applied sciences

MDPI

Article

Hybrid Neural Network-Based Maritime Carbon Dioxide Emission Prediction: Incorporating Dynamics for Enhanced Accuracy

Seunghun Lim [1] and Jungmo Oh [2,*]

[1] Department of Marine Engineering, Mokpo National Maritime University, Mokpo 58628, Republic of Korea; seunghun3902@naver.com
[2] Division of Marine System Engineering, Mokpo National Maritime University, Mokpo 58628, Republic of Korea
* Correspondence: jmoh@mmu.ac.kr

Abstract: The rapid expansion of international maritime transportation has led to rising greenhouse gas emissions, exacerbating climate change and environmental sustainability concerns. According to the International Maritime Organization, carbon dioxide (CO_2) emissions from vessels are projected to increase by over 17% by 2050. Traditional emission estimation methods are prone to inaccuracies due to uncertainties in emission factors, and inconsistencies in fuel consumption data. This study proposes deep learning-based CO_2 emission prediction models leveraging engine operation data. Unlike previous approaches that primarily relied on fuel consumption, this model incorporates multiple parameters capturing the relationship between combustion characteristics and emissions to enhance predictive accuracy. We developed and evaluated individual models—convolutional neural network (CNN), long short-term memory (LSTM), and temporal convolutional network (TCN)—as well as hybrid model (TCN–LSTM). The hybrid model achieved the highest predictive performance, with a coefficient of determination of 0.9726, outperforming other models across multiple quantitative metrics. These findings demonstrate the potential of deep learning for vessel emission assessment, providing a scientific basis for carbon management strategies and policy development in the international shipping industry. This study thus holds major academic and industrial value, advancing the field of deep learning-based emission prediction and extending its applicability to diverse operational scenarios.

Keywords: vessel emission prediction; carbon dioxide; hybrid neural network; diesel engine; emission inventory

check for updates

Academic Editor: Juan García Rodríguez

Received: 18 March 2025
Revised: 17 April 2025
Accepted: 21 April 2025
Published: 23 April 2025

Citation: Lim, S.; Oh, J. Hybrid Neural Network-Based Maritime Carbon Dioxide Emission Prediction: Incorporating Dynamics for Enhanced Accuracy. *Appl. Sci.* **2025**, *15*, 4654. https://doi.org/10.3390/app15094654

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Energy is fundamental to industrial and economic activities, with fossil fuels remaining the primary energy source. However, their excessive use has significantly increased greenhouse gas (GHG) emissions, contributing to severe environmental pollution [1]. According to the International Energy Agency, environmental pollution causes approximately 6.5 million deaths annually, prompting many countries, including those in Europe, to implement GHG reduction strategies [2]. Among various environmental concerns, GHG emissions have gained increasing attention, particularly in the maritime sector, where their regulation remains a critical issue [3,4]. According to the International Maritime Organization (IMO), carbon dioxide (CO_2) emissions from vessels rose by approximately 9.7%, from 962 Mt in 2012 to 1056 Mt in 2018, accounting for around 3% of global anthropogenic CO_2

emissions [5]. Projections indicate that emissions from the shipping sector could increase by up to 17% by 2050. In response, the IMO has introduced regulatory measures such as the Energy Efficiency Design Index, Energy Efficiency Existing Ship Index, and Carbon Intensity Indicator as part of its global carbon reduction strategy [6].

To achieve net-zero GHG emissions, annual statistics are compiled by analyzing CO_2 emissions across various industries, including on-road mobile sources such as automobiles [7]. In sectors such as automotive, construction, and manufacturing, extensive research has been conducted to measure and analyze emissions, leading to the development of emission factors with relatively low uncertainty [8–10]. However, direct measurement of emissions from vessels presents significant challenges due to the high costs and time required for installing monitoring equipment, evaluating emissions under varying environmental conditions, and establishing emission factors. Consequently, many studies have employed modeling techniques that incorporate vessel navigation direction and airflow characteristics to estimate exhaust emissions indirectly.

Vessel emissions are typically estimated using two approaches: the top-down and bottom-up methods [11]. The top-down method calculates emissions based on fuel sales or global fuel consumption data [12]. In contrast, the bottom-up method estimates emissions using operational data from individual vessels, offering greater accuracy in reflecting vessel-specific characteristics and operating conditions [5]. Consequently, the bottom-up approach is more widely adopted for regional and vessel-specific emission assessments [1–14]. However, the bottom-up method has several limitations. It does not account for emission variations caused by technical differences in engine performance, and calculation errors may arise due to uncertainties in emission factors and reliance on average values. Additionally, discrepancies between reported and actual fuel consumption can lead to significant deviations in emission estimates [5].

To address these limitations, recent research has explored emission prediction and assessment models leveraging artificial intelligence (AI), including deep learning and machine learning. Shen et al. (2023) developed a prediction model combining a convolutional neural network (CNN) and long short-term memory (LSTM) using engine data such as revolutions per minute (RPM), torque, fuel consumption, and nitrogen oxide (NO_x) emissions from a diesel engine test bench. The model achieved a coefficient of determination (R^2) of 0.977 and a mean absolute percentage error (MAPE) of 18.4% [15]. Chen et al. (2024) proposed an artificial neural network (ANN) model for predicting NO_x and carbon monoxide (CO) emissions, utilizing emission measurement data, RPM, shaft power, speed, and wind direction from operating vessels. Their approach, incorporating vessel-related and weather variables, significantly outperformed the traditional bottom-up method in prediction accuracy [16]. Cammin et al. (2023) employed automatic identification system (AIS) data—including vessel speed, position, route, engine power, operating time, mode data, vessel type, and gross tonnage—alongside emissions estimated via the bottom-up method. They developed prediction models using ANN, multiple linear regression, and support vector regression, demonstrating that ANN-based models effectively mitigate the limitations of traditional bottom-up approaches [17]. Šilas et al. (2023) collected particulate matter emission data by integrating vessel tonnage, size, power, weather conditions, and AIS data with measured exhaust gas plumes. Their ANN-based prediction model, utilizing 17 input variables, achieved higher accuracy than conventional bottom-up methods [18]. Recently, transformer-based models—originally developed for natural language processing (NLP) and computer vision (CV) [19–26]—have been actively extended to emission prediction tasks as well. For example, Z. Li et al. (2022) proposed a time series forecasting (TSF) transformer model to predict exhaust gas emissions from commercial trucks. The proposed model outperformed traditional machine learning models such as gradient-boosted regres-

sion tree (GBRT), support vector machine (SVM), extreme gradient boosting (XGBoost), as well as deep learning approaches like LSTM [27]. Similarly, J. Li et al. (2024) applied a transformer-based model to predict NO_x and CO emissions from gas turbines. Compared to LSTM and CNN, their model demonstrated both superior prediction accuracy and faster execution time [28]. In addition, physics-informed neural networks (PINN) have gained attention for their ability to incorporate physical constraints into deep learning models. For instance, Zhu et al. (2024) proposed a PINN-based model to predict NO_x emissions from coal-fired boilers by embedding a monotonic relationship between NO_x emissions and three typical operating parameters. Their result demonstrated superior prediction accuracy and generalization capability compared to conventional machine learning models [29].

However, existing emission prediction models have several limitations. Most studies primarily rely on external environmental variables, such as fuel consumption and wind direction, to estimate CO_2 emissions, with relatively little focus on factors directly influencing engine combustion. Notably, few studies have explicitly analyzed the correlations between engine parameters—such as exhaust gas temperature, maximum cylinder pressure, and operating conditions—and emission characteristics for modeling purposes. This gap is largely due to technical and experimental challenges in collecting comprehensive engine operation data.

This study addresses the limitations of existing research by proposing a deep learning model for predicting CO_2 emissions from vessel engines using engine operation data. The model incorporates 109 engine parameters, including fuel consumption, operating conditions (RPM, torque, power, etc.), and cylinder-related variables such as exhaust gas temperature, maximum cylinder pressure, and maximum compression pressure, to enhance prediction accuracy. Both single-architecture models (CNN, LSTM, and temporal convolutional network [TCN]) and hybrid architecture (TCN–LSTM) were implemented to estimate CO_2 emissions based on engine operation data. The model's reliability was validated by comparing predictions with measured data collected under actual operating conditions. Through this study, we propose an effective method for estimating CO_2 emissions—a critical foundation for carbon reduction policies—and analyze the factors influencing emission levels, thereby providing insight for formulating strategies to reduce overall emissions.

The remainder of this paper is organized as follows:

Section 2 describes the experimental setup and data preprocessing.

Section 3 introduces the deep learning model's architectures used in this study.

Section 4 presents and discusses the prediction results and performance of the proposed prediction model

Finally, Section 5 concludes the study and outlines future research directions.

2. Materials and Methods

2.1. Experimental Equipment and Data Description

This study was conducted on the Hyundai–MAN B&W 6S40ME main propulsion engine, installed on a 133-m long vessel with a training capacity of approximately 240 people. The engine was a mid-sized, six-cylinder diesel unit with a maximum power output of 6618 kW at 146 RPM. Unlike conventional mechanical engines, it featured electronic control, allowing precise regulation of fuel injection timing and quantity. The engine operated on light fuel oil (bunker A)—a blend of 70% diesel and 30% heavy fuel oil (bunker C). The detailed specification of experimental vessel and fuel is summarized in Table 1.

Table 1. Vessel and fuel specifications.

