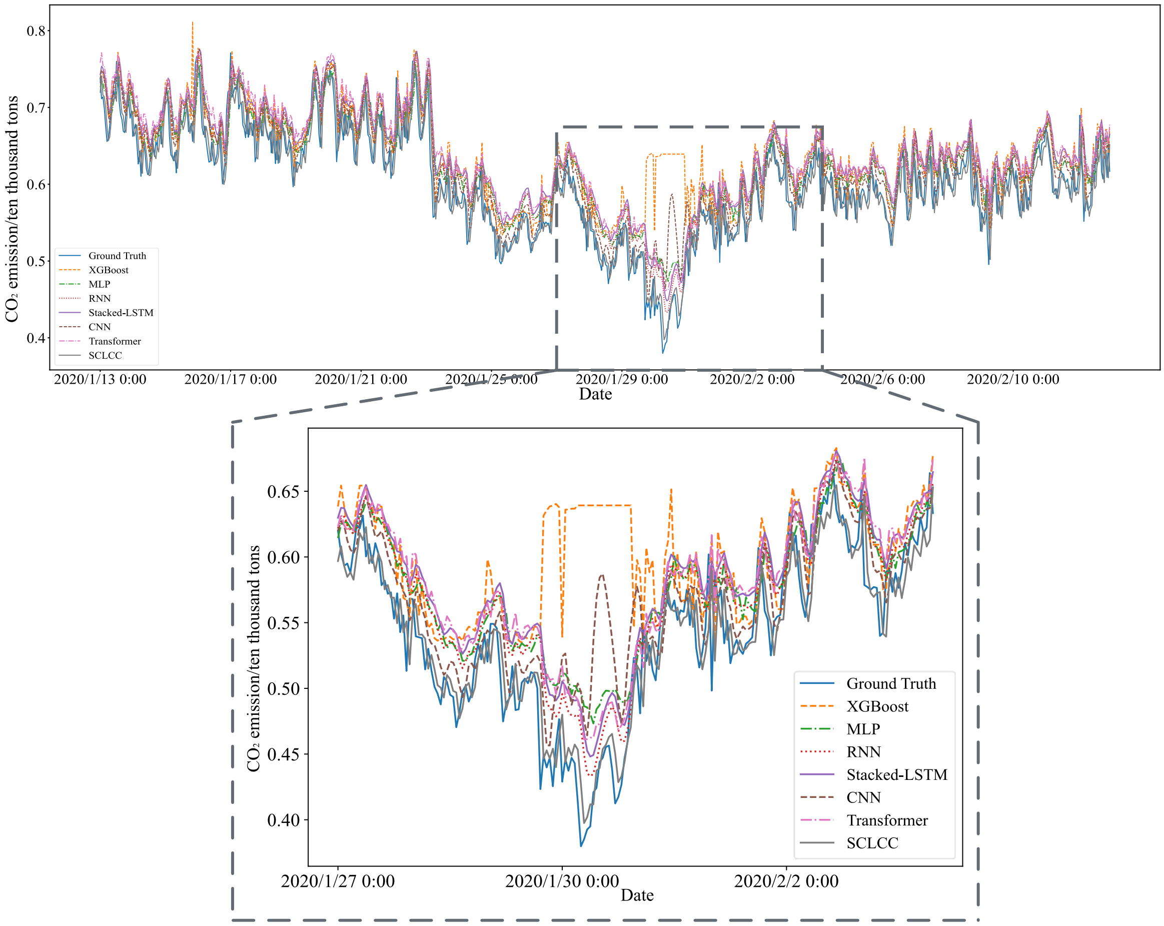


Fig. 11. Comparison of ground truth and predicted values across models ($d_{LB} = 12$ and $d_{HF} = 1$).

these timeframes increase. This could be due to its struggle to capture complex global temporal information and its tendency to give excessive weight to limited time nodes. Additionally, XGBoost's recursive generation training approach leads to error accumulation in multi-step outputs.

Fig. 11 visually presents the prediction outcomes for a 12-hour look-back window and a 1-hour forecasting horizon. Overall, the SCLCC model aligns most closely with the ground truth. It is notable that, during the period from 2020 to 01–27 to 2020–02–03, deviations between predictions for ship carbon emissions and ground truth values are more noticeable, especially for XGBoost and CNN models.

Moreover, to demonstrate the effectiveness of the SCLCC model across different datasets, it is applied to global maritime carbon emissions data. The results, presented in Table C1 and Figs. C1 and C2 in Appendix C, further highlight the model's superiority.

5.1.2. *Performances across look-back window*

Table 9 depicts how altering the size of the look-back window impacts the model's prediction performance. The optimal window size is crucial for both evaluating the model's performance and training an effective prediction model. A short window might hinder the learning of critical time series information, while an excessively long window can introduce redundancy, reducing model efficiency.

Across the extensive data in Appendix D and Table 8, it's consistent that SCLCC outperforms all other models, regardless of the window size. This indicates SCLCC's proficiency in extracting relevant information for future predictions, irrespective of redundant data volume. Specifically, Table 9 highlights that SCLCC achieves its peak predictive performance with a 12-hour look-back window.

5.1.3. *Performances across forecasting horizons*

Table 10 presents a clear negative correlation between the model's predictive ability and the forecasting horizon, and Fig. 12 visually depicts the SCLCC model's predictive performance at various forecasting horizons and time points. Two significant observations can be made based on the data from Table 10 and Fig. 12:

1) There is a notable decline in model performance as the forecasting horizon increases. For instance, when the prediction horizon extends from 1 to 6 days, the MAPE of SCLCC increases by 50.91 %, rising from 3.1052 % to 4.6860 %.

2) Within a constant forecasting horizon, longer time periods pose increased prediction challenges. For instance, at a forecasting horizon of 8, the MAPE for SCLCC surges by 100.79 %, escalating from 3.3192 % to 6.8998 %.

These observations underscore the known difficulties in long-term predictions compared to short-term ones (Bogaerts et al., 2020), stemming from amplified white noise accumulation as the prediction horizon extends. This accumulation inevitably reduces the forecasting model's effectiveness (Hu et al., 2021; Hu and Xiong, 2023).

5.2. *Ablation experiment*

In the realm of deep learning, ablation experiments are used to dissect a model's performance by altering or removing specific components, offering insights into their individual contributions (Liu et al., 2022; Zeng et al., 2023). For these experiments to be informative, a model should gracefully degrade, maintaining a high level of functionality even when components are altered (Zeng et al., 2023). To comprehensively understand the role of each constituent within the SCLCC model, this study conducted ablation experiments where it removed the Stacked-LSTM component from the trend-term structure, the CNN component from the seasonal-term structure, and the CF structure, resulting in a model structure similar to the Dlinear model when these three components are simultaneously omitted, as depicted in Fig. E1 in Appendix E. The significance of the STL decomposition strategy can be found in Table 8 and Appendix D. Additionally, to assess the impact of residual terms with high levels of white noise on the model's performance, this study sought to construct a model with a residual structure, as illustrated in Fig. E2 in Appendix E. The results of these ablation experiments are presented in Table 11 and Fig. 13.

Results in Table 11 and Fig. 13 indicate a moderate decline in SCLCC's performance with specific component removal or residual structure inclusion. Notably, omitting the CNN from the seasonal structure affects the model most, likely due to intricate information in seasonal terms not effectively captured by a feedforward network. Interestingly, even with all components removed, SCLCC outperforms models like the Transformer (Zeng et al., 2023). However, adding a residual structure leads to reduced performance, suggesting that the residual term contains substantial white noise, hindering effective learning.

Table 9
Comparison of different look-back windows.

Look-back window	MSE	MAE	MAPE (%)	Adjusted R^2
4	0.00066	0.0199	3.2558	0.8700
8	0.00065	0.0196	3.2622	0.8613
12	**0.00059**	**0.0187**	**3.1052**	**0.8702**
18	0.00081	0.0219	3.6414	0.7984
24	0.00070	0.0211	3.6177	0.8662
36	0.00063	0.0193	3.2008	0.8360
48	0.00071	0.0208	3.3988	0.8402

Table 10
Comparison of different forecasting horizons.

Forecasting horizons	MSE	MAE	MAPE (%)	Adjusted R^2
1	**0.00059**	**0.0187**	**3.1052**	**0.8702**
2	0.00078	0.0214	3.5558	0.8063
4	0.00108	0.0255	4.2263	0.7215
6	0.00132	0.0284	4.6860	0.6046
8	0.00155	0.0308	5.0718	0.5141
10	0.00185	0.0336	5.4848	0.2986
12	0.00202	0.0353	5.6854	0.3155

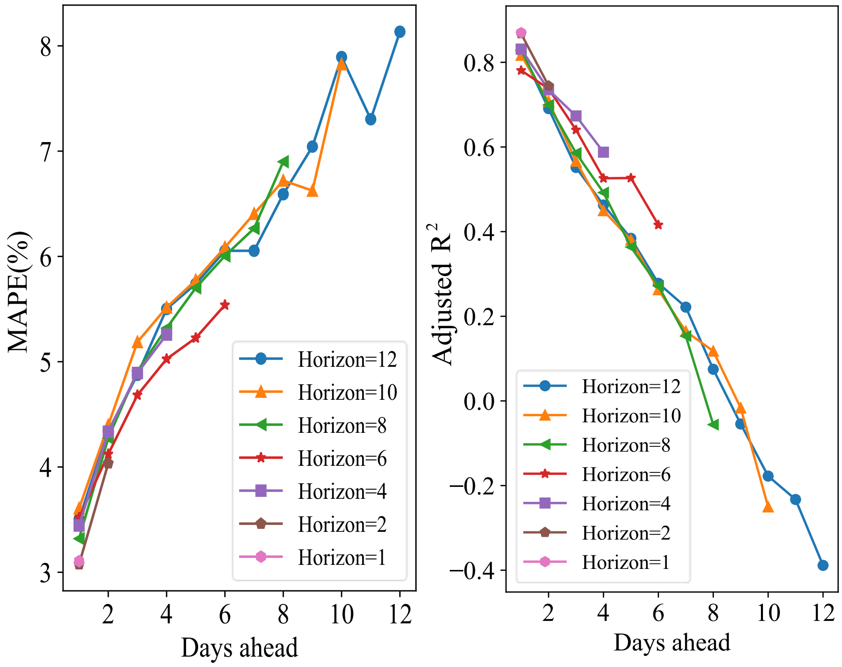


Fig. 12. The comparison results of different forecasting horizons.

Table 11
Ablation experiment results if $d_{LB} = 12$ and $d_{HF} = 1$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
SCLCC (baseline)	**0.00059**	**0.0187**	**3.1052**	**0.8702**
+Residual	0.00072 (−22.03 %)	0.0206 (−10.16 %)	3.4514 (−11.15 %)	0.7987 (−8.22 %)
−Stacked-LSTM	0.00072 (−22.03 %)	0.0211 (−12.83 %)	3.4116 (−9.87 %)	0.8455 (−2.84 %)
–CNN	0.00073 (−23.73 %)	0.0205 (−9.63 %)	3.4561 (−11.30 %)	0.7942 (−8.73 %)
−CF	0.00079 (−33.90 %)	0.0224 (−19.79 %)	3.6495 (−17.53 %)	0.8098 (−6.94 %)
−Stacked-LSTM and CNN	0.00073 (−23.73 %)	0.0213 (−13.90 %)	3.4703 (−11.76 %)	0.8322 (−4.37 %)
−Stacked-LSTM, CNN and CF	0.00092 (−55.93 %)	0.0303 (−62.03 %)	3.9780 (−28.11 %)	0.7615 (−12.49 %)

Note: Percentages in brackets are the change in metrics versus SCLCC (baseline).