Vessel Specification	
Building date	28 December 2018
Gross tonnage	9196 ton
LOA	133 m
Breadth	19.4 m
Speed	17.6 kn
Engine type	Hyundai–MAN B&W 6S40ME–B9.5 (146 RPM/6618 kW)
Fuel Specification	
Fuel type	Bunker A (70% diesel + 30% heavy fuel oil)
Gravity API @60 °F, SG @15/4 °C	0.8673 kg/L
Kinematic viscosity	4 cSt @50 °C

LOA: Length overall; API: American Petroleum Institute; SG: Specific gravity.

Exhaust emission data were collected using a Portable Emission-Measuring System (PEMS), specifically the Semtech DS+ (Sensors Inc., Saline, MI, USA) and the Horiba OBS-One (Horiba Ltd., Kyoto, Japan). Engine operation parameters, including RPM, fuel consumption, load, torque, and power, were recorded using the vessel's Alarm Monitoring System, which continuously tracks and logs equipment status. In total, 109 engine-related parameters were acquired from the main propulsion engine.

Engine and exhaust emissions data were collected as continuous time-series data throughout the experiment, as summarized in Table 2. The emission and engine operation data were collected over approximately 20,000 s, encompassing diverse conditions from low-speed maneuvering (dead slow) to full-speed cruising (navigation full), and transitional phase during departure and arrival. This range ensures that the dataset reflects realistic variations in operating and emission conditions. These data included key parameters such as RPM, torque, power, load, and maximum cylinder pressure, providing insights into overall engine performance and condition.

Table 2. Engine and exhaust emissions data for experiment.

No.	Time	CO_2 (g)	RPM (rpm)	Torque (N·m)	Power (kW)	No. 1 Cylinder MCP (bar)	No. 1 Cylinder Exhaust Gas Temperature (°C)	...
1	11:06:19	807.7	136.8	351.2	3035	164.3	355.3	...
2	11:06:20	807.6	136.8	350.8	5030	164.2	355.3	...
3	11:06:21	807.7	136.8	350.4	5025	16.1	355.2	...
...	...	...	...	...	...	...	...	...
3253	12:05:20	807.8	137.5	347.2	4977.2	161.8	354.9	...
...	...	...	...	...	...	...	...	...
21,879	21:35:40	808.0	136.5	345.6	4947.3	157.8	355.6	...

MCP: Maximum combustion pressure.

The specifications of the PEMS used for emission data acquisition are listed in Table 3. Both the Semtech DS+ and Horiba OBS-One instruments, widely used for vehicle and construction machinery emission evaluations, have demonstrated reliability in numerous studies and experiments. These instruments measure both gaseous and particulate emissions, determining CO_2 concentrations using the nondispersive infrared (NDIR) method. To minimize measurement errors, data collection was limited to a maximum of 4 h, following the maker's recommendations. Additionally, the instruments were calibrated with certified calibration gases, and internal filters were replaced before and after the experiment. The schematic diagram for experiment is shown in Figure 1.

Table 3. Specification of the portable emission-measuring systems.

Model	SEMTECH DS+	HORIBA OBS-ONE
Measuring method (CO_2)	NDIR	NDIR
Maximum range	18% vol.	20% vol.
Accuracy	<±2% of reading or ≤±0.3% of full scale (whichever is larger)	Within ±0.3% of full scale or ±2% of readings (whichever is larger)
Zero drift	≤±0.1% vol. (over 4 h)	Within ±0.5% of full scale per 4 h
Span drift	≤2% of span value or ≤±0.1% vol.(whichever is larger, over 4 h)	Within ±1% of full scale per 4 h
Response time	$T_{0–90} \leq 10$ s	Within 6 s
Recording frequency	1–5 Hz	100 ms

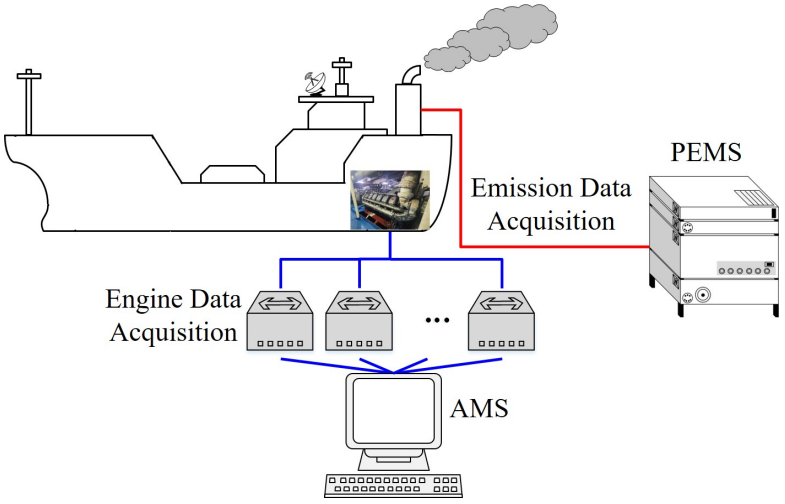


Figure 1. Experimental setup for data acquisition.

To ensure efficient training and reproducibility of the deep learning model, all experiments were conducted in a controlled computing environment. The hardware used included an Intel Core i9–12900K CPU, and an NVIDIA GeForce 2070 SUPER GPU with 8 GB of memory. On the software side, the model was developed using Python 3.9.13 and implemented with TensorFlow 2.18.0 and Keras 3.8.0. CUDA version 12.8 was utilized to enable GPU-accelerated computation. The detailed specifications of the hardware and software environment are summarized in Table 4.

Table 4. Hardware and software configuration used for training and evaluating the proposed deep learning model.

Component	Specification
CPU	Intel Core i9–12900K
GPU	NVIDIA GeForce RTX 2070 SUPER (8 GB)
RAM	32 GB
Operating system	Window 10 Pro 64-bit
Python version	3.9.13
Deep learning framework	TensorFlow 2.18.0/Keras 3.8.0
CUDA version	12.8

The dataset comprises one training dataset with the training dataset consisting of 109 rows and 11,958 columns and a test dataset with 109 rows and 9921 columns. As shown in Table 5, the Anderson–Darling, and Shapiro–Wilk tests for parametric techniques, along with the Kolmogorov–Smirnov (K–S) test for nonparametric techniques, confirmed

that the training data did not follow a normal distribution. Consequently, Spearman's correlation analysis was conducted to identify the factors influencing CO_2 emissions from vessel engines.

Table 5. Results of the normality test for the experimental data.

	Instantaneous CO_2	Main Engine Load	Shaft Power	Fuel Flow	M/E T/C RPM	···
Anderson–darling	Rejected null hypothesis					
Shapiro–Wilk (p-value)	3.47E-61	1.64E-64	9.62E-78	8.88E-93	2.10E-66	···
K–S (p-value)	0	0	0	0	0	···

M/E: Main engine; T/C: Turbocharger.

The Anderson–Darling test is commonly used to assess whether a dataset follows a specific distribution, as described by the following formula [30–32]. The weight function is defined as follows:

$$\Psi(u) = \frac{1}{u(1-u)}. \quad (1)$$

where $\Psi(u)$ is some nonnegative weight function chosen by the experimenter to accentuate the values, u denotes the cumulative distribution function value $F(x)$ at a given data point x.

The Anderson–Darling statistic (A^2) is given by

$$A^2 = n\int_{-\infty}^{\infty} [F_n(x) - F(x)]^2 \Psi(F(x)) dF(x). \quad (2)$$

where n is the sample size, $F_n(x)$ and $F(x)$ refer to the empirical and theoretical cumulative distribution function, respectively.

Substituting the weight function into Equation (2), the Anderson–Darling statistic is transformed as follows:

$$A^2 = -n + \sum_{i=1}^{n} \frac{1-2i}{n} [\ln(u_i) + \ln(1 - u_{n+1-i})]. \quad (3)$$

where u_i is the cumulative probability for the i-th ordered observation in the sample.

The Shapiro–Wilk test is used to assess data normality, rejecting the null hypothesis and accepting the alternative if the p-value is less than 0.05. In other words, if the p-value is 0.05 or lower, the data do not follow a normal distribution, as shown in Equation (4). Given that each row of the input data contains 11,958 data points, a Shapiro–Wilk test was conducted using a randomly sampled subset of 5000 data points. The equation is presented below [33–35].

$$\mathrm{W} = \frac{\left(\sum_{i=1}^{n} a_i y_i\right)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2}. \quad (4)$$

In Equation (4), W is the Shapiro–Wilk test statistic used for assessing normality. Here, y_i represents the i-th ordered observation, $\overline{y}$ is the sample mean, and a_i is the predetermined weight based on expected values of order statistics from a standard normal distribution.

The Kolmogorov–Smirnov test is a nonparametric method used to quantitatively compare differences between data distributions or assess whether they follow a normal distribution. If the p-value is lower than the significance level, the null hypothesis is rejected, indicating that the data do not follow a normal distribution. The equations are as follows:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i \le x}. \quad (5)$$

$$D = \max|F_n(x) - F(x)|. \quad (6)$$

Here, $F(x)$ represents the empirical distribution function, n is the number of observations, and $I_{Xi \leq x}$ is the indicator function. When D is lower than the significance level, the null hypothesis is rejected, indicating that the data do not follow a normal distribution.

2.2. Analysis of Data Correlation

Because the data did not follow a normal distribution, we analyzed the correlation between variables using Spearman's correlation. While Pearson's correlation is typically used for normally distributed data, Spearman's correlation is more appropriate for non-normally distributed data. From the 109 available variables, we identified the 23 most influential factors affecting CO_2 emissions. As shown in Figure 2, the significant input variables selected for this study include main engine (M/E) load, shaft power, fuel flow, M/E scavenge air receiver inlet pressure, shaft torque, M/E turbocharger (T/C) RPM, M/E piston cooling oil temperature, M/E air cooler cooling water outlet temperature, M/E fuel oil inlet temperature, M/E T/C exhaust gas inlet temperature, exhaust gas boiler outlet temperature, average mean indicated pressure, average maximum cylinder pressure, M/E RPM, average cylinder compression pressure, M/E T/C exhaust gas outlet temperature, M/E jacket cooling fresh water inlet temperature, and M/E No. 1–6 cylinder exhaust gas outlet temperatures.

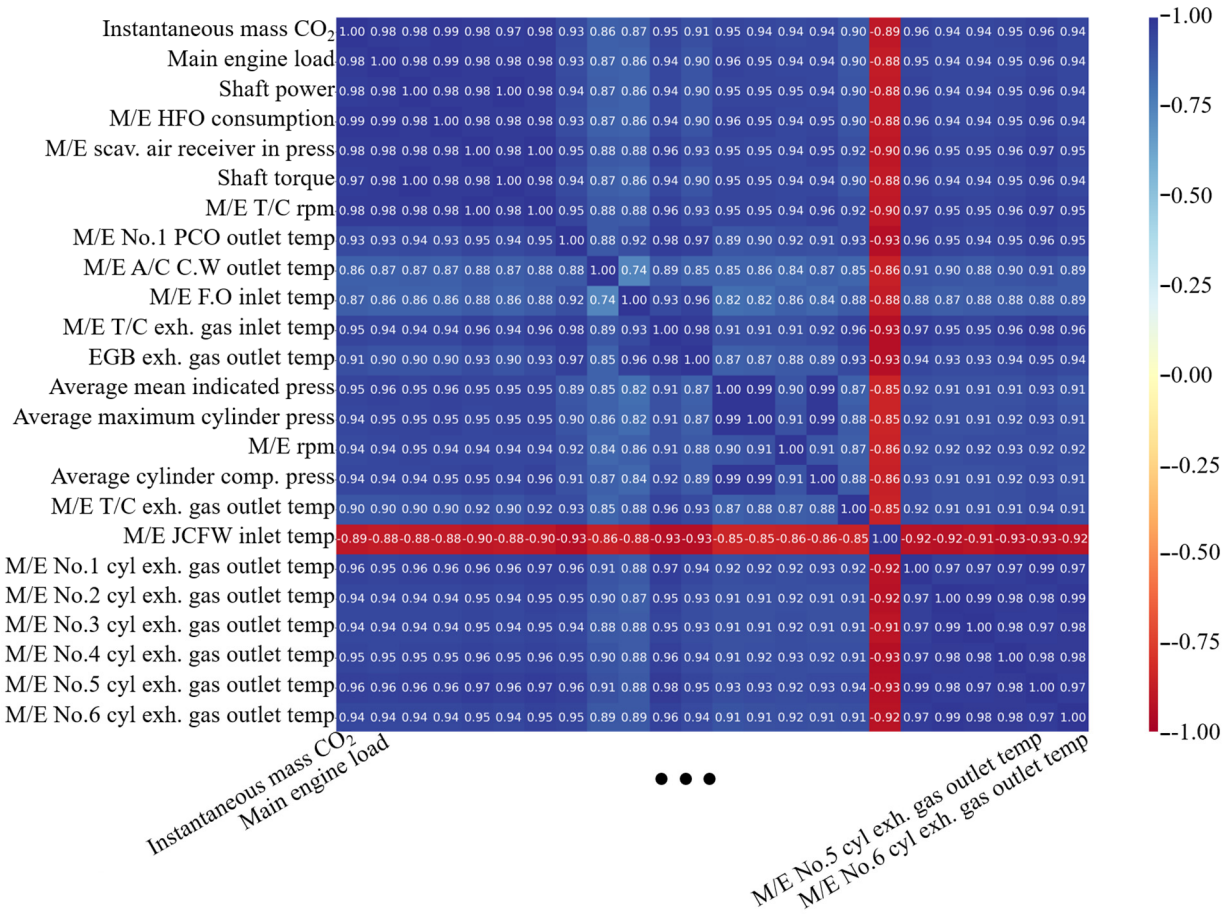


Figure 2. Heat−map of Spearman's correlation analysis for the input experimental data.

According to Table 6, fuel flow was identified as the most influential factor affecting CO_2 emissions, with a Spearman's correlation coefficient of 0.9850, followed by M/E T/C RPM (0.9791), M/E load (0.9772), and M/E air cooler cooling water outlet temperature

(0.8610), which had the weakest correlation among the significant factors. To further ensure the reliability of the model, we also conducted a Pearson correlation analysis, selecting only the variables that directly impact CO_2 emissions for use as training data.

Table 6. Results of the data correlation analysis.

	Spearman's Correlation (p)	Pearson Correlation (R^2)	Rank
Fuel flow	0.9850	0.995201	1
M/E T/C RPM	0.9791	0.985149	2
Main engine load	0.9772	0.997298	3
M/E scavenge air receiver in press	0.9767	0.991914	4
Shaft power	0.9755	0.997075	5
Shaft torque	0.9750	0.990015	6
M/E A/C C.W outlet temperature	0.8610	0.9672	23

A/C: Air cooler; C.W: Cooling water.

2.3. Experimental Data Normalization

Data preprocessing is a crucial step in model training, as it enhances accuracy and accelerates the learning process [36,37]. Among various preprocessing techniques, we applied input data normalization to optimize model performance by adjusting the data range and distribution. This approach improves training speed, enhances learning accuracy, and ensures model stability. Normalization is widely used in image processing and numerical data models [38]. Because the training and test datasets did not follow a normal distribution, we applied min-max normalization, which scales the input data to a range of [0, 1] based on data size. The transformation is defined in Equation (7).

$$x_{nm} = \frac{x_i - x_{min}}{x_{max} - x_{min}}. \quad (7)$$

Here, x_{nm} represents the normalized input data; x_i is the input data; and x_{min} and x_{max} are the minimum and maximum values in the input data, respectively.

For effective machine learning with neural networks, the emission and engine data used in the training model were scaled within the range of [0, 1], as shown in Table 7.

Table 7. Normalization result of the experimental data.

No.	Time	CO_2	RPM	Torque (N·m)	Power (kW)	No. 1 Cylinder MCP (bar)	No. 1 Cylinder Exhaust Gas Temperature (°C)	...
1	11:06:19	0.6366	0.9971	0.9070	0.8968	0.5680	0.9970	...
2	11:06:20	0.6371	0.9971	0.9060	0.8959	0.5675	0.9967	...
3	11:06:21	0.6381	0.9971	0.9050	0.8950	0.5670	0.9965	...
...	...	...	...	...	...	...	...	...
3253	12:05:20	0.6393	0.9821	0.8969	0.8865	0.5493	0.9954	...
...	...	...	...	...	...	...	...	...
21,879	21:35:40	0.8131	0.9752	0.8541	0.8385	0.5355	0.9876	...

3. Summary of the Artificial Neural Network (ANN) for the CO_2 Emission Prediction Model

ANN is a mathematical model that mimics the behavior of biological neurons to perform complex functions. Initially, fully connected architectures such as multilayer perceptrons (MLPs) were commonly used. However, as AI applications have expanded and data complexity has increased, more advanced architectures have been developed [39,40]. This study considers CNNs, LSTM networks, and TCNs as key architectures, ultimately developing a CO_2 emission prediction model based on a TCN–LSTM hybrid approach. Model layer parameters were not assigned arbitrarily but optimized to achieve the best predictive performance.

3.1. Convolutional Neural Network (CNN)

As depicted in Figure 3, CNN is a type of deep ANN that consists of three main layers: convolutional layers, pooling layers, and fully connected layers [41,42]. Unlike traditional MLPs, CNNs leverage shared weights to efficiently process high-dimensional data, reducing computational complexity. In CNN architecture, the convolutional layer extracts patterns from the input features, while the pooling layer reduces the spatial dimensions of the intermediate representations. This lowers the number of parameters and improves the training efficiency. Finally, the fully connected layers perform a linear transformation to produce the final output [15,40].

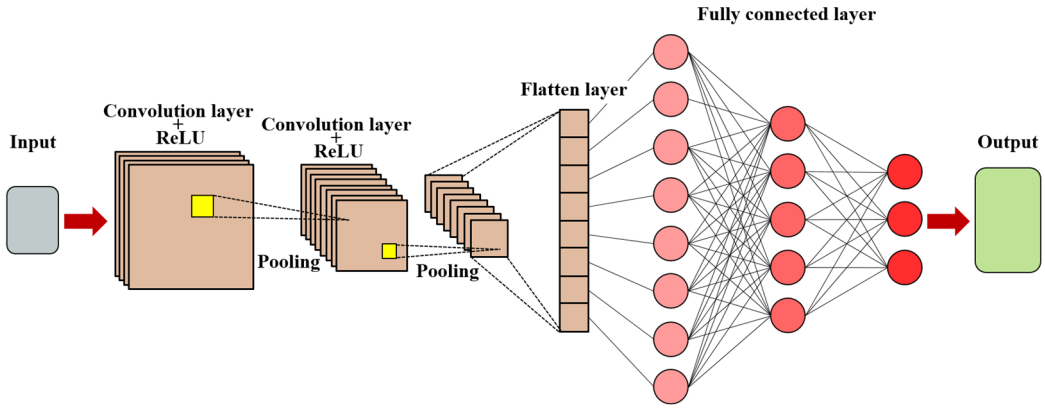


Figure 3. Structure of the one-dimensional (1D) convolutional layer.