5.3. Examining changes in predictive ability of models before and after COVID-19

The COVID-19 pandemic exemplifies how a prolonged outbreak can significantly alter the trajectories of various industries, potentially impacting the performance of TSF models (Li et al., 2021; Liu et al., 2021; Mroua and Lamine, 2023). To gauge the effect of this outbreak on model performance, this section uses data from January 1, 2019, to November 30, 2019, for training and data from December 1, 2019, to December 31, 2019, for testing to assess the model's pre-outbreak forecasting ability. Given the test data's influence on metrics like MSE and MAE (Hou et al., 2022), this study focuses on comparing MAPE and Adjusted R^2 before and after the outbreak (Alfaseeh et al., 2020).

This section compares model performance before and after the outbreak using a 24-hour look-back window and a 1-hour forecasting horizon, detailed in Table 12. The results in Table 12 indicate a noticeable decline in the models' performance due to the

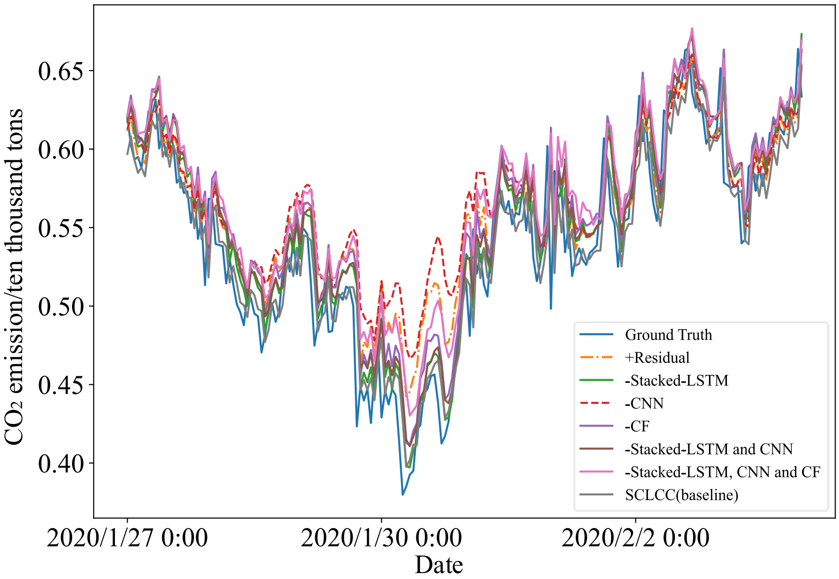


Fig. 13. Ablation experiment visualization results.

Table 12
Changes in the predictive ability of models before and after COVID-19.

Model	Before COVID-19		After COVID-19		Change	
	MAPE	Adjusted R^2	MAPE	Adjusted R^2	MAPE	Adjusted R^2
XGBoost	3.9801	0.8968	14.0339	0.0520	252.60 %	94.20 %
MLP	3.7360	0.8845	6.1725	0.4078	65.22 %	53.89 %
RNN	3.7608	0.8817	5.1921	0.6343	38.06 %	28.06 %
Stacked-LSTM	3.9994	0.8637	5.2434	0.5166	31.10 %	40.19 %
CNN	**3.2360**	**0.9192**	3.9797	0.7207	22.98 %	21.59 %
Transformer	3.5142	0.9113	6.0772	0.4571	78.63 %	49.84 %
SCLCC	3.4545	0.9039	**3.6177**	**0.8662**	**4.72 %**	**4.17 %**

Table 13
Change in correlation before and after decomposition.

Data	All data(before-after)			Jan. 2019-Jan. 2020			Dec. 2019-Jan. 2020		
	Raw	Trend	Season	Raw	Trend	Season	Raw	Trend	Season
R	0.1172	0.3179	0.1269	0.1666	0.8238	0.2974	−0.1143	0.8238	0.8151
P	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
DDTW	226.43	0.34	136.55	18.37	0.06	14.24	12.95	0.14	7.19

Note: R is the correlation coefficient; P is the P-value.

epidemic’s impact on the shipping industry. Reduced ship activity and altered emission patterns led to decreased predictive accuracy. Before the outbreak, the CNN, Transformer, and SCLCC models performed similarly well. However, post-epidemic, SCLCC emerged as the most accurate predictor. Notably, SCLCC demonstrated the least reduction in prediction accuracy compared to others, highlighting its resistance to such disruptive events.

5.4. Interpretation

5.4.1. Changes in data relationships before and after STL decomposition

Table 13 summarizes the impact of STL decomposition on the correlation and similarity measures within the data. The results indicate notable improvements in correlation strengths between trend and seasonal terms compared to the original dataset. Notably, the trend term has exhibited the most significant enhancement in correlation, possibly due to its simpler information content in contrast to the seasonal term. As shown in Table 13 and Fig. 14, the correlation between the seasonal terms in the data for December 2019 and January 2020 is stronger than the correlation between the seasonal terms in the data for January 2019 and January 2020. This could be attributed to the proximity of December 2019 to January 2020.

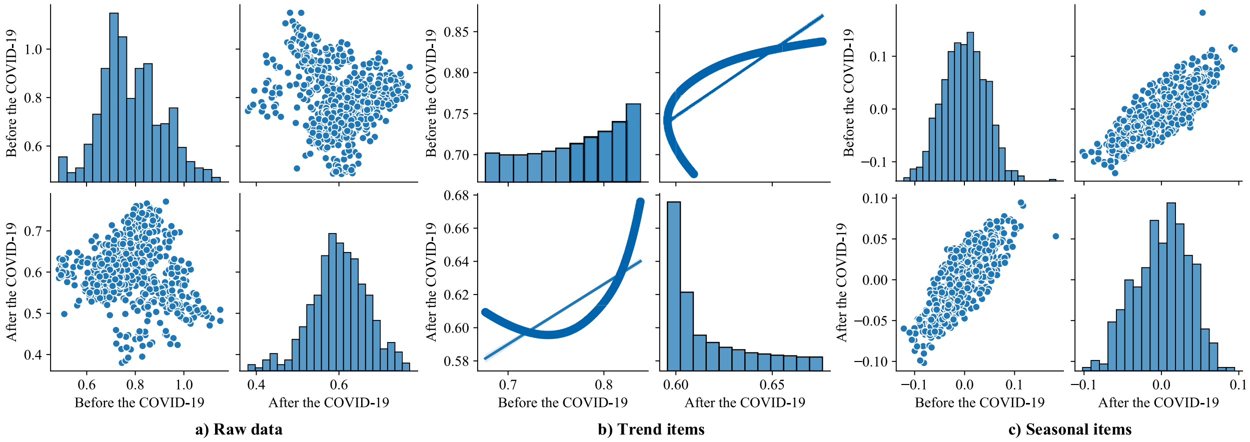


Fig. 14. Statistical distribution and correlation diagrams of carbon emissions from ships Dec. 2019-Jan.2020.

Table 13 also demonstrates that decomposing the data using STL improves the similarity of time series data, indicated by the decreasing DDTW values. Reduced DDTW values signify increased similarity in the time series data (Górecki and Łuczak, 2015). Notably, the trend term exhibits the highest level of similarity. When comparing the carbon emission data for December 2019 and January 2020, the trend term shows greater similarity in the January 2019 and January 2020 data. This implies that the overall trend of vessel activity in these same months is more alike. Moreover, the comparison of seasonal terms between the data for December 2019 and January 2020 indicates greater similarity compared to the data for January 2019 and January 2020. This suggests more analogous seasonal fluctuations in consecutive months.

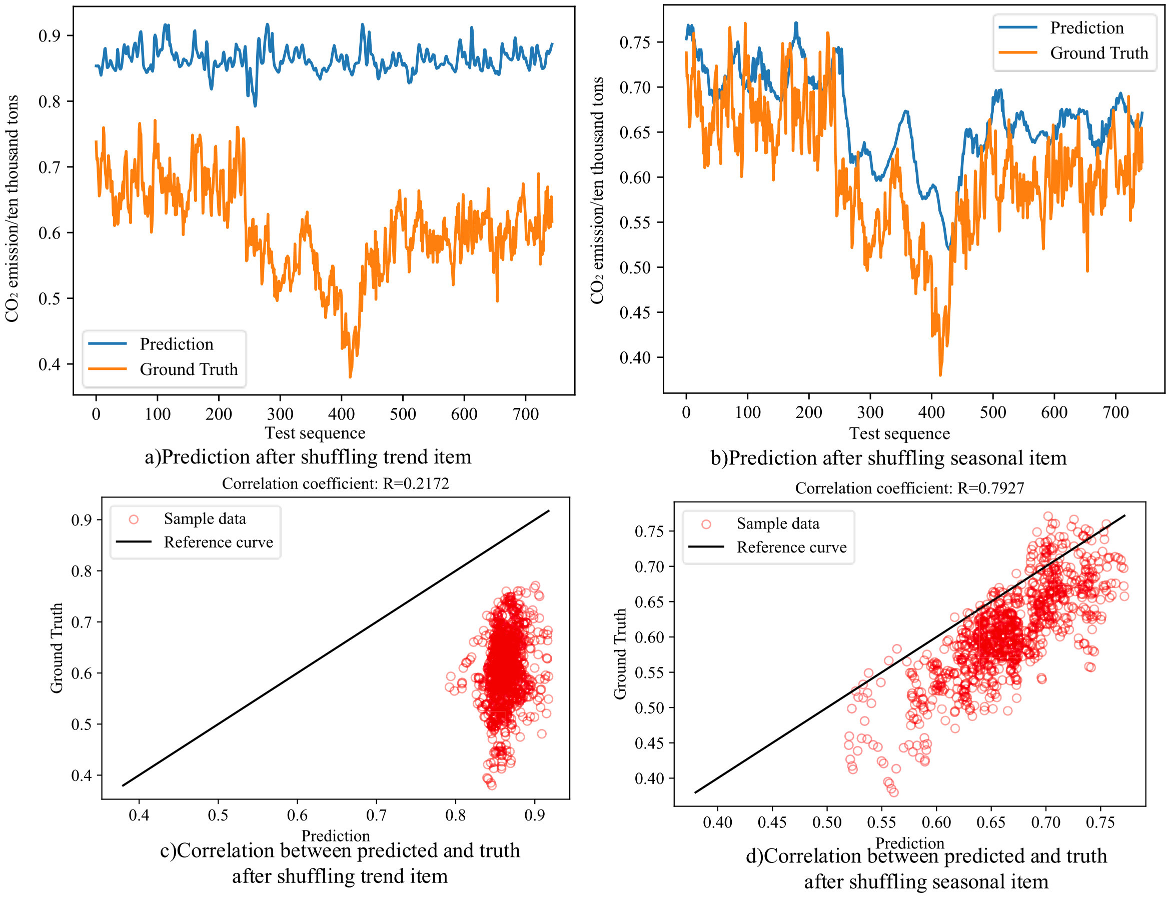


Fig. 15. Predictions after shuffling the trend and seasonal terms, respectively (Model is SCLCC, $d_{LB} = 48$ and $d_{HF} = 1$).

5.4.2. Feature contributions

In this section, this study assesses the contributions of the trend and seasonal terms to the model through separate experiments where each term's data is independently shuffled. The results reveal that the trend term contributes significantly more to the model, approximately 97 %, while the seasonal term has a much smaller contribution, around 3 %. Fig. 15 provides a comprehensive visualization of SCLCC's behavior with a 48-hour look-back window and a 1-hour forecasting horizon when the trend and seasonal terms are shuffled respectively.

As observed in Fig. 15, shuffling the trend term results in the model's inability to capture the trend accurately, leading to predictions that deviate significantly from the actual trend. Conversely, shuffling the seasonal term affects the model's ability to learn the correct seasonal fluctuations. Nonetheless, it can still capture the overall trend of carbon emission time series, resulting in predictions that align with the trend. The findings indicate that the model's predictive performance is less impacted when the seasonal term is disturbed compared to when the trend term is shuffled.

Furthermore, this study assesses the effectiveness of SCLCC in utilizing information from each time node within a 12-hour look-back window and a 1-hour forecasting horizon. For comparison, XGBoost, a machine learning technique known for its strong interpretability, is analysed (Hu and Xiong, 2023; Yan et al., 2023). Fig. 16 presents the contribution of each time point to the model, revealing differences between SCLCC and XGBoost. In the case of XGBoost, the final time node is considered the most important, contributing to over 70 % of the overall importance, as indicated by both the "Gain" criterion and the "Permutation" calculation. On the other hand, SCLCC shows that the contributions of the latter half of the time points are equal to or greater than 10 %, with no single time point exhibiting excessive significance. This suggests that SCLCC is more proficient at assimilating and utilizing retrospective information to make precise predictions than XGBoost. This ability may explain SCLCC's superior predictive performance and resilience to emergency events when compared to XGBoost. In addition, an analysis of the contribution of each time point within the trend and seasonal terms demonstrates a more balanced distribution for each time point within the seasonal term than that of the trend term. Similarly, the contribution of each time point within the trend term is partially similar to the contributions of each time point in the overall look-back window.

Finally, this subsection conducts a visual analysis to depict the contribution of each time point from the trend and seasonal terms when predicting a specific time point within a fixed forecast horizon. This study focuses on SCLCC with a 12-hour look-back window and a 12-hour forecasting horizon. The results are illustrated in Fig. 17. By comparing Fig. 17 a) and b), in contrast to the seasonal term, the trend term exerts a more evenly distributed influence on each predicted time point can be observed. The seasonal term tends to have a more substantial impact on the later predicted time points while showing less influence on the earlier predicted times.

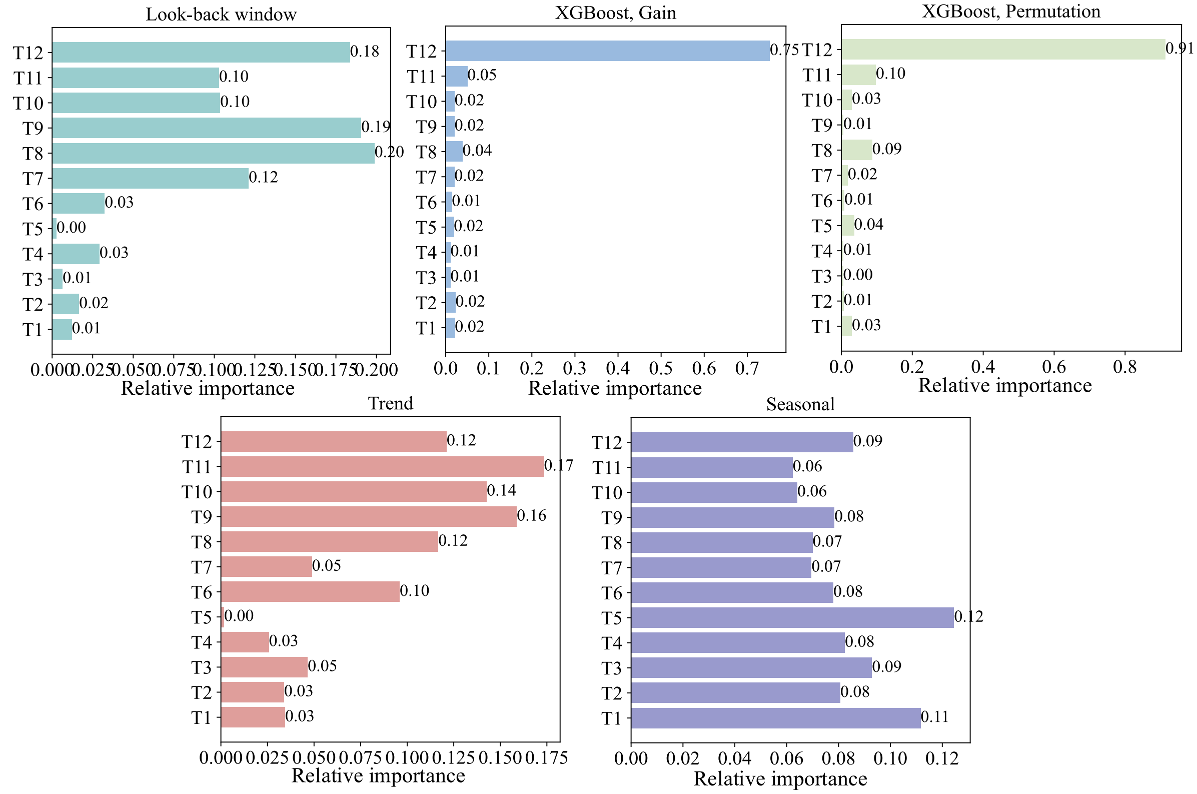


Fig. 16. Time feature contribution comparison (All importance is normalized to 0–1 with the sum equal to 1, and the model is SCLCC, $d_{LB} = 12$ and $d_{HF} = 1$).

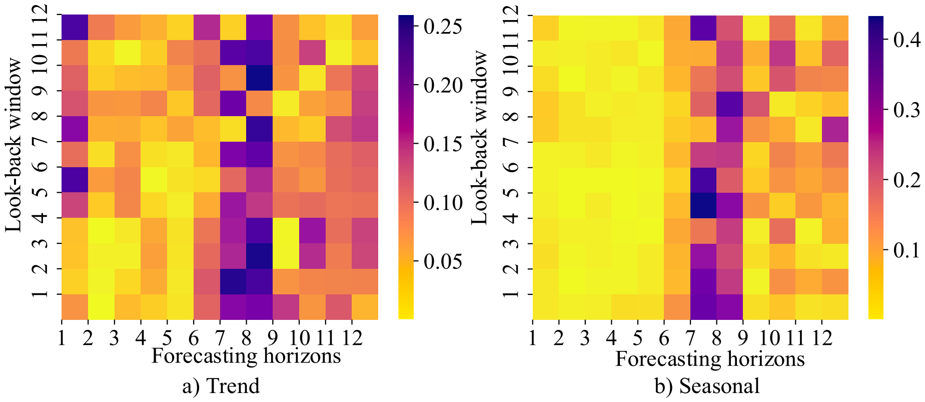


Fig. 17. Visualization of the importance of the time feature (SCLCC is trained with a lock-back window $d_{LB} = 12$ (Y-axis) and forecasting horizons $d_{HF} = 12$ (X-axis)).

Additionally, it is noticeable that prediction time points 7 and 8 are significantly affected by both the trend and seasonal terms.

5.5. Implications

This study holds significant implications for various stakeholders in the shipping industry and researchers in three primary areas: carbon emissions estimation, ship emissions TSF tasks, and the application of deep learning models to address CD:

1) **Carbon Emissions Estimation:** This study addresses the issue of inaccuracies in estimating fuel consumption by auxiliary engines or boilers on ships. It proposes an STSA algorithm, which can help accurately identify a ship's navigation state. This algorithm has several implications:

- The shipping industry can use the ESSNA emission estimation model, which incorporates the STSA algorithm, to precisely calculate emission levels in specific maritime areas or globally. This can better facilitate rational decision-making regarding emissions control.
- Researchers can build on the STSA algorithm and develop more precise navigation state identification algorithms or algorithms involving other functionalities.
- The STSA algorithm can be utilized in emission reduction studies, such as ship speed or trim optimization, to improve the accuracy of objective function modelling and enhance optimization models' effectiveness.

2) **Ship Emission TSF Tasks:** Ship emissions are particularly prominent in areas close to land (Eyring et al., 2010), and their temporal patterns exhibit noticeable heterogeneity (Aliabadi et al., 2015). This study enables TSF in specific sea regions, which can help predict emissions at specific times.

- Such stakeholders in the shipping industry, such as shipowners and administrators, can benefit from this approach by implementing seasonal or temporal emission control policies.
- The framework and techniques employed for carbon emissions forecasting can be extended to predict time series data related to other emissions like PM, SOx, or NOx, offering advantages for various stakeholders. For instance, predicting PM emissions in a harbour region can provide advance notice to harbour personnel and residents, allowing them to take appropriate precautions based on the PM predictions (Park et al., 2023).

3) **Deep Learning Models for Addressing CD:** The shipping industry is susceptible to changes driven by emerging technologies, evolving policies, and significant events such as the COVID-19 pandemic, the Russia-Ukraine crisis, etc. One of the most evident manifestations of these frequent shifts in the shipping industry is the occurrence of three distinct instances of CD in CO_2 emissions from the global merchant fleet sector throughout the decade from 2012 to 2022. These transformative phases are visually represented in Fig. 1, with red dotted lines indicating the approximate time points when these CD events occurred. As such, this study's approach to handling CD bears practical significance:

- The model developed in this study can adapt to changes in the shipping industry without requiring complete retraining, particularly during the early stages of CD when data availability is limited. This allows for continued reliable predictions.
- The model can accurately forecast changes in shipping behaviors during the initial phases of new technologies or policies, providing a quantitative assessment of their effectiveness. This insight is valuable for decision-makers in the industry.

In summary, this study contributes new solutions that can enhance carbon emissions estimation, empower TSF tasks for ship emissions, and equip stakeholders with the tools to address CD in the dynamic shipping industry. These implications are significant for improving the environmental and operational aspects of the industry while fostering adaptability to evolving circumstances.

6. Conclusions

In this study, an enhanced ship emission calculation method, ESSNA, is newly proposed, including a new navigation state identification algorithm, STSA, and a novel interpretable modular deep learning algorithm, SCLCC, for predicting future ship carbon emissions in specific maritime areas. The STSA algorithm within ESSNA aids to accurately identify ships' navigational states within the surveyed zone, enabling precise and effective ship carbon emission calculations. SCLCC efficiently extracts significant information from trend and seasonal terms in time series data, demonstrating robust performance even in the face of emergency events. Its straightforward structure and use of data enhancement techniques ensure reliable interpretability.

While this study presents an advanced approach for predicting ship emissions and a high-performing deep learning model, offering valuable insights for future ship emission reduction measures in the shipping sector, it also encounters some limitations that warrant further investigation. Firstly, due to data constraints, the predictions are univariate and fail to account for factors like port turnover and weather conditions. Secondly, the focus is solely on temporal changes in ship emissions within the study area, without developing a spatiotemporal predictive model to analyse the spatial and temporal variations in emissions. Furthermore, the conclusions drawn from the SCLCC model are based on the analysed data, and its broader applicability needs verification with additional datasets. Thirdly, although both newly proposed methods exhibit superior performance in experiments, they could be improved. For instance, the STSA algorithm requires iterative experiments to fine-tune parameter thresholds, which may vary significantly across different water areas. Moreover, the time-consuming nature of the STSA algorithm in searching for each ship's trajectory poses limitations in swiftly processing large-scale maritime data. Future research could focus on leveraging advanced deep learning models to discern differences in various navigation states, enhancing the rapid processing of extensive data with higher accuracy. Lastly, the current SCLCC model primarily focuses on passively resisting CD while maintaining high accuracy. Therefore, future research could explore the development of an online learning SCLCC model capable of continuously learning from new data, adjusting parameters in real-time, and actively adapting to CD, ultimately improving prediction accuracy.