3.2. Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) designed to handle sequential data by considering temporal dependencies. As shown in Figure 4, unlike traditional RNNs, LSTM networks incorporate three specialized gates—input gate, output gate, and forget gate—which regulate the flow of information and mitigate the vanishing gradient problem [43]. The memory cell structure of LSTM networks enables the retention of both short-term and long-term dependencies through the linear transfer of cell states and controlled gate interactions. This architecture makes LSTM networks particularly effective for processing and predicting time-series data, especially when learning from sequences with long time steps [41].

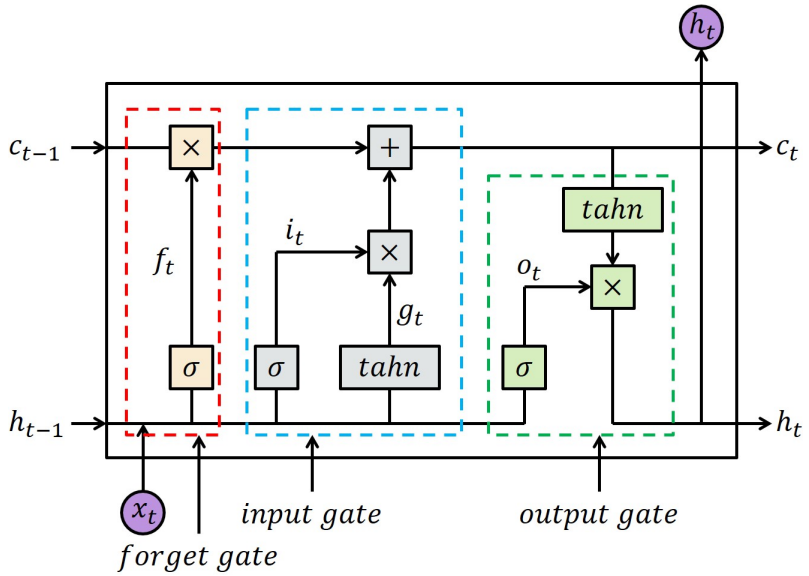


Figure 4. Inner structure of long short-term memory (LTSM).

3.3. Temporal Convolutional Network (TCN)

TCNs are a type of CNN architecture specifically designed for processing time-series data, similar to LSTM [44,45]. Unlike recurrent networks, TCNs process sequential data in parallel, allowing for faster training. They also address the vanishing gradient problem through residual connections and the use of standard backpropagation algorithms, enhancing model stability and efficiency [46,47]. The structure of TCN is illustrated in Figure 5.

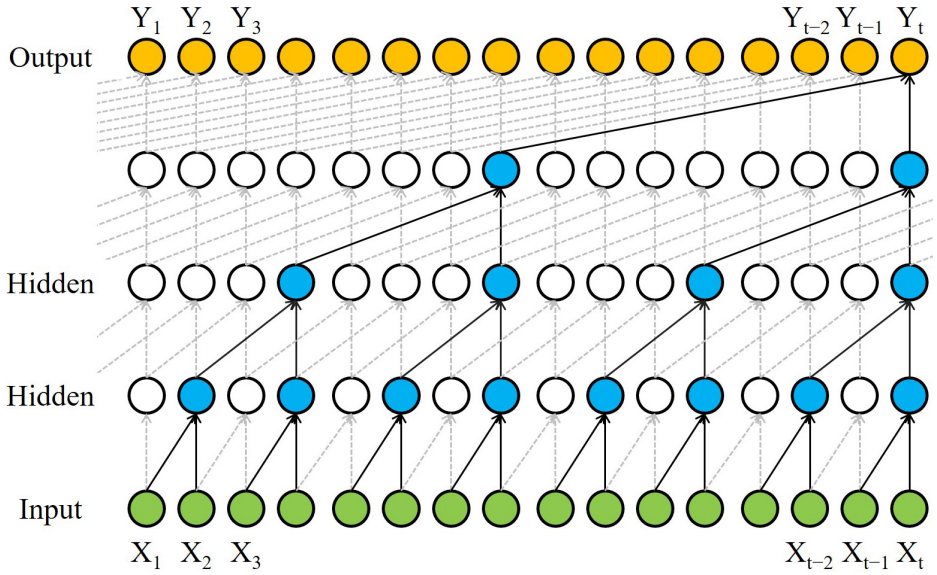


Figure 5. Structure of temporal convolutional networks (TCNs).

3.4. TCN–LSTM

Among the hybrid architecture models proposed in this study, we introduce the TCN–LSTM deep learning model, which integrates TCNs and LSTM networks to effectively predict CO_2 emissions from vessel engine data, as illustrated in Figure 6. This model leverages TCNs to efficiently extract features from complex data structures, while LSTM networks enhance the model's ability to process time-series data and capture long-term dependencies. Additionally, to improve data quality and minimize noise, we applied the

Savitzky–Golay (SG) filter, as proposed by Bi et al. (2021), to the hybrid model before training [48].

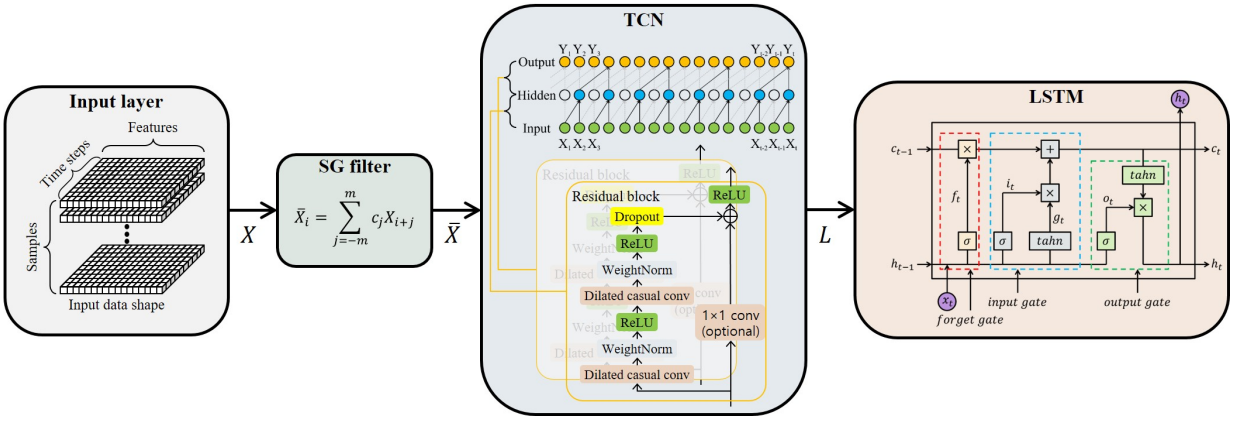


Figure 6. Structure of the TCN–LSTM hybrid model with a Savitzky–Golay (SG) filter.

In the first step, the SG filter is applied to the normalized input data (X_{nm}) to reduce noise, producing $\overline{X}$:

$$\overline{X} = SG(X_{nm}, w, t), \tag{8}$$

where $SG()$ represents the SG filter function, w is the window size, and t is the highest polynomial order used for smoothing.

In the second step, the denoised data $\overline{X}$ is passed through a TCN consisting of two residual blocks, which extract relevant temporal features. The empirical hyperparameters for each residual block are used.

$$\mathrm{L} = RB\left(\overline{X},\ k,\ d\right), \tag{9}$$

where $RB()$ represents the residual block function of the TCN, k is the kernel size, and d is the dilation factor.

In the third step, the extracted feature set L is input into the LSTM network, which captures long-term dependencies and outputs h_t at time t according to Equations (10)–(14). The hyperparameters for the LSTM layer were also empirical values.

$$i_t = \sigma(W_i \cdot [h_{t-1,L_t}] + b_i), \tag{10}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1,L_t}] + b_f\right), \tag{11}$$

where i_t and f_t are the computed values of the input and forget gates, respectively. W_i and W_f are weight matrices, h_{t-1} is the output of the previous cell, L_t is the input at time t, and b_i and b_f are bias vectors.

The next step is to update the cell state C_t via Equation (12).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot (tanh(W_c \cdot [h_{t-1},\ L_t] + b_c)), \tag{12}$$

where W_c is the weight matrix, b_c is the bias vector, and C_{t-1} is the previous cell.

The output gate o_t and the final output h_t are represented as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1,L_t}] + b_o) \tag{13}$$

$$h_t = o_t \cdot \tanh\,(C_t), \tag{14}$$

where W_o is the weight matrix, and b_o is the bias vector.

The output h_t from the LSTM layer is then passed through a fully connected layer with a rectified linear unit (ReLU) activation function to generate the output z_t.

$$z_t = relu(v \cdot h_t + bias), \tag{15}$$

where *relu* is the ReLU activation function, v is the weight matrix, and *bias* refers to the bias vector.

The fully connected layer output z_t is input into the output layer to generate the prediction value y_t.

$$\hat{y}_t = linear(u \cdot z_t + q), \tag{16}$$

where *linear*() is the linear function, u is the weight parameter of the fully connected layer, and q is the bias parameter.