CRediT authorship contribution statement

Yinwei Feng: Writing – original draft, Validation, Methodology, Conceptualization. **Xinjian Wang:** Writing – original draft, Validation, Funding acquisition, Conceptualization. **Jianlin Luan:** Writing – review & editing, Validation, Conceptualization. **Hua Wang:** Writing – review & editing, Validation, Conceptualization. **Haijiang Li:** Writing – review & editing, Validation, Methodology. **Huanhuan Li:** Formal analysis, Validation, Writing – review & editing. **Zhengjiang Liu:** Validation, Funding acquisition, Conceptualization. **Zaili Yang:** Writing – review & editing, Validation, Funding acquisition, Formal analysis.

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

Data will be made available on request.

Acknowledgements

The authors gratefully acknowledge support from the National Natural Science Foundation of China [Grant No. 52101399], the Fundamental Research Funds for the Central Universities [Grant No. 3132024139] and the Bolian Research Funds of Dalian Maritime University [Grant No. 3132023617]. This work was also supported by the EU H2020 ERC Consolidator Grant program [TRUST Grant No. 864724].

Appendix A. Pseudocodes

Table A1
The pseudocode of STSA.

```
Algorithm 1: Spatiotemporal trajectory search algorithm based on AIS
Input: Ship trajectory, nt; Mooring state space and time thresholds, $\xi_{\mathrm{dis}} = (\xi_a, \xi_b)$ and $\tau_{dis} = (\tau_a, \tau_b)$; Threshold for course change angle, $\eta$; Spatiotemporal coordinate
    search number threshold, $\lambda$ ; Threshold for number of speed anomalies,$\vartheta$
Output: Navigation state of ship sub-trajectories
Begin
Call mooring state identification sub-algorithm, input nt, $\xi_a$, $\xi_b$, $\tau_a$ and $\tau_b$
return at berth sub-trajectories and at anchorage sub- trajectories
```

(continued on next page)

Table A1 (*continued*)

Algorithm 1: Spatiotemporal trajectory search algorithm based on AIS

```
Remaining sub-trajectories are obtained,$rnt$
Call sailing state identification sub-algorithm, input $rnt$, $\eta$ and $\lambda$
return Manoeuvring and cruising sub-trajectories
Call cruising state identification sub-algorithm, input cruising sub-trajectories and $\vartheta$
return Slow-steaming and normal cruising sub-trajectories
End
```

Table A2
The pseudocode of mooring state identification sub-algorithm.

Algorithm 2: Mooring state identification sub-algorithm

Input: Ship trajectory, *nt*; At berth state space and time thresholds, ξ_b and τ_b; At anchorage state space and time thresholds ξ_a and τ_a
Output: At berth state and at anchorage state of ship sub-trajectories

```
Begin
Set the spatiotemporal coordinate of the ship with the earliest time as the starting point for the trajectory,$p_i \leftarrow p_1$
while $i < Numberoftrajectorypoints$:
  for $m = i+1 : Numberoftrajectorypoints$ do
    Calculate the spatial distance and time span between $p_i$ and $p_m$,$dis(p_i,p_m)$ and $\Delta t(p_i,p_m)$
    if $dis(p_i,p_m) < \xi_b$:
      Marking sub-trajectory $snt(p_i,p_m)$ as a possible berthing state
      if $m >= Numberoftrajectorypoints$:
        Determine sub-trajectory $snt(p_i,p_m)$ as a berthing state
    elif $dis(p_i,p_m) > \xi_b$ and $\Delta t(p_i,p_m) > \tau_b$:
      if sub-trajectory $snt(p_i,p_{m-1})$ has been flagged as a possible berthing state:
        Determine sub-trajectory $snt(p_i,p_{m-1})$ as a berthing state
        $i \leftarrow m$, use spatiotemporal coordinate point $p_m$ as the new starting point
        break
    if $dis(p_i,p_m) < \xi_a$:
      Marking sub-trajectory $snt(p_i,p_m)$ as a possible anchorage state
      if $m >= Numberoftrajectorypoints$:
        Determine sub-trajectory $snt(p_i,p_m)$ an anchorage state
    elif $dis(p_i,p_m) > \xi_a$ and $\Delta t(p_i,p_m) > \tau_a$:
      if sub-trajectory $snt(p_i,p_{m-1})$ has been flagged as a possible anchorage state:
        Determine sub-trajectory $snt(p_i,p_{m-1})$ as an anchorage state
        $i \leftarrow m$, use spatiotemporal coordinate point $p_m$ as the new starting point
        break
  if sub-trajectory $snt(p_i,p_m)$ is not at berth or anchor:
    $i + = 1$
End
```

Table A3
The pseudocode of sailing state identification sub-algorithm.

Algorithm 3: Sailing state identification sub-algorithm

Input: Ship trajectory, *nt*; Threshold for course change angle, η; Spatiotemporal coordinate search number threshold,λ
Output: Manoeuvring state and cruising state of ship sub-trajectories

```
Begin
Set the spatiotemporal coordinate of the ship with the earliest time as the starting point for the trajectory,$p_i \leftarrow p_1$
while $i < Numberoftrajectorypoints$:
  Calculate initial course,$C\left(p_i,p_j\right)$
  Mark $p_j$ as the temporary endpoint,$p_e \leftarrow p_j$
  spatiotemporal coordinate search number is 0,$n \leftarrow 0$
  for $k = j+1 : Numberoftrajectorypoints$ do
    if $\Delta C\left(p_i,p_j,p_k\right) \geqslant \Delta C\left(p_i,p_j,p_e\right)$:
      Mark $p_k$ as the temporary endpoint, $p_e \leftarrow p_k$
      $n \leftarrow 0$
    else:
      $n + = 1$
    if $n \geqslant \lambda$:
      break
  if $\Delta C\left(p_i,p_j,p_e\right) \geqslant \eta$:
    Determine sub-trajectory $snt(p_i,p_e)$ as a manoeuvring
  else:
    Determine sub-trajectory $snt(p_i,p_e)$ as a cruising
  $i \leftarrow e + 1$
End
```

Table A4
The pseudocode of cruising state identification sub-algorithm.

Algorithm 4: Cruising state identification sub-algorithm

```
Input: Ship trajectory, $nt$; Threshold for number of speed anomalies,$\vartheta$
Output: Slow-steaming state and normal cruising state of ship sub-trajectories
Begin
Set the spatiotemporal coordinate of the ship with the earliest time as the starting point for the trajectory,$p_i \leftarrow p_1$
while $i < Numberoftrajectorypoints$:
  Calculate sailing speed,$v(p_i, p_{i+1})$
  if $v(p_i, p_{i+1}) > 12$:
    number of consecutive abnormal speed is 0, $n \leftarrow 0$
    for $j = i+2$ : $Numberoftrajectorypoints$ do
      if $\overline{v}\left(snt\left(p_i, p_j\right)\right) > 12$:
        if $v\left(p_{j-1}, p_{j+1}\right) \leqslant 12$:
          $n+=1$
        else:
          $n \leftarrow 0$
        if $n > \vartheta$:
          Determine sub-trajectory $snt\left(p_i, p_{j-\vartheta}\right)$ as a normal cruising
          $i \leftarrow j - \vartheta + 1$
          break
        elif $n < \vartheta$ and $j = Numberoftrajectorypoints$:
          Determine sub-trajectory $snt\left(p_i, p_j\right)$ as a normal cruising
            $i \leftarrow j$
      else:
        Determine sub-trajectory $snt\left(p_i, p_{j-1}\right)$ as a normal cruising
          $i \leftarrow j$
        break
  else:
    $i+=1$
sub-trajectories that are not slow-steaming are all slow-steaming
End
```

Table A5
The pseudocode of STL algorithm.

Algorithm 5: Seasonal-trend decomposition procedure based on loess algorithm.

```
Input: Time-series data, $X_t$.
Output: Trend item $Trend_t$, Seasonal item $Seasonal_t$, and Residual term $Res_t$.
01  Begin
02  Initialize the trend term $Trend_t$ and robust weight $\zeta$ to 0 and 1, respectively.
03  if the inner or outer loop condition is not satisfied:
04    Remove trend item retain $X_t^{-Trend}$, $X_t^{-Trend} \leftarrow X_t - Trend_t^k$.
05    Seasonal element $S_t^{k+1}$ is obtained using Loess smoothing $X_t^{k+1}$.
06    Get trend element $T_t^{k+1}$ by using Loess and Low-pass filter smoothing $S_t^{k+1}$.
07    Seasonal item $Seasonal_t^{k+1}$ is obtained by subtracting the trend element,$Seasonal_t^{k+1} \leftarrow S_t^{k+1} - T_t^{k+1}$.
08    Remove seasonal item for trend pattern $X_t^{-Seasonal}$, $X_t^{-Seasonal} \leftarrow X_t - Seasonal_t^{k+1}$.
09    Trend item $Trend_t^{k+1}$ is obtained using Loess smoothing $X_t^{-Seasonal}$.
10    if the inner loop condition is satisfied:
11      if the outer loop condition is not satisfied:
12        Back to step 04.
13      else:
14        Residual term $Res_t$ equals the original data minus the trend $Trend_t$ and seasonal $Seasonal_t$ terms, $Res_t = X_t - Trend_t - Seasonal_t$.
15        Return Trend item $Trend_t$, Seasonal item $Seasonal_t$, and Residual term $Res_t$.
16    else:
17      Back to step 04.
18  End
```

Appendix B. Supplementary material on the STEAM model

As a micro modelling approach, STEAM estimates CO_2 emissions from ships based on their activities in a bottom-up manner (Gan et al., 2022; Weng et al., 2020). The emissions from ships in the study area within a given time frame are calculated as the sum of CO_2 emissions from all ships passing through the area during that time period. The calculation formulas are presented in Eqs. (B1) to (B4).

$$E_{total}^{T} = \sum \left(E_{i,main}^{T} + E_{i,other}^{T}\right) \tag{B1}$$

$$E_{i,main}^{T} = ef_{i,main} \times P_{i,main} \times \sum \left(LF_{i,t} \times A_{i,LF,t} \times \Delta T_{i,t}\right) \tag{B2}$$

$$LF_{i,t} = P_{i,t} \Big/ P_{i,main} = \left(v_{i,t}/MDS_i\right)^3 \tag{B3}$$

$$E_{i,other}^{T} = ef_{i,other} \times \sum \left(P_{i,other,t,type,state} \times \Delta T_{i,t}\right) \tag{B4}$$

where E_{total}^{T} denotes the total amount of CO$_2$ emissions from the ship in the water areas at time T; $E_{i,main}^{T}$ and $E_{i,other}^{T}$ denote the CO$_2$ emissions from the main engine and auxiliary engine or boiler of ship i at time T, respectively. $ef_{i,main}$ and $ef_{i,other}$ denote the basic emission factors of the ship's main engine and auxiliary engine or boiler of the ship i in units of g/kWh, respectively. $P_{i,main}$ denotes the maximum rated power of the ship's main engine of the ship i in units of kWh. $LF_{i,t}$ denotes the instantaneous load factor of the main engine of ship i at the moment t, which is mainly expressed as the ratio of the instantaneous power of the main engine to the rated power, as shown in Eq. (B3). $A_{i,LF,t}$ is the emission adjustment factor when the $LF_{i,t}$ is lower than 20 %, and the specific value can be referred to the Fourth Greenhouse Gas Study (International Maritime Organization, 2020). $\Delta T_{i,t}$ denotes the time interval between two adjacent trajectory points of the ship i. $P_{i,t}$, $v_{i,t}$ and MDS_i denote the ship's instantaneous power of the main engine, instantaneous speed, and maximum design speed, respectively. $P_{i,other,t,type,state}$ denotes the auxiliary engine or boiler power of ship i belonging to ship type *type* at the moment t when the ship is in the navigation state *state*, the value of these parameters is referenced in literature (International Maritime Organization, 2020).

Information such as ship type, ship main engine rated power, and ship design maximum speed is obtained from the STSD (Li et al., 2023), and the ship's navigation state is identified using the STSA algorithm (detailed in Section 3.1). Furthermore, the CO$_2$ emission factor is derived from specific fuel consumption and the fuel-based emission factor, as depicted in Eq. (B5).

$$ef_e = SFC \times ef_{fuel} \tag{B5}$$

where ef_e represents the emission factor in Eqs. (B2) and (B4); SFC indicates the specific fuel consumption in g/kWh (International Maritime Organization, 2020); and ef_{fuel} represents the fuel-specific emission factor based on the specific fuel in gCO_2/g, the precise values of these parameters are referenced in (International Maritime Organization, 2021).

Appendix C. Supplementary experiments

To thoroughly validate the proposed SCLCC model, this study employs global merchant marine CO$_2$ data sourced from UNCTAD for both model training and testing. The experimental findings are illustrated in Figs. C1 and C2, as well as Table C1. Analysis of the experimental results indicates the superior performance of the SCLCC model compared to alternative models when applied to this dataset.

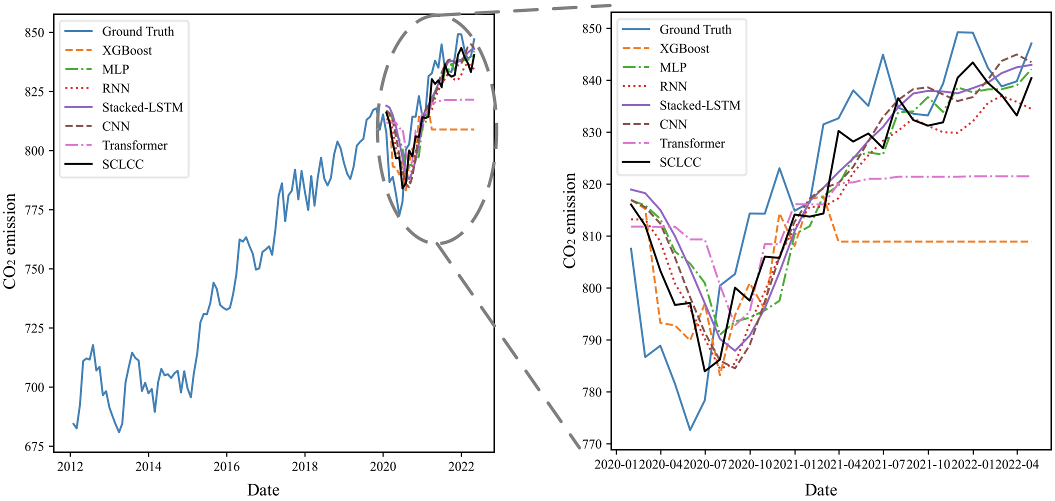


Fig. C1. The results of global CO$_2$ emissions predictions.$d_{LB} = 6$ and $d_{HF} = 1$.

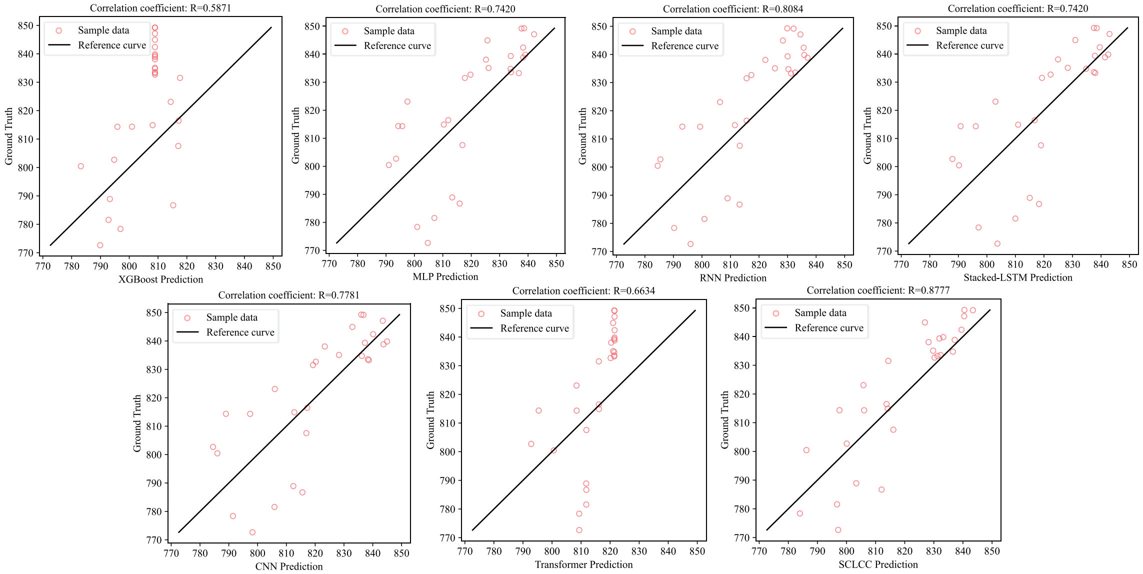


Fig. C2. Correlation between global CO_2 emissions predictions and truth. $d_{LB} = 6$ and $d_{HF} = 1$.

Table C1
Comparing global CO_2 projections using different models ($d_{LB} = 6$ and $d_{HF} = 1$).

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	647741.8513 (493704.67 %)	804.7783 (8783.06 %)	81110.3456 (7269806.39 %)	−8916.6751 (1621604.09 %)
MLP	237.3043 (80.91 %)	12.3454 (36.27 %)	1.5363 (37.70 %)	0.0698 (87.31 %)
RNN	208.0164 (58.58 %)	12.4516 (37.44 %)	1.5282 (36.97 %)	0.1833 (66.67 %)
Stacked-LSTM	234.1574 (75.08 %)	12.1246 (33.83 %)	1.5014 (34.57 %)	0.2026 (63.16 %)
CNN	208.0105 (58.58 %)	11.8972 (31.32 %)	1.4686 (31.63 %)	0.3881 (29.42 %)
Transformer	379.4305 (189.26 %)	17.0542 (88.24 %)	2.0837 (86.76 %)	−5.0524 (1018.79 %)
SCLCC	**131.1737**	**9.0597**	**1.1157**	**0.5499**
Improvement	**58.58 %**	**31.32 %**	**31.63 %**	**29.42 %**

To demonstrate the role of the pooling layer, this study compares the effects of utilizing maximum pooling, average pooling, and no pooling layer in the SCLCC model, as presented in Table C2. Upon reviewing Table C2, it becomes evident that the performance of the model without a pooling layer surpasses that of models employing a pooling layer, given the same training cost.

Table C2
Pool layers comparison.

SCLCC	MSE	MAE	MAPE (%)	Adjusted R^2	Time
Max-Pool	0.00089	0.0232	3.8958	0.7588	213 ± 1.14 s
Average-Pool	0.00083	0.0225	3.7483	0.7800	212 ± 0.329 s
No-Pool	0.00059	0.0187	3.1052	0.8702	244 ± 0.886 s

Note: Models are trained with a lock-back window $d_{LB} = 12$, forecasting horizon $d_{HF} = 1$, and epochs = 100. Time is obtained by training the model for ten runs, with ten loops per run.

To comprehensively evaluate the impact of errors within ship emission estimations derived from the ESSNA model on the conclusions, this study takes into account that the discrepancy between the STEAM model's calculated results and actual values is within 6 % (Wang et al., 2022). Utilizing the formula $Newdata = data/X$ and $X \sim N(1.06, 0.00001)$, new data is generated with a 6 % margin of error from the calculated ship emission data. Subsequently, this new data is employed to retrain and assess the TSF model. The experimental outcomes are shown in Fig. C3 and Table C3. The analysis of the experimental results indicates that the calculation error within ship emissions does not significantly impact the conclusions, reaffirming the superior performance of the SCLCC model compared to alternative models.

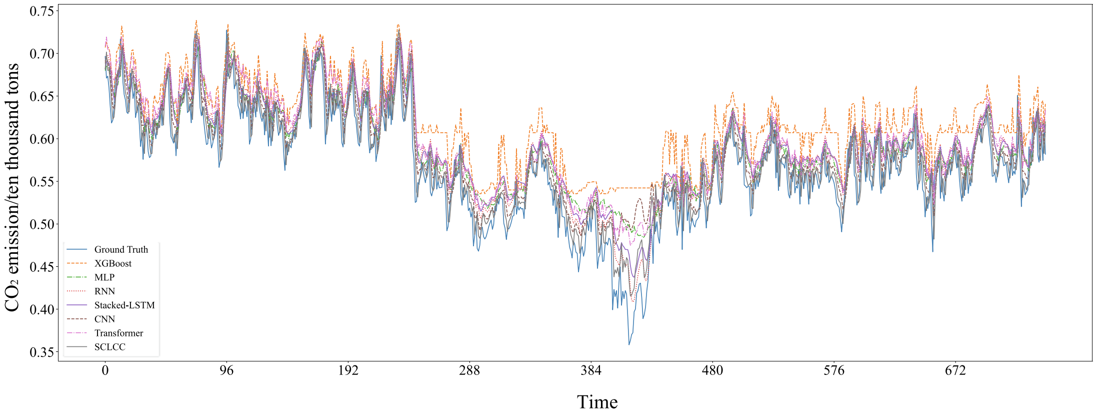


Fig. C3. The visual comparison results of new data predictions.