The TCN–LSTM network for CO_2 emission prediction using engine data was configured according to the specifications in Table 8. The architecture includes four temporal convolutional layers and two bidirectional LSTM layers, along with an attention mechanism to focus on critical patterns within the training data. The optimal parameters—such as the number of filters, kernel size, and dilation rate of the TCN blocks—were determined through multiple experiments. Table 8 provides a detailed breakdown of the parameters across all layers. To maintain optimal performance and prevent overfitting, we implemented a learning rate schedule and early stopping during training.

Table 8. Proposed parameters for the TCN–LSTM architecture.

Type	Filter	Kernel Size	Dilation	Activation	#Parameter
TCN block	256	2	1	ReLU	143,872
TCN block	256	2	2	ReLU	262,656
TCN block	256	2	4	ReLU	262,656
TCN block	256	2	8	ReLU	262,656
Bidirectional LSTM	256	–	–	–	526,336
Dropout (0.4)	–	–	–	–	–
Attention	–	–	–	–	–
Bidirectional LSTM	128	–	–	–	263,168
Dropout (0.5)	–	–	–	–	–
Dense	64	–	–	–	16,448
Dense	1	–	–	–	65

While the TCN–LSTM based hybrid model is expected to offer superior prediction accuracy, it carries interpretability limitations commonly associated with deep learning models. To address this to some extent, Table 6 presents the correlation-based feature importance rankings derived from the training dataset, providing insights into which engine parameters most significantly influence CO_2 emissions.

However, additional post hoc interpretability analysis is needed to elucidate the relationship between input features and the model's predictions. Future work will consider applying explainable AI (XAI) techniques—such as SHAP (SHapley Additive exPlanations) or integrated gradients—to analyze how individual features contribute to each prediction. This would enhance the model's transparency and improve its practical applicability for decision-making in real-world maritime operations.

The proposed TCN–LSTM hybrid model was trained using the Adam optimizer with an initial learning rate of 0.0001. A dynamic learning rate schedule was applied, reducing the rate by half after 10 epochs and to 10% of the initial value after 30 epochs. The training

was conducted with a batch size of 64 for up to 100 epochs. To prevent overfitting and improve generalization, early stopping was applied with a patience of 5 epochs, and the best–performing model weights were restored. Model checkpoints were also saved during training.

4. Results and Discussion

The validation loss change rates for the CNN, LSTM, TCN, and TCN–LSTM models are shown in Figure 7. These rates were analyzed to assess the stability of the models. With the exception of the TCN model, all models exhibited a sharp decrease in loss during the initial training phase, indicating rapid optimization in the early learning stages. The TCN model, however, showed the largest initial change rate in loss at the beginning of training, followed by stable learning, with its change rate converging to zero between epochs 13–17. For the LSTM model, the first 0–5 epochs demonstrated gradual optimization due to its recurrent structure. Most models exhibited stabilization, with the change rate approaching zero after 10 epochs. Among all models, the TCN–LSTM architecture displayed the fastest decline in change rate, indicating stable learning. These findings suggest that the hybrid model offers better generalization ability and a lower risk of overfitting compared to individual architectures.

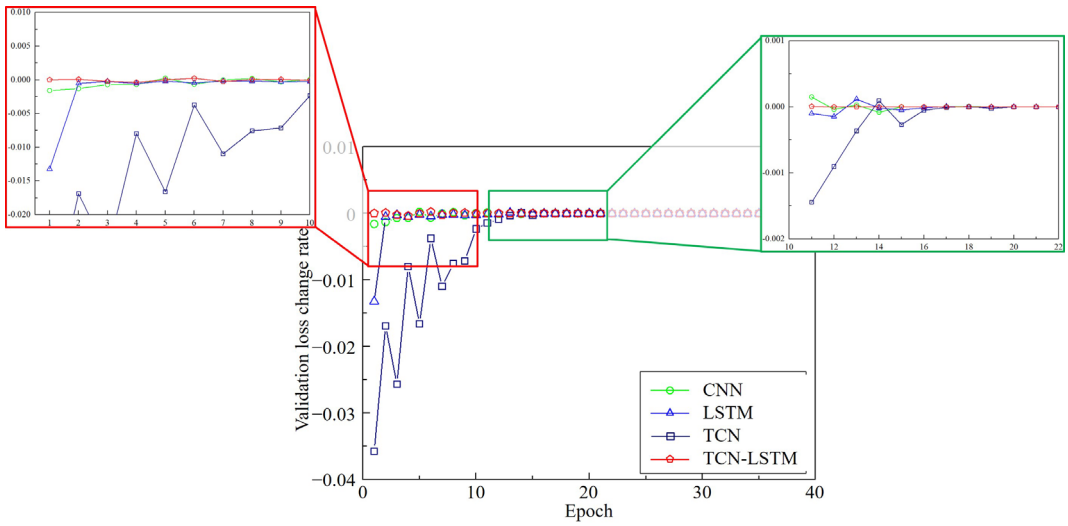


Figure 7. Validation loss change rate of all the prediction models.

To prevent overfitting, we applied early stopping before 10 epochs for the hybrid model and after 10 epochs for the single models. The validation change rate analysis demonstrated that the TCN–LSTM model proposed in this study exhibited a consistently decreasing change rate, confirming their stability and applicability in predicting CO_2 emissions from engine data. However, for models that displayed fluctuations in the change rate during training, we anticipate that the introduction of advanced optimization techniques and further hyperparameter tuning will be necessary to enhance their performance and stability.

The prediction results of the CNN, LSTM, and TCN single architectures and the TCN–LSTM hybrid model are shown in Figure 8. Figure 8a shows the overall time-series comparison, while Figure 8b–e provides zoomed-in views of representative segments with notable prediction performance differences. In Figure 8b, all models exhibit noticeable prediction inaccuracies in this segment, with significant deviations from the measured values. This discrepancy may stem from the model's limited capacity to respond effectively to abrupt transitions or irregular input patterns. These results indicate that the architectures

including the proposed model may struggle to fully capture nonlinear behaviors or unmodeled dynamics in such regions. To address this limitation, future work should consider incorporating additional representative training data and performing cross–model analysis to enhance robustness and generalization under transient conditions. While all models generally performed well, some exhibited underprediction or overprediction tendencies. Compared to the hybrid model, the LSTM and TCN single architectures (excluding CNN) showed greater deviations, particularly in sections where engine RPM gradually increased (see Figure 8d) or decreased (see Figure 8c). This discrepancy likely stems from differences in training methods influenced by each model's structural characteristics and the nature of the data. LSTM model excels at capturing long-term dependencies by retaining past information. However, in sections with rapid changes, their predictive accuracy decreases, as they rely heavily on historical data, leading to noticeable deviations. TCN model is effective at learning localized patterns, but their predictions may be unstable due to sensitivity to noise or outliers in the input data [49]. Further parameter optimization could mitigate this issue. CNN models, although traditionally used for image processing, have proven highly effective for time-series prediction [50]. Their convolutional layers excel at detecting localized features, allowing them to capture pattern changes in short segments of extensive time-series data. Additionally, CNNs naturally filter noise, leading to superior predictive performance compared to LSTM and TCN models, even in rapidly changing data [51].

A hybrid architecture model can compensate for the weaknesses of a single model by leveraging their complementary strengths. The TCN–LSTM model effectively captures complex time-series patterns by simultaneously considering short-term fluctuations and long-term dependencies, thereby enhancing prediction accuracy. As shown in Figure 8, the hybrid model demonstrates high predictive accuracy, particularly in variation sections where single models struggle. This confirms their robustness in handling dynamic changes in engine data (see Figure 8c,e).

To enable a quantitative evaluation of each model, Table 9 presents the R^2, mean absolute error (MAE), root mean squared error (RMSE), MAPE, and Pearson's correlation coefficient (R) for the test results. Additionally, Figure 9 provides a bar graph for a visual comparison of the results. All four models exhibited high accuracy, with $R^2 \geq 0.9$, and the TCN–LSTM model outperformed the others across all metrics. Among the single-architecture models, the CNN model achieved the highest accuracy ($R^2 = 0.9697$) and the smallest errors, as indicated by its MAE (49.6663), RMSE (60.0875), and MAPE (4.2337%)—highlighting its minimal deviation from actual measurements.

Table 9. Evaluation of the models.

Contents	CNN	LSTM	TCN	TCN–LSTM
R^2	0.9697	0.9386	0.9391	0.9726
MAE	49.6663	60.0086	63.0889	47.3447
RMSE	60.0875	73.1944	73.0336	58.5737
MAPE	4.2337	7.7985	8.2610	4.2335
R	0.9874	0.9737	0.9711	0.9889

Overall, the hybrid architecture model demonstrated strong performance, with the TCN–LSTM model achieving the highest accuracy ($R^2 = 0.9726$), slightly outperforming the CNN-based model. Additionally, the TCN–LSTM model exhibited the lowest prediction deviation and variance, as indicated by its MAE (47.3447) and RMSE (58.5737). These results suggest that the TCN–LSTM hybrid model provides the most accurate CO_2 emission predictions, particularly in response to variations in engine operation.