Table C3
The comparison results of new data prediction using different models.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00318 (421.31 %)	0.0486 (153.13 %)	14.2581 (277.60 %)	0.2670 (67.64 %)
MLP	0.00132 (116.39 %)	0.0285 (48.44 %)	4.9342 (30.67 %)	0.5034 (39.00 %)
RNN	0.00117 (108.20 %)	0.0282 (46.88 %)	4.8061 (27.28 %)	0.6660 (19.29 %)
Stacked-LSTM	0.00124 (103.28 %)	0.0289 (50.52 %)	4.9799 (31.88 %)	0.5775 (30.02 %)
CNN	0.00102 (67.21 %)	0.0237 (23.44 %)	4.1479 (9.84 %)	0.6504 (21.18 %)
Transformer	0.00146 (139.34 %)	0.0312 (62.50 %)	5.3323 (41.22 %)	0.5435 (34.13 %)
SCLCC	**0.00061**	**0.0192**	**3.3776**	**0.8252**
Improvement	**67.21 %**	**23.44 %**	**9.84 %**	**19.29 %**

Appendix D. Performances across models

Table D1
Model performances comparison if $d_{LB} = 4$ and $d_{HF} = 1$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00384 (481.82 %)	0.0412 (107.04 %)	13.9459 (328.34 %)	0.2128 (75.54 %)
MLP	0.00119 (80.30 %)	0.0266 (33.67 %)	4.3611 (33.95 %)	0.6583 (24.33 %)
RNN	0.00137 (107.58 %)	0.0298 (49.75 %)	4.7538 (46.01 %)	0.6895 (20.75 %)
Stacked-LSTM	0.00116 (75.76 %)	0.0277 (39.20 %)	4.5274 (39.06 %)	0.6389 (26.56 %)
CNN	0.00139 (110.61 %)	0.0304 (52.76 %)	4.9692 (52.63 %)	0.5220 (40.00 %)
Transformer	0.00121 (83.33 %)	0.0285 (43.22 %)	4.6130 (41.69 %)	0.6836 (21.43 %)
SCLCC	**0.00066**	**0.0199**	**3.2558**	**0.8700**
Improvement	**75.76 %**	**33.67 %**	**33.95 %**	**20.75 %**

Table D2
Model performances comparison if $d_{LB} = 8$ and $d_{HF} = 1$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00366 (463.08 %)	0.0402 (105.10 %)	13.7087 (320.23 %)	0.2492 (71.07 %)
MLP	0.00091 (40.00 %)	0.0240 (22.45 %)	3.9377 (20.71 %)	0.7632 (11.39 %)
RNN	0.00115 (76.92 %)	0.0267 (36.22 %)	4.2804 (31.21 %)	0.7555 (12.28 %)
Stacked-LSTM	0.00089 (36.92 %)	0.0241 (22.96 %)	3.9743 (21.83 %)	0.7295 (15.30 %)
CNN	0.00132 (103.08 %)	0.0285 (45.41 %)	4.6400 (42.24 %)	0.6339 (26.40 %)
Transformer	0.00075 (15.38 %)	0.0219 (11.73 %)	3.5936 (10.16 %)	0.8096 (6.00 %)
SCLCC	**0.00065**	**0.0196**	**3.2622**	**0.8613**
Improvement	**15.38 %**	**11.73 %**	**10.16 %**	**6.00 %**

Table D3
Model performances comparison if $d_{LB} = 18$ and $d_{HF} = 1$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00462 (470.37 %)	0.0433 (97.72 %)	14.0299 (285.29 %)	0.0531 (93.35 %)
MLP	0.00140 (72.84 %)	0.0305 (39.27 %)	4.9011 (34.59 %)	0.6453 (19.18 %)
RNN	0.00190 (134.57 %)	0.0358 (63.47 %)	5.6234 (54.43 %)	0.5807 (27.27 %)
Stacked-LSTM	0.00177 (118.52 %)	0.0349 (59.36 %)	5.6204 (54.35 %)	0.4036 (49.45 %)
CNN	0.00106 (30.86 %)	0.0258 (17.81 %)	4.2350 (16.30 %)	0.6918 (13.35 %)
Transformer	0.00113 (39.51 %)	0.0274 (25.11 %)	4.5115 (23.89 %)	0.6203 (22.31 %)
SCLCC	**0.00081**	**0.0219**	**3.6414**	**0.7984**
Improvement	**30.86 %**	**17.81 %**	**16.30 %**	**13.35 %**

Table D4
Model performances comparison if $d_{LB} = 24$ and $d_{HF} = 1$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00463 (561.43 %)	0.0434 (105.69 %)	14.0339 (287.92 %)	0.0520 (94.00 %)
MLP	0.00209 (198.57 %)	0.0389 (84.36 %)	6.1725 (70.62 %)	0.4078 (52.92 %)
RNN	0.00165 (135.71 %)	0.0328 (55.45 %)	5.1921 (43.52 %)	0.6343 (26.77 %)
Stacked-LSTM	0.00153 (118.57 %)	0.0324 (53.55 %)	5.2434 (44.94 %)	0.5166 (40.36 %)
CNN	0.00097 (38.57 %)	0.0240 (13.74 %)	3.9797 (10.01 %)	0.7207 (16.80 %)
Transformer	0.00197 (181.43 %)	0.0379 (79.62 %)	6.0772 (67.99 %)	0.4571 (47.23 %)
SCLCC	**0.00070**	**0.0211**	**3.6177**	**0.8662**
Improvement	**38.57 %**	**13.74 %**	**10.01 %**	**16.80 %**

Table D5
Model performances comparison if $d_{LB} = 36$ and $d_{HF} = 1$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00848 (1246.03 %)	0.0622 (222.28 %)	14.9709 (367.72 %)	−0.7372 (188.18 %)
MLP	0.00138 (119.05 %)	0.0306 (58.55 %)	4.9445 (54.48 %)	0.6063 (27.48 %)
RNN	0.00136 (115.87 %)	0.0292 (51.30 %)	4.6854 (46.38 %)	0.7534 (9.88 %)
Stacked-LSTM	0.00145 (130.16 %)	0.0315 (63.21 %)	5.0986 (59.29 %)	0.5531 (33.84 %)
CNN	0.00087 (38.10 %)	0.0226 (17.10 %)	3.7888 (18.37 %)	0.7258 (13.18 %)
Transformer	0.00135 (114.29 %)	0.0305 (58.03 %)	4.9719 (55.33 %)	0.5930 (29.07 %)
SCLCC	**0.00063**	**0.0193**	**3.2008**	**0.8360**
Improvement	**38.10 %**	**17.10 %**	**18.37 %**	**9.88 %**

Table D6
Model performances comparison if $d_{LB} = 48$ and $d_{HF} = 1$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00806 (1035.21 %)	0.0622 (199.04 %)	14.8654 (337.37 %)	−0.6520 (177.60 %)
MLP	0.00126 (77.46 %)	0.0291 (39.90 %)	4.7123 (38.65 %)	0.6387 (23.98 %)
RNN	0.00128 (80.28 %)	0.0285 (37.02 %)	4.5544 (34.00 %)	0.7214 (14.14 %)
Stacked-LSTM	0.00145 (104.23 %)	0.0314 (50.96 %)	5.0823 (49.53 %)	0.5528 (34.21 %)
CNN	0.00087 (22.54 %)	0.0226 (8.65 %)	3.7919 (11.57 %)	0.7287 (13.27 %)
Transformer	0.00182 (156.34 %)	0.0359 (72.60 %)	5.7311 (68.62 %)	0.7469 (11.10 %)
SCLCC	**0.00071**	**0.0208**	**3.3988**	**0.8402**
Improvement	**22.54 %**	**8.65 %**	**11.57 %**	**11.10 %**

Table D7
Model performances comparison if $d_{LB} = 12$ and $d_{HF} = 2$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00211 (170.51 %)	0.0293 (36.92 %)	4.7700 (34.15 %)	0.3544 (56.05 %)
MLP	0.00189 (142.31 %)	0.0350 (63.55 %)	5.6187 (58.02 %)	0.4595 (43.01 %)
RNN	0.00158 (102.56 %)	0.0320 (49.53 %)	5.0584 (42.26 %)	0.6555 (18.70 %)
Stacked-LSTM	0.00163 (108.97 %)	0.0329 (53.74 %)	5.3199 (49.61 %)	0.4578 (43.22 %)
CNN	0.00107 (37.18 %)	0.0250 (16.82 %)	4.1928 (17.91 %)	0.6557 (18.68 %)
Transformer	0.00123 (57.69 %)	0.0279 (30.37 %)	4.5222 (27.18 %)	0.6235 (22.67 %)
SCLCC	**0.00078**	**0.0214**	**3.5558**	**0.8063**
Improvement	**37.18 %**	**16.82 %**	**17.91 %**	**18.68 %**

Table D8
Model performances comparison if $d_{LB} = 12$ and $d_{HF} = 4$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00275 (154.63 %)	0.0358 (40.39 %)	5.7736 (36.61 %)	0.2100 (70.89 %)
MLP	0.00192 (77.78 %)	0.0350 (37.25 %)	5.5856 (32.16 %)	0.4143 (42.58 %)
RNN	0.00248 (129.63 %)	0.0403 (58.04 %)	6.4553 (52.74 %)	0.5352 (25.82 %)
Stacked-LSTM	0.00182 (68.52 %)	0.0344 (34.90 %)	5.5203 (30.62 %)	0.3984 (44.78 %)
CNN	0.00149 (37.96 %)	0.0295 (15.69 %)	4.8876 (15.65 %)	0.5050 (30.01 %)
Transformer	0.00157 (45.37 %)	0.0312 (22.35 %)	5.0486 (19.46 %)	0.4512 (37.46 %)
SCLCC	**0.00108**	**0.0255**	**4.2263**	**0.7215**
Improvement	**37.96 %**	**15.69 %**	**15.65 %**	**25.82 %**

Table D9
Model performances comparison if $d_{LB} = 12$ and $d_{HF} = 6$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00339 (156.82 %)	0.0408 (43.66 %)	6.5298 (39.35 %)	0.1495 (75.27 %)
MLP	0.00203 (53.79 %)	0.0357 (25.70 %)	5.6925 (21.48 %)	0.3389 (43.95 %)
RNN	0.00192 (45.45 %)	0.0347 (22.18 %)	5.5134 (17.66 %)	0.4748 (21.47 %)
Stacked-LSTM	0.00210 (59.09 %)	0.0370 (30.28 %)	5.8872 (25.63 %)	0.4128 (31.72 %)
CNN	0.00170 (28.79 %)	0.0319 (12.32 %)	5.1736 (10.41 %)	0.4592 (24.05 %)
Transformer	0.00194 (46.97 %)	0.0346 (21.83 %)	5.5520 (18.48 %)	0.3308 (45.29 %)
SCLCC	**0.00132**	**0.0284**	**4.6860**	**0.6046**
Improvement	**28.79 %**	**12.32 %**	**10.41 %**	**24.05 %**

Table D10
Model performances comparison if $d_{LB} = 12$ and $d_{HF} = 8$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00414 (167.10 %)	0.0470 (52.60 %)	7.4749 (47.38 %)	0.0859 (83.29 %)
MLP	0.00231 (49.03 %)	0.0382 (24.02 %)	6.0549 (19.38 %)	0.2569 (50.03 %)
RNN	0.00221 (42.58 %)	0.0372 (20.78 %)	5.9090 (16.51 %)	0.3057 (40.54 %)
Stacked-LSTM	0.00239 (54.19 %)	0.0395 (28.25 %)	6.2461 (23.15 %)	0.3220 (37.37 %)
CNN	0.00168 (8.39 %)	0.0320 (3.90 %)	5.2291 (3.10 %)	0.4073 (20.77 %)
Transformer	0.00247 (59.35 %)	0.0389 (26.30 %)	6.1700 (21.65 %)	0.1025 (80.06 %)
SCLCC	**0.00155**	**0.0308**	**5.0718**	**0.5141**
Improvement	**8.39 %**	**3.90 %**	**3.10 %**	**20.77 %**

Table D11
Model performances comparison if $d_{LB} = 12$ and $d_{HF} = 10$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00641 (246.49 %)	0.0515 (53.27 %)	7.8649 (43.39 %)	−0.0468 (115.67 %)
MLP	0.00257 (38.92 %)	0.0396 (17.86 %)	6.2400 (13.77 %)	0.0930 (68.85 %)
RNN	0.00511 (176.22 %)	0.0625 (86.01 %)	9.4401 (72.11 %)	−0.5569 (286.50 %)
Stacked-LSTM	0.00360 (94.59 %)	0.0493 (46.73 %)	7.5622 (37.88 %)	0.1103 (63.06 %)
CNN	0.00242 (30.81 %)	0.0380 (13.10 %)	6.1015 (11.24 %)	0.0085 (97.15 %)
Transformer	0.00361 (95.14 %)	0.0476 (41.67 %)	7.3866 (34.67 %)	−0.9232 (409.18 %)
SCLCC	**0.00185**	**0.0336**	**5.4848**	**0.2986**
Improvement	**30.81 %**	**13.10 %**	**11.24 %**	**63.06 %**

Table D12
Model performances comparison if $d_{LB} = 12$ and $d_{HF} = 12$.