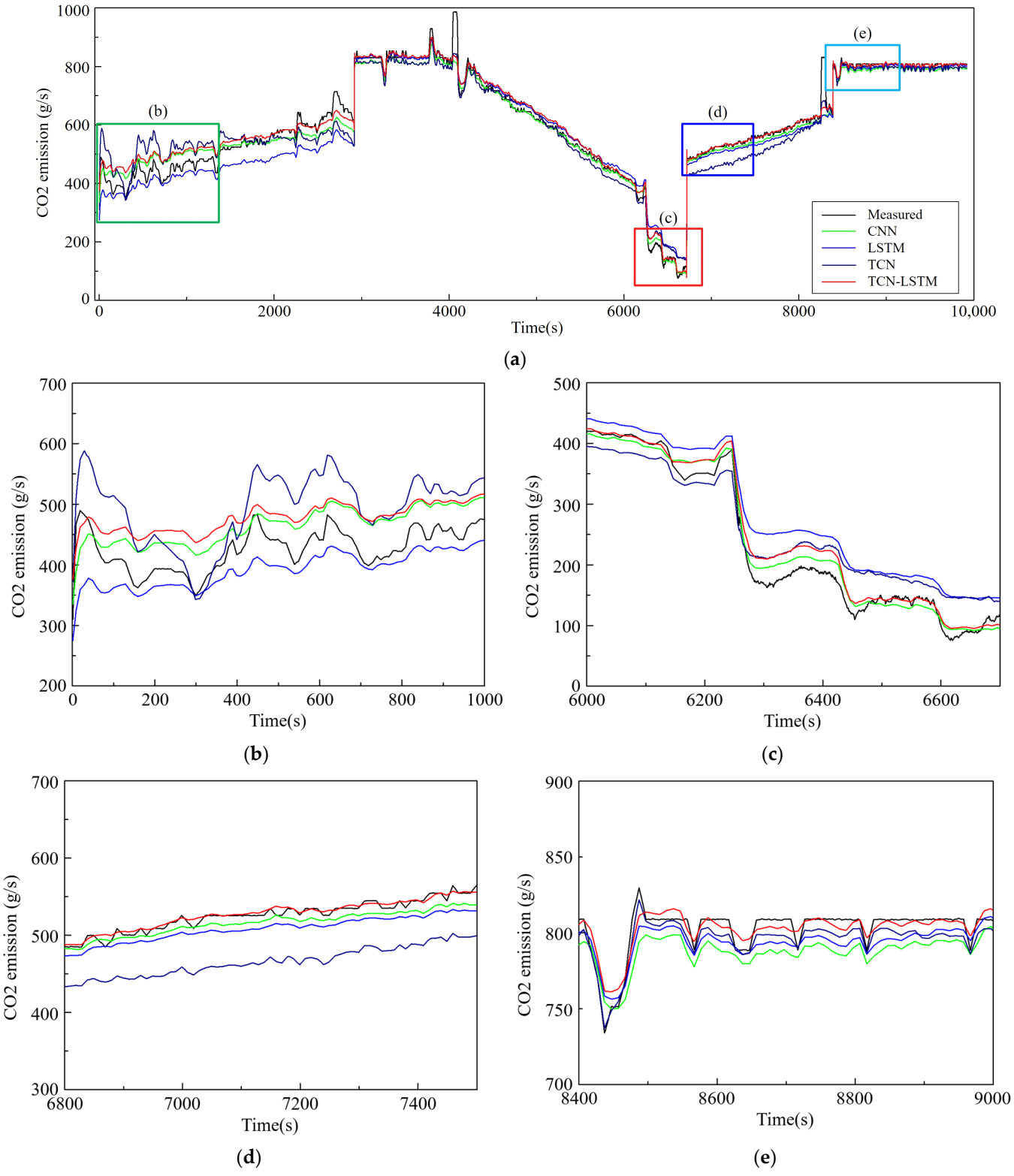


Figure 8. Comparison of the prediction results of the models. (**a**) Over all-time series prediction results of the models. (**b**) Prediction results in the segment with significant variance. (**c**) Prediction results during a load decrease. (**d**) Prediction results during a load increase. (**e**) Prediction results under steady load conditions.

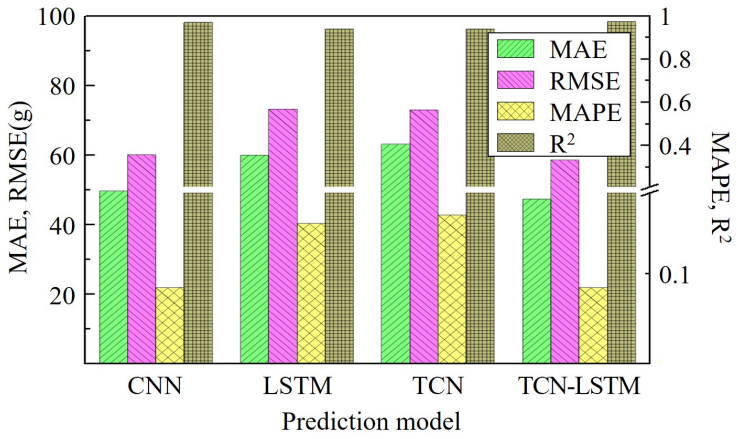


Figure 9. The evaluation results of each prediction models.

The TCN–LSTM hybrid model outperformed all other models across all evaluation metrics, as it effectively leverages the strengths of both TCN and LSTM to capture complex time-series patterns in the input data. Compared to the TCN single model, the TCN–LSTM model achieved improvements of 3.6% in R^2, 24.9% in MAE, 19.8% in RMSE, and 48.8% in MAPE. Additionally, compared to the LSTM single model, it showed enhancements of approximately 3.6% in R^2, 21.1% in MAE, 19% in RMSE, and 45.7% in MAPE. These results highlight the effectiveness of the hybrid approach in improving CO_2 emission prediction accuracy.

The TCN–LSTM hybrid model also has the best evaluation metrics, with higher prediction accuracy than the single-architecture models. In contrast, the TCN and LSTM model exhibited relatively higher errors and lower R^2 values, indicating their individual limitations in accurately predicting CO_2 emissions.

Figure 10a visually compares the errors between the actual and predicted values across the four models. The LSTM and TCN models exhibit larger residual variances, with noticeable outliers in sections where rapid changes in engine-operating conditions occur. This suggests that these models struggle to adapt to sudden variations in the data, highlighting the need for additional data supplementation and model correlation analysis in future research. In contrast, the hybrid model shows residuals that are more evenly distributed around zero (green color sector), indicating better stability and accuracy in capturing fluctuations in engine operation.

Figure 10b presents the mean of the residuals, illustrating the extent of underestimation and overestimation for each model. The single-architecture models exhibited larger discrepancies between underestimation and overestimation compared to the hybrid model. Among them, the TCN–LSTM model had the smallest difference (34.7827) and the highest prediction accuracy, further demonstrating its effectiveness in minimizing prediction errors.

Figure 11 illustrates the regression accuracy of each prediction model, with adjusted R^2 values exceeding 0.9 for all models—indicating generally high prediction accuracy. However, distinct deviations appear in certain sections, likely due to correlations between model structure and data characteristics or suboptimal parameter settings, leading to errors. These deviations were particularly evident during transitional engine states, such as sudden changes in rpm or torque, where the input data distributions shift rapidly. Such conditions are often underrepresented in the training dataset, which may lead to localized prediction biases. In addition, structure characteristics of the hybrid model—such as the temporal sensitivity of LSTM and the local responsiveness of TCN—may contribute to deviations in specific intervals. To address these limitations, future work will involve augmenting the training database with more samples of dynamic operating conditions and exploring

advanced techniques such as attention mechanisms or adaptive loss functions to enhance the model's robustness and generalization performance in these regions.

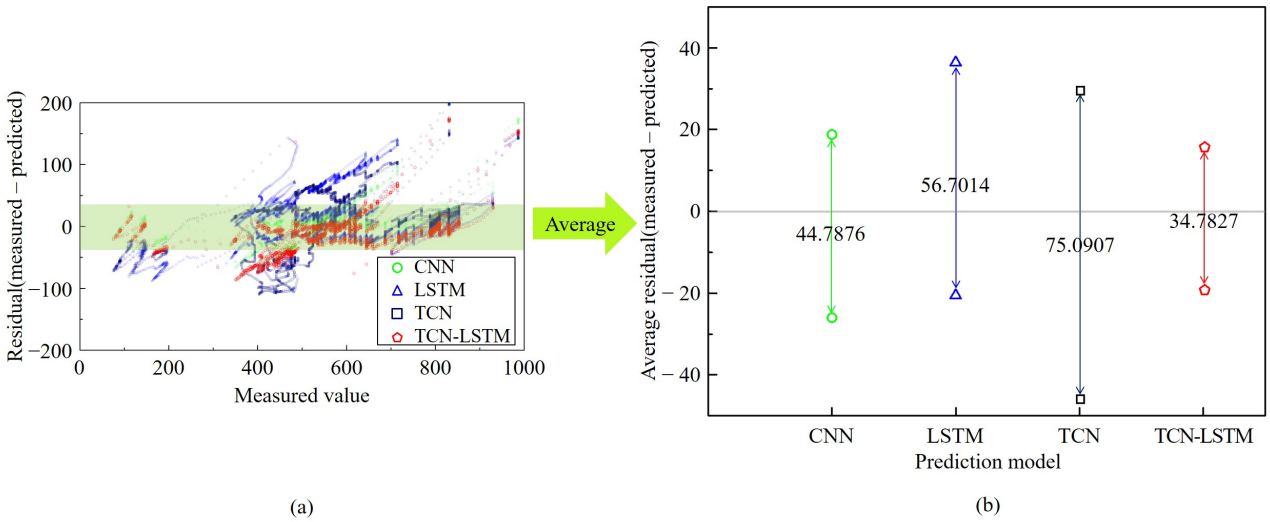


Figure 10. Analysis of the prediction variance for each model. (**a**) Residual variance of each model. (**b**) Average value of underestimation and overestimation for each training model.

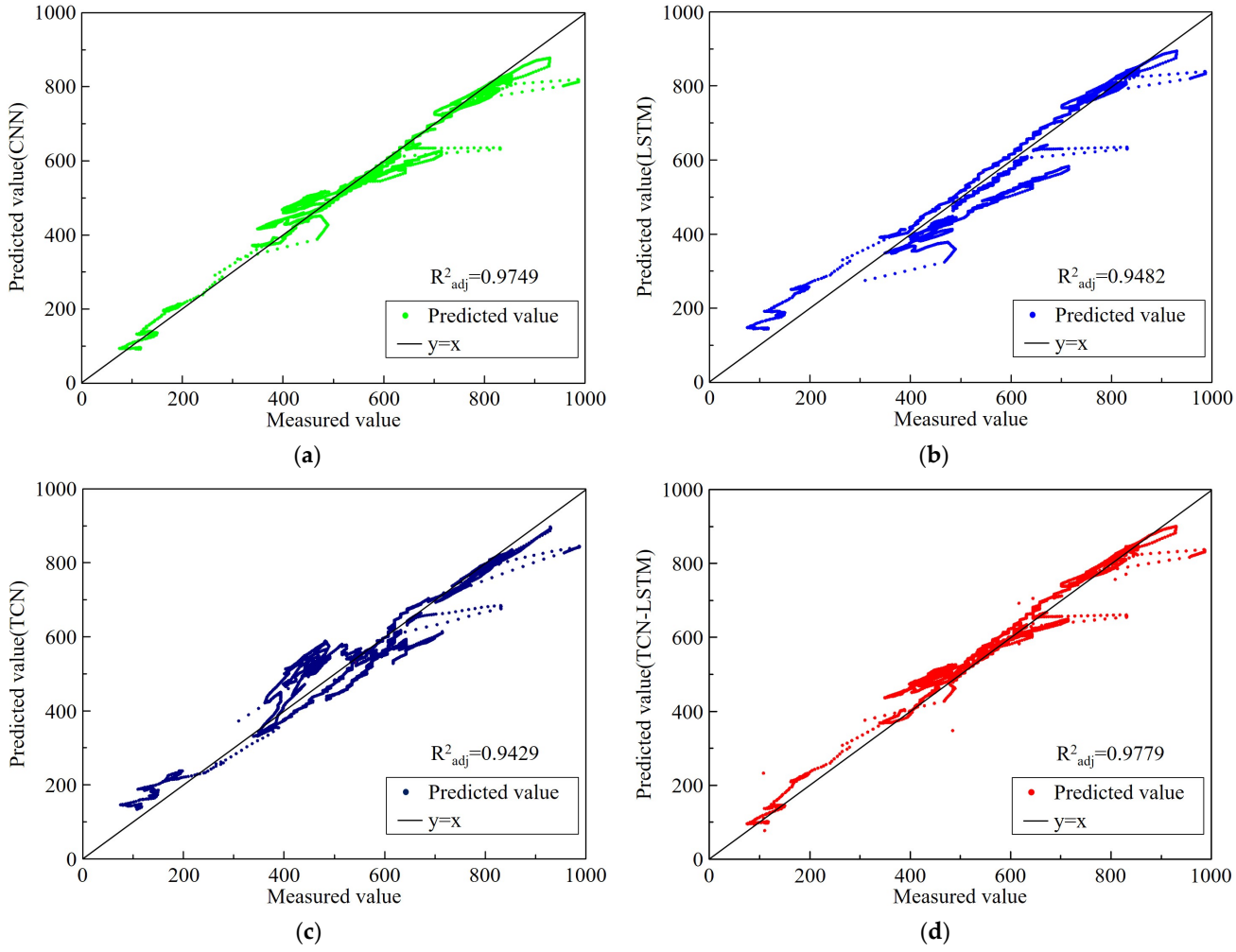


Figure 11. Measured and predicted values of each model: (**a**) the CNN, (**b**) LSTM, (**c**) TCN, (**d**) TCN–LSTM models.

Among the single-architecture models, the CNN model demonstrated more consistent deviations, except in sections where all models exhibited large errors. Unlike LSTM and TCN, it did not show a wide range of errors, suggesting greater robustness. In contrast, LSTM and TCN models exhibited high variance even in high-density data sections (400–800), implying that they lack sufficient explanatory power when used individually for CO_2 emission predictions. However, the hybrid model exhibited uniformly distributed deviations across all sections, including in sparsely distributed data ranges (0–200), with the exception of common error-prone regions. This suggests that the hybrid model effectively mitigated the weaknesses of the individual architectures. Notably, the TCN–LSTM hybrid model achieved the highest adjusted R^2 value (0.9779), confirming its superior performance in accurately predicting CO_2 emissions based solely on engine operation data.

5. Conclusions

In this study, we propose a non-experimental approach to predicting CO_2 exhaust emissions from vessel propulsion engines during operation, addressing a globally significant issue. We explored TCN–LSTM hybrid model, integrating TCNs and LSTM networks, to analyze complex time-series data. The models were trained using engine operation data and exhaust emission measurements, and based on various validation processes, we derived the following conclusions:

1. Among the CNN, LSTM, and TCN single architectures, the CNN model demonstrated the best predictive performance. This is likely due to its ability to recognize local features and extract patterns from noisy data, which are key strengths of CNNs. By extending CNN architecture—traditionally used for image processing—we experimentally verified its effectiveness in estimating engine exhaust emissions.
2. We developed a TCN–LSTM hybrid model by integrating multiple architectures, resulting in more complex neural network structures. These hybrid models exhibited faster convergence and more stable training compared to single architectures. Moreover, they proved to be more robust, achieving higher prediction accuracy and significantly lower residual distributions.
3. The TCN–LSTM hybrid model achieved the best overall performance, significantly improving accuracy metrics. It outperformed the TCN single model by 24.9% in MAE, 19.8% in RMSE, and 48.8% in MAPE, and the LSTM single model by 21.1% in MAE, 19% in RMSE, and 45.7% in MAPE, confirming its superior predictive capability.
4. All models exhibited deviations and reduced prediction accuracy in sections with rapid engine-operating changes. This is likely due to limited data in these sections and inherent correlations between the model structure and input data. These findings suggest the need for further analysis, potential structural refinements, and data augmentation to enhance model performance in these areas.
5. The hybrid model demonstrated significantly higher predictive performance than single architectures. This suggests that a predictive model can effectively capture engine combustion characteristics and exhaust emissions under varying conditions.
6. Future research will focus on analyzing the complex correlations between model structure and data to address observed inconsistencies. Additional measures– such as structural refinements and enhanced data processing–will be explored to further improve prediction performance. Moreover, data from the vessel's external environment (e.g., ocean currents, wind direction, speed, etc.) will be incorporated into the training data to achieve a higher goodness-of-fit in sections with frequent time-series variations.
7. The proposed hybrid prediction model has strong potential for integration into real-time engine monitoring systems and emission control platforms onboard vessels.

It provides valuable insights into emission levels under various engine operating conditions and may also be applied to assess emissions associated with engine aging, components degradation, or system faults. Furthermore, the results of this study can serve as a foundation for future research by offering data-driven insights for the development of emission inventories and carbon reduction strategies.

8. The hybrid model developed in this study was validated on the Hyundai-MAN B&W 6S40ME engine. Its general applicability to other engine types remains to be verified. Future research may involve retraining the model using data from different engine types, optimizing hyperparameters, and conducting comparative performance evaluations to assess the generalization capability of the proposed model.

Author Contributions: Conceptualization, J.O.; Software, S.L.; Validation, J.O.; Formal analysis, S.L.; Investigation, S.L.; Data curation, S.L.; Writing—original draft, S.L.; Writing—review & editing, J.O.; Visualization, S.L.; Supervision, J.O.; Project administration, J.O.; Funding acquisition, J.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Programme through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NO. 2020R1I1A2073426).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusion of the article will be made available by the authors upon request.

Conflicts of Interest: The authors declare on conflict of interest.