Model	MSE	MAE	MAPE (%)	Adjusted R^2
XGBoost	0.00644 (218.81 %)	0.0551 (56.09 %)	8.5125 (49.73 %)	0.0306 (90.30 %)
MLP	0.00321 (58.91 %)	0.0462 (30.88 %)	7.2262 (27.10 %)	−0.2827 (189.60 %)
RNN	0.00825 (308.42 %)	0.0830 (135.13 %)	12.1889 (114.39 %)	−1.8994 (702.03 %)
Stacked-LSTM	0.00416 (105.94 %)	0.0545 (54.39 %)	8.2686 (45.44 %)	−0.0061 (101.93 %)
CNN	0.00362 (79.21 %)	0.0491 (39.09 %)	7.6791 (35.07 %)	−0.6500 (306.02 %)
Transformer	0.00349 (72.77 %)	0.0469 (32.86 %)	7.2769 (27.99 %)	−0.4643 (247.16 %)
SCLCC	**0.00202**	**0.0353**	**5.6854**	**0.3155**
Improvement	**58.91 %**	**30.88 %**	**27.10 %**	**90.30 %**

Appendix E. Models framework

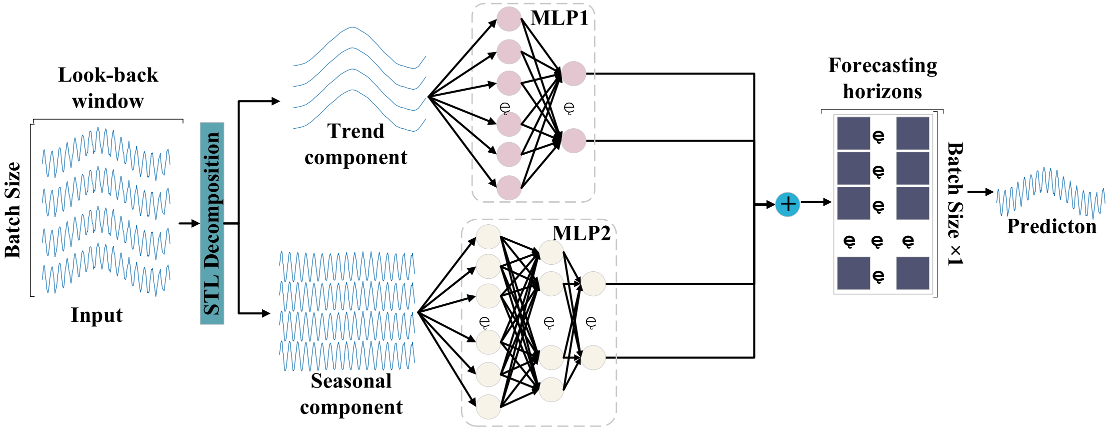


Fig. E1. Model infrastructure with three components removed.

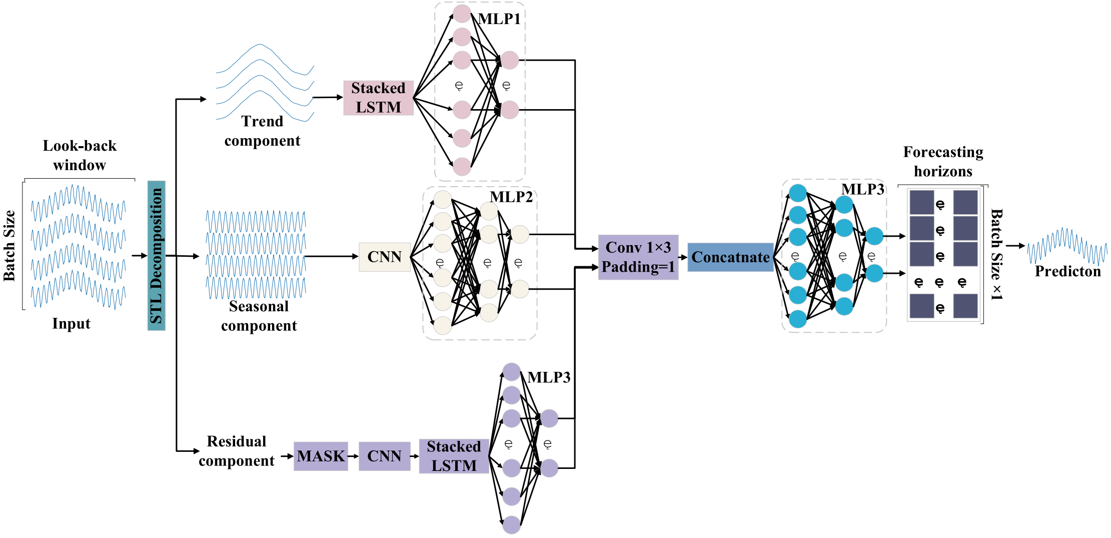


Fig. E2. Model infrastructure with Residual component.

References

Abdelaty, H., Al-Obaidi, A., Mohamed, M., Farag, H.E.Z., 2021. Machine learning prediction models for battery-electric bus energy consumption in transit. Transp. Res. Part d: Transp. Environ. 96, 102868 https://doi.org/10.1016/j.trd.2021.102868.

Acharya, S., Singleton, P.A., 2022. Associations of inclement weather and poor air quality with non-motorized trail volumes. Transp. Res. Part d: Transp. Environ. 109, 103337 https://doi.org/10.1016/j.trd.2022.103337.

Alfaseeh, L., Tu, R., Farooq, B., Hatzopoulou, M., 2020. Greenhouse gas emission prediction on road network using deep sequence learning. Transp. Res. Part d: Transp. Environ. 88, 102593 https://doi.org/10.1016/j.trd.2020.102593.

Aliabadi, A.A., Staebler, R.M., Sharma, S., 2015. Air quality monitoring in communities of the Canadian Arctic during the high shipping season with a focus on local and marine pollution. Atmos. Chem. Phys. 15 (5), 2651–2673. https://doi.org/10.5194/acp-15-2651-2015.

Andreou, A., Mavromoustakis, C.X., Batalla, J.M., Markakis, E.K., Mastorakis, G., Mumtaz, S., 2023. UAV trajectory optimisation in smart cities using modified A* algorithm combined with Haversine and Vincenty formulas. IEEE Trans. Veh. Technol. 72 (8), 9757–9769. https://doi.org/10.1109/TVT.2023.3254604.

Bogaerts, T., Masegosa, A.D., Angarita-Zapata, J.S., Onieva, E., Hellinckx, P., 2020. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. Transp. Res. Part c: Emerg. Technol. 112, 62–77. https://doi.org/10.1016/j.trc.2020.01.010.

Chen, S., Meng, Q., Jia, P., Kuang, H., 2021. An operational-mode-based method for estimating ship emissions in port waters. Transp. Res. Part d: Transp. Environ. 101, 103080 https://doi.org/10.1016/j.trd.2021.103080.

Damman, S., Steen, M., 2021. A socio-technical perspective on the scope for ports to enable energy transition. Transp. Res. Part d: Transp. Environ. 91, 102691 https://doi.org/10.1016/j.trd.2020.102691.

Eyring, V., Isaksen, I.S.A., Berntsen, T., Collins, W.J., Corbett, J.J., Endresen, O., Grainger, R.G., Moldanova, J., Schlager, H., Stevenson, D.S., 2010. Transport impacts on atmosphere and climate: Shipping. Atmos. Environ. 44 (37), 4735–4771. https://doi.org/10.1016/j.atmosenv.2009.04.059.

Feng, Y., Wang, X., Chen, Q., Yang, Z., Wang, J., Li, H., Xia, G., Liu, Z., 2024. Prediction of the severity of marine accidents using improved machine learning. Transp. Res. Part E: Logist. Transp. Rev. 188, 103647. https://doi.org/10.1016/j.tre.2024.103647.

Fuentes García, G., Baldasano Recio, J.M., Sosa Echeverría, R., Granados Hernández, E., Zamora Vargas, E., Antonio Duran, R., Kahl, J.W., 2021. Estimation of atmospheric emissions from maritime activity in the Veracruz port, Mexico. J. Air Waste Manag. Assoc. 71 (8), 934–948. https://doi.org/10.1080/10962247.2021.1902421.

Gan, L., Che, W., Zhou, M., Zhou, C., Zheng, Y., Zhang, L., Rangel-Buitrago, N., Song, L., 2022. Ship exhaust emission estimation and analysis using Automatic Identification System data: The west area of Shenzhen port, China, as a case study. Ocean Coast. Manag. 226, 106245 https://doi.org/10.1016/j.ocecoaman.2022.106245.

Gharib, C., Mefteh-Wali, S., Serret, V., Ben Jabeur, S., 2021. Impact of COVID-19 pandemic on crude oil prices: Evidence from Econophysics approach. Resour. Policy 74. https://doi.org/10.1016/j.resourpol.2021.102392.

Górecki, T., Łuczak, M., 2015. Multivariate time series classification with parametric derivative dynamic time warping. Expert Syst. Appl. 42 (5), 2305–2312. https://doi.org/10.1016/j.eswa.2014.11.007.

Han, I., Samarneh, L., Stock, T.H., Symanski, E., 2018. Impact of transient truck and train traffic on ambient air and noise levels in underserved communities. Transp. Res. Part d: Transp. Environ. 63, 706–717. https://doi.org/10.1016/j.trd.2018.07.010.

Hellton, K.H., Tveten, M., Stakkeland, M., Engebretsen, S., Haug, O., Aldrin, M., 2022. Real-time prediction of propulsion motor overheating using machine learning. J. Mar. Eng. Technol. 21 (6), 334–342. https://doi.org/10.1080/20464177.2021.1978745.

Herrema, F., Curran, R., Hartjes, S., Ellejmi, M., Bancroft, S., Schultz, M., 2019. A machine learning model to predict runway exit at Vienna airport. Transp. Res. Part e: Log. Transp. Rev. 131, 329–342. https://doi.org/10.1016/j.tre.2019.10.002.

Hickman, R., Ashiru, O., Banister, D., 2011. Transitions to low carbon transport futures: Strategic conversations from London and Delhi. J. Transp. Geogr. 19 (6), 1553–1562. https://doi.org/10.1016/j.jtrangeo.2011.03.013.

Hou, J., Wang, Y., Zhou, J., Tian, Q., 2022. Prediction of hourly air temperature based on CNN–LSTM. Geomat. Nat. Haz. Risk 13, 1962–1986. https://doi.org/10.1080/19475705.2022.2102942.

Hsiao, C.-W., Chan, Y.-C., Lee, M.-Y., Lu, H.-P., 2021. Heteroscedasticity and precise estimation model approach for complex financial time-series data: an example of Taiwan Stock Index Futures before and during COVID-19. Mathematics 9 (21), 2719. https://doi.org/10.3390/math9212719.

Hu, S., Chen, P., Chen, X., 2021. Do personalized economic incentives work in promoting shared mobility? Examining customer churn using a time-varying Cox model. Transp. Res. Part c: Emerg. Technol. 128, 103224 https://doi.org/10.1016/j.trc.2021.103224.

Hu, S., Xiong, C., 2023. High-dimensional population inflow time series forecasting via an interpretable hierarchical transformer. Transp. Res. Part c: Emerg. Technol. 146, 103962 https://doi.org/10.1016/j.trc.2022.103962.