Abbreviations

AI	Artificial intelligence
AIS	Automatic identification system
ANN	Artificial neural network
CNN	Convolutional neural network
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
PEMS	Portable Emission-Measuring System
ReLU	Rectified linear unit
RMSE	Root mean squared error
RNN	Recurrent neural network
TCN	Temporal convolutional network

References

1. Oh, J.; Lee, K. Spray characteristics of a urea solution injector and optimal mixer location to improve droplet uniformity and NOx conversion efficiency for selective catalytic reduction. *Fuel* **2014**, *119*, 90–97. [CrossRef]
2. International Energy Agency. IEA Official Wesite. Available online: https://www.iea.org/ (accessed on 22 February 2018).
3. Park, S.; Oh, J. Uniformity index measurement technology using thermocouples to improve performance in urea-selective catalytic reduction systems. *Heat Mass Transf.* **2018**, *54*, 3253–3264. [CrossRef]
4. Deng, S.; Mi, Z. A review on carbon emissions of global warming. *Mar. Dev.* **2023**, *1*, 4. [CrossRef]
5. International Maritime Organization. Fourth IMO Greenhouse Gas Study. 2020. Available online: https://www.imo.org/en/ourwork/Environment/Pages/Fourth-IMO-Greenhouse-Gas-Study-2020.aspx (accessed on 14 March 2025).
6. ISEMAR. Global Sulphur Cap 2020 Etat des lieux [Summary Note ISEMAR No. 216]. 2020. Available online: https://www.isemar.org/wp-content/uploads/2020/01/Note-de-Synthese-216-Global-sulphur-cap-2020-Etat-des-lieux.pdf (accessed on 14 March 2025).
7. Kongboon, R.; Gheewala, S.H.; Sampattagul, S. Greenhouse gas emissions inventory data acquisition and analytics for low carbon cities. *J. Clean. Prod.* **2022**, *343*, 130711. [CrossRef]

8. Watson, J.G.; Chow, J.C.; Chen, L.-W.A.; Lowenthal, D.H.; Fujita, E.M.; Kuhns, H.D.; Sodeman, D.A.; Campbell, D.E.; Moosmüller, H.; Zhu, D.; et al. Particulate emission factors for mobile fossil fuel and biomass combustion sources. *Sci. Total Environ.* **2011**, *409*, 2384–2396. [CrossRef] [PubMed]
9. Beddows, D.C.S.; Harrison, R.M. PM10 and PM2.5 emission factors for non-exhaust particles from road vehicles: Dependence upon vehicle mass and implications for battery electric vehicles. *Atmos. Environ.* **2021**, *244*, 117886. [CrossRef]
10. Gao, C.-K.; You, H.; Na, H.-M.; Xu, Q.-J.; Li, X.-J.; Liu, H.-T. Analysis of passenger vehicle pollutant emission factor based on on-board measurement. *Atmos. Pollut. Res.* **2022**, *13*, 101421. [CrossRef]
11. Nunes, R.A.; Alvim-Ferraz, M.C.; Martins, F.G.; Sousa, S.I. The activity-based methodology to assess ship emissions—A review. *Environ. Pollut.* **2017**, *231*, 87–103. [CrossRef]
12. Peng, X.; Wen, Y.; Wu, L.; Xiao, C.; Zhou, C.; Han, D. A sampling method for calculating regional ship emission inventories. *Transp. Res. D* **2020**, *89*, 102617. [CrossRef]
13. Alver, F.; Saraç, B.A.; Şahin, Ü.A. Estimating of shipping emissions in the Samsun Port from 2010 to 2015. *Atmos. Pollut. Res.* **2018**, *9*, 822–828. [CrossRef]
14. Moreno-Gutiérrez, J.; Calderay, F.; Saborido, N.; Boile, M.; Valero, R.R.; Durán-Grados, V. Methodologies for estimating shipping emissions and energy consumption: A comparative analysis of current methods. *Energy* **2015**, *86*, 603–616. [CrossRef]
15. Shen, Q.; Wang, G.; Wang, Y.; Zeng, B.; Yu, X.; He, S. Prediction model for transient NOx emission of diesel engine based on CNN-LSTM network. *Energies* **2023**, *16*, 5347. [CrossRef]
16. Chen, Z.S.; Lam, J.S.L.; Xiao, Z. Prediction of harbour vessel emissions based on machine learning approach. *Transp. Res. D* **2024**, *131*, 104214. [CrossRef]
17. Cammin, P.; Yu, J.; Voß, S. Tiered prediction models for port vessel emissions inventories. *Flex. Serv. Manuf. J.* **2023**, *35*, 142–169. [CrossRef]
18. Šilas, G.; Rapalis, P.; Lebedevas, S. Particulate matter (PM1, 2.5, 10) concentration prediction in ship exhaust gas plume through an artificial neural network. *J. Mar. Sci. Eng.* **2023**, *11*, 150. [CrossRef]
19. Ashish, V. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
20. Tetko, I.V.; Karpov, P.; Van Deursen, R.; Godin, G. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nat. Commun.* **2020**, *11*, 5575. [CrossRef]
21. Acheampong, F.A.; Nunoo-Mensah, H.; Chen, W. Transformer models for text-based emotion detection: A review of BERT-based approaches. *Artif. Intell. Rev.* **2021**, *54*, 5789–5829. [CrossRef]
22. Ma, X.; Zhang, P.; Zhang, S.; Duan, N.; Hou, Y.; Zhou, M.; Song, D. A tensorized transformer for language modeling. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 2232–2242.
23. Yu, J.; Li, J.; Yu, Z.; Huang, Q. Multimodal transformer with multi-view visual representation for image captioning. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 4467–4480. [CrossRef]
24. Li, Y.; Yao, T.; Pan, Y.; Mei, T. Contextual transformer networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 1489–1500. [CrossRef] [PubMed]
25. Zhong, H.; Chen, J.; Shen, C.; Zhang, H.; Huang, J.; Hua, X.-S. Self-adaptive neural module transformer for visual question answering. *IEEE Trans. Multimed.* **2020**, *23*, 1264–1273. [CrossRef]
26. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [CrossRef] [PubMed]
27. Li, Z.; Zhang, X.; Dong, Z. TSF-transformer: A time series forecasting model for exhaust gas emission using transformer. *Appl. Intell.* **2023**, *53*, 17211–17225. [CrossRef]
28. Li, J.; Han, J.; Niu, D.; Jiang, X.Z. Fast and accurate gas turbine emission prediction based on a light and enhanced Transformer model. *Fuel* **2024**, *376*, 132750. [CrossRef]
29. Zhu, B.; Ren, S.; Weng, Q.; Si, F. A physics-informed neural network that considers monotonic relationships for predicting NOx emissions from coal-fired boilers. *Fuel* **2024**, *364*, 131026. [CrossRef]
30. Anderson, T.W.; Darling, D.A. A test of goodness of fit. *J. Am. Stat. Assoc.* **1954**, *49*, 765–769. [CrossRef]
31. Jiang, J. *Large Sample Techniques for Statistics*; Springer: New York, NY, USA, 2010; pp. 357–392. [CrossRef]
32. Berlinger, M.; Kolling, S.; Schneider, J. A generalized Anderson-Darling test for the goodness-of-fit evaluation of the fracture strain distribution of acrylic glass. *Glass Struct. Eng.* **2021**, *6*, 195–208. [CrossRef]
33. Yap, B.W.; Sim, C.H. Comparisons of various types of normality tests. *J. Stat. Comput. Simul.* **2011**, *81*, 2141–2155. [CrossRef]
34. Razali, N.M.; Wah, Y.B. Power comparisons of some selected normality tests. *J. Stat. Model. Anal.* **2011**, *2*, 21–33.
35. Shapphiro, S.; Wilk, M.B. An analysis of variance test for normality. *Biometrika* **1965**, *52*, 591–611. [CrossRef]
36. Zhong, X.; Gallagher, B.; Eves, K.; Robertson, E.; Mundhenk, T.N.; Han, T.Y. A study of real-world micrograph data quality and machine learning model robustness. *NPJ Comput. Mater.* **2021**, *7*, 161. [CrossRef]

37. Pal, K.K.; Sudeep, K.S. Preprocessing for image classification by convolutional neural networks. In Proceedings of the IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 20–21 May 2016; pp. 1778–1781. [CrossRef]
38. Pei, X.; Zhao, Y.H.; Chen, L.; Guo, Q.; Duan, Z.; Pan, Y.; Hou, H. Robustness of machine learning to color, size change, normalization, and image enhancement on micrograph datasets with large sample differences. *Mater. Des.* **2023**, *232*, 112086. [CrossRef]
39. Rosenblatt, F. Principles of neurodynamics. In *Perceptrons and the Theory of Brain Mechanisms*; Cornell Aeronautical Laboratory Inc.: Buffalo, NY, USA, 1961.
40. Huang, C.-J.; Kuo, P.-H. A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in smart cities. *Sensors* **2018**, *18*, 2220. [CrossRef] [PubMed]
41. Tian, C.; Ma, J.; Zhang, C.; Zhan, P. A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network. *Energies* **2018**, *11*, 3493. [CrossRef]
42. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm.* **2021**, *173*, 24–49. [CrossRef]
43. Dunea, D.; Pohoata, A.; Iordache, S. Using wavelet–feedforward neural networks to improve air pollution forecasting in urban environments. *Environ. Monit. Assess.* **2015**, *187*, 477. [CrossRef]
44. Lara-Benítez, P.; Carranza-García, M.; Luna-Romera, J.M.; Riquelme, J.C. Temporal convolutional networks applied to energy-related time series forecasting. *Appl. Sci.* **2020**, *10*, 2322. [CrossRef]
45. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
46. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
47. Pascanu, R. On the difficulty of training recurrent neural networks. *arXiv* **2013**, arXiv:1211.5063.
48. Bi, J.; Zhang, X.; Yuan, H.; Zhang, J.; Zhou, M. A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 1869–1879. [CrossRef]
49. Liu, M.; Sun, X.; Wang, Q.; Deng, S. Short-term load forecasting using EMD with feature selection and TCN-based deep learning model. *Energies* **2022**, *15*, 7170. [CrossRef]
50. Li, D.; Homma, Y.; Nakamura, A.; Shimizu, Y.; Honda, F.; Yamamura, T.; Aoki, D. Ferromagnetic cluster glass behavior and large magnetocaloric effect in Ho2 PtSi3. *Solid State Commun.* **2017**, *268*, 1–6. [CrossRef]
51. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.