Huang, L., Wen, Y., Zhang, Y., Zhou, C., Zhang, F., Yang, T., 2020. Dynamic calculation of ship exhaust emissions based on real-time AIS data. Transp. Res. Part d: Transp. Environ. 80, 102277 https://doi.org/10.1016/j.trd.2020.102277.

Hussain, I., Wang, H., Safdar, M., Ho, Q.B., Wemegah, T.D., Noor, S., 2022. Estimation of shipping emissions in developing country: A case study of Mohammad Bin Qasim Port, Pakistan. Int. J. Environ. Res. Public 19 (19), 11868. https://doi.org/10.3390/ijerph191911868.

International Maritime Organization, 2020. Fourth Greenhouse Gas Study 2020. International Maritime Organization, London, UK.

International Maritime Organization, 2021. Guidelines on the method of calculation of the attained energy efficiency design index (EEDI) for new ships. International Maritime Organization, London, UK.

Istaiteh, O., Owais, T., Al-Madi, N., Abu-Soud, S., 2020. Machine Learning Approaches for COVID-19 Forecasting, *2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pp. 50-57.

Jalkanen, J.P., Brink, A., Kalli, J., Pettersson, H., Kukkonen, J., Stipa, T., 2009. A modelling system for the exhaust emissions of marine traffic and its application in the Baltic Sea area. Atmos. Chem. Phys. 9 (23), 9209–9223. https://doi.org/10.5194/acp-9-9209-2009.

Kabir, R., Remias, S.M., Waddell, J., Zhu, D., 2023. Time-Series fuel consumption prediction assessing delay impacts on energy using vehicular trajectory. Transp. Res. Part d: Transp. Environ. 117, 103678 https://doi.org/10.1016/j.trd.2023.103678.

Karimian, M., Beigy, H., 2023. Concept drift handling: A domain adaptation perspective. Expert Syst. Appl. 224, 119946 https://doi.org/10.1016/j.eswa.2023.119946.

Khosravi, M., Ghazani, M.M., 2023. Novel insights into the modeling financial time-series through machine learning methods: Evidence from the cryptocurrency market. Expert Syst. Appl. 234, 121012 https://doi.org/10.1016/j.eswa.2023.121012.

Li, H., Çelik, C., Bashir, M., Zou, L., Yang, Z., 2024a. Incorporation of a global perspective into data-driven analysis of maritime collision accident risk. Reliab. Eng. Syst. Saf. 249, 110187 https://doi.org/10.1016/j.ress.2024.110187.

Li, J., Guo, F., Sivakumar, A., Dong, Y., Krishnan, R., 2021. Transferability improvement in short-term traffic prediction using stacked LSTM network. Transp. Res. Part c: Emerg. Technol. 124, 102977 https://doi.org/10.1016/j.trc.2021.102977.

Li, H., Jia, P., Wang, X., Yang, Z., Wang, J., Kuang, H., 2023. Ship carbon dioxide emission estimation in coastal domestic emission control areas using high spatial-temporal resolution data: A China case. Ocean Coast. Manag. 232, 106419 https://doi.org/10.1016/j.ocecoaman.2022.106419.

Li, Y., Li, H., Zhang, C., Zhao, Y., Yang, Z., 2024b. Incorporation of adaptive compression into a GPU parallel computing framework for analyzing large-scale vessel trajectories. Transp. Res. Part c: Emerg. Technol. 163, 104648 https://doi.org/10.1016/j.trc.2024.104648.

Liu, Y., Tong, L.C., Zhu, X., Du, W., 2021. Dynamic activity chain pattern estimation under mobility demand changes during COVID-19. Transp. Res. Part c: Emerg. Technol. 131, 103361 https://doi.org/10.1016/j.trc.2021.103361.

Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., Xu, Q., 2022. Scinet: Time series modeling and forecasting with sample convolution and interaction. Adv. Neural Inf. Proces. Syst. 35, 5816–5828.

Luan, J., Feng, Y., Li, H., Wang, X., Jia, P., Kuang, H., 2023. Research on the spatiotemporal prediction of ship carbon emissions based on the deep learning model. Sci. Res. Manage. 44 (3), 75–85.

Luo, X., Yan, R., Wang, S., 2023. Comparison of deterministic and ensemble weather forecasts on ship sailing speed optimization. Transp. Res. Part d: Transp. Environ. 121, 103801 https://doi.org/10.1016/j.trd.2023.103801.

Ma, T., Antoniou, C., Toledo, T., 2020. Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. Transp. Res. Part c: Emerg. Technol. 111, 352–372. https://doi.org/10.1016/j.trc.2019.12.022.

Moreno-Gutiérrez, J., Calderay, F., Saborido, N., Boile, M., Rodríguez Valero, R., Durán-Grados, V., 2015. Methodologies for estimating shipping emissions and energy consumption: A comparative analysis of current methods. Energy 86, 603–616. https://doi.org/10.1016/j.energy.2015.04.083.

Mroua, M., Lamine, A., 2023. Financial time series prediction under Covid-19 pandemic crisis with Long Short-Term Memory (LSTM) network. Hum. Soc. Sci. Commun. 10 (1), 530. https://doi.org/10.1057/s41599-023-02042-w.

Muhammad, S., Long, X., 2020. China's seaborne oil import and shipping emissions: The prospect of belt and road initiative. Mar. Pollut. Bull. 158, 111422 https://doi.org/10.1016/j.marpolbul.2020.111422.

Murcia González, J.-C., 2021. Analysis and measurement of SOx, CO2, PM and NOx emissions in port auxiliary vessels. Environ. Monit. Assess. 193 (6), 374. https://doi.org/10.1007/s10661-021-09145-4.

Nocera, S., Basso, M., Cavallaro, F., 2017. Micro and Macro modelling approaches for the evaluation of the carbon impacts of transportation. Transp. Res. Procedia 24, 146–154. https://doi.org/10.1016/j.trpro.2017.05.080.

Park, S.-Y., Woo, S.-H., Lim, C., 2023. Predicting PM10 and PM2.5 concentration in container ports: A deep learning approach. Transp. Res. Part d: Transp. Environ. 115, 103601 https://doi.org/10.1016/j.trd.2022.103601.

Peng, X., Wen, Y., Wu, L., Xiao, C., Zhou, C., Han, D., 2020. A sampling method for calculating regional ship emission inventories. Transp. Res. Part d: Transp. Environ. 89, 102617 https://doi.org/10.1016/j.trd.2020.102617.

Qian, Z., Pei, Y., Zareipour, H., Chen, N., 2019. A review and discussion of decomposition-based hybrid models for wind energy forecasting applications. Appl. Energy 235, 939–953. https://doi.org/10.1016/j.apenergy.2018.10.080.

Servizi, V., Petersen, N.C., Pereira, F.C., Nielsen, O.A., 2020. Stop detection for smartphone-based travel surveys using geo-spatial context and artificial neural networks. Transp. Res. Part c: Emerg. Technol. 121, 102834 https://doi.org/10.1016/j.trc.2020.102834.

Sim, S., Park, J.-H., Bae, H., 2022. Deep collaborative learning model for port-air pollutants prediction using automatic identification system. Transp. Res. Part d: Transp. Environ. 111, 103431 https://doi.org/10.1016/j.trd.2022.103431.

Tang, H., Yin, Y., Shen, H., 2022. A model for vessel trajectory prediction based on long short-term memory neural network. J. Mar. Eng. Technol. 21 (3), 136–145. https://doi.org/10.1080/20464177.2019.1665258.

Tebong, N.K., Simo, T., Takougang, A.N., Sandjon, A.T., Herve, N.P., 2023. Application of deep learning algorithms to confluent flow-rate forecast with multivariate decomposed variables. J. Hydrol.: Reg. Stud. 46, 101357 https://doi.org/10.1016/j.ejrh.2023.101357.

Trull, O., García-Díaz, J.C., Peiró-Signes, A., 2022. Multiple seasonal STL decomposition with discrete-interval moving seasonalities. Appl. Math Comput. 433, 127398 https://doi.org/10.1016/j.amc.2022.127398.

Wan, Z., Ji, S., Liu, Y., Zhang, Q., Chen, J., Wang, Q., 2020. Shipping emission inventories in China's Bohai Bay, Yangtze River Delta, and Pearl River Delta in 2018. Mar. Pollut. Bull. 151, 110882 https://doi.org/10.1016/j.marpolbul.2019.110882.

Wan, Z., Cai, Z., Zhao, R., Zhang, Q., Chen, J., Wang, Z., 2023. Quantifying the air quality impact of ship emissions in China's Bohai Bay. Mar. Pollut. Bull. 193, 115169 https://doi.org/10.1016/j.marpolbul.2023.115169.

Wang, K., Wang, J., Huang, L., Yuan, Y., Wu, G., Xing, H., Wang, Z., Wang, Z., Jiang, X., 2022. A comprehensive review on the prediction of ship energy consumption and pollution gas emissions. Ocean Eng. 266, 112826 https://doi.org/10.1016/j.oceaneng.2022.112826.

Wang, Q., Wang, J., Qu, Y., Yu, T., 2023. Assessing the impact of COVID-19 on air pollutant emissions from vessels in Lianyungang Port. Mar. Pollut. Bull. 194, 115313 https://doi.org/10.1016/j.marpolbul.2023.115313.

Weng, J., Shi, K., Gan, X., Li, G., Huang, Z., 2020. Ship emission estimation with high spatial-temporal resolution in the Yangtze River estuary using AIS data. J. Clean. Prod. 248, 119297 https://doi.org/10.1016/j.jclepro.2019.119297.

Wu, J., Jia, P., Feng, T., Li, H., Kuang, H., 2023. Spatiotemporal analysis of built environment restrained traffic carbon emissions and policy implications. Transp. Res. Part d: Transp. Environ. 121, 103839 https://doi.org/10.1016/j.trd.2023.103839.

Wu, S., Peng, D., 2022. Pre-SMATS: A multi-task learning based prediction model for small multi-stage seasonal time series. Expert Syst. Appl. 201, 117121 https://doi.org/10.1016/j.eswa.2022.117121.

Xin, X., Liu, K., Li, H., Yang, Z., 2024. Maritime traffic partitioning: An adaptive semi-supervised spectral regularization approach for leveraging multi-graph evolutionary traffic interactions. Transp. Res. Part c: Emerg. Technol. 164, 104670 https://doi.org/10.1016/j.trc.2024.104670.

Yan, R., Wang, S., Zhen, L., 2023. An extended smart "predict, and optimize" (SPO) framework based on similar sets for ship inspection planning. Transp. Res. Part e: Log. Transp. Rev. 173, 103109 https://doi.org/10.1016/j.tre.2023.103109.

Yuan, W., Frey, H.C., 2021. Multi-scale evaluation of diesel commuter rail fuel use, emissions, and eco-driving. Transp. Res. Part d: Transp. Environ. 99, 102995 https://doi.org/10.1016/j.trd.2021.102995.

Zeng, A., Chen, M., Zhang, L., Xu, Q., 2023. Are transformers effective for time series forecasting?, *Proceedings of the AAAI conference on artificial intelligence*, pp. 11121-11128.

Zhang, X., Wang, J., 2018. A novel decomposition-ensemble model for forecasting short-term load-time series with multiple seasonal patterns. Appl. Soft Comput. 65, 478–494. https://doi.org/10.1016/j.asoc.2018.01.017.

Zhu, X., Li, L., 2021. Flight time prediction for fuel loading decisions with a deep learning approach. Transp. Res. Part c: Emerg. Technol. 128, 103179 https://doi.org/10.1016/j.trc.2021.103179.