The object of the study is CO_2 emission prediction using deep learning models. The problem lies in developing accurate models capable of handling temporal dependencies and periodic patterns in CO_2 data. To address this, three deep learning models – temporal convolutional network (TCN), long short-term memory (LSTM), and a hybrid TCN-LSTM are evaluated. These models are optimized using random search and Bayesian optimization. Results indicate that the Hybrid TCN-LSTM model, optimized via random search, performs best, achieving MAE: 1.0269, R^2: 0.9305, and MAPE: 4.47%. TCN excels at capturing periodic patterns through dilated convolutions, while LSTM handles long-term dependencies. Their integration combines these strengths, improving accuracy. Optimal hyperparameters (learning rate: 0.000539, dropout rate: 0.5) enhance robustness. Random search outperforms Bayesian optimization in navigating complex search spaces and avoiding local optima. Key findings include the hybrid model's ability to address short-term periodicity and long-term trends, and Random Search's reliability over Bayesian methods in this context. These insights advance time series forecasting methodologies and support robust predictive frameworks. Practically, they aid environmental policy, energy planning, and carbon trading by enabling data-driven decisions for emission reduction. However, implementation requires high-quality historical data and sufficient computational resources

Keywords: CO_2 prediction, deep learning, random search, Bayesian optimization, hyperparameter, accuracy

UDC 504:004.8

DOI: 10.15587/1729-4061.2025.331523

EVALUATING DEEP LEARNING ARCHITECTURES FOR CO_2 EMISSIONS FORECASTING: TCN, LSTM, AND HYBRID APPROACHES WITH HYPERPARAMETER OPTIMIZATION

Roni Yunis
Doctoral Student of Computer Science*,
Lecture of Information Systems
Department of Information Systems
Universitas Mikroskil
Thamrin str., 112, Medan, Indonesia, 20212

Tengku Henny Febriana Harumy
Corresponding author
Doctor of Computer Science*
E-mail: hennyharumy@usu.ac.id

Syahril Efendi
Doctor of Mathematics, Professor*
*Department of Computer Science
Universitas Sumatera Utara
Dr. T. Mansur str., 9, Sumatera Utara, Indonesia, 20155

Received 03.06.2025
Received in revised form 25.08.2025
Accepted 08.09.2025
Published 28.10.2025

How to Cite: *Yunis, R., Harumy, T. H. F., Efendi, S. (2025). Evaluating deep learning architectures for CO_2 emissions forecasting: TCN, LSTM, and hybrid approaches with hyperparameter optimization. Eastern-European Journal of Enterprise Technologies, 5 (10 (137)), 20–32. https://doi.org/10.15587/1729-4061.2025.331523*

1. Introduction

In the current global context, where nations are committing to ambitious carbon neutrality targets, the ability to generate reliable CO_2 emission forecasts is not only a scientific challenge but also a policy imperative. Accurate predictions of CO_2 emissions are vital for effective climate change mitigation and informed policy design. As global emissions continue to rise, advanced forecasting systems are increasingly needed.

Both the Intergovernmental Panel on Climate Change (IPCC) and the International Energy Agency (IEA) emphasize that the accuracy of emission forecasts is crucial to the success of net-zero strategies [1]. However, traditional statistical approaches often fail to capture the nonlinear and dynamic nature of emission trends, highlighting the need for more sophisticated predictive tools [2].

Deep learning models have emerged as powerful methods to address this need. Specifically, temporal convolutional networks (TCNs) excel in parallel computation and modeling long-term dependencies through dilated convolutions, making them efficient for handling structured, periodic emission data [3]. Conversely, long short-term memory (LSTM) networks utilize recurrent memory to capture complex temporal sequences and adapt well to irregular patterns [3].

A hybrid architecture that leverages the complementary strengths of TCN (for periodic patterns) and LSTM (for long-term dependencies) offers a promising pathway to a more comprehensive predictive capability for CO_2 emissions [4]. Furthermore, integrating hyperparameter optimization (HPO) techniques is critical to enhance the predictive accuracy and computational efficiency of these models [5].

The development of such optimized deep learning frameworks is therefore a key scientific priority. These advances highlight the potential of modern machine learning approaches to significantly improve the reliability of CO_2 forecasting, which is increasingly vital for supporting climate action and net-zero transition strategies.

2. Literature review and problem statement

In recent years, machine learning and deep learning models have gained increasing prominence in climate change mitigation research, particularly in the prediction of CO_2 emissions. Among these methods, temporal convolutional networks (TCNs) are generally favored for structured datasets due to their computational efficiency, while long short-term memory (LSTM) networks are better suited for datasets with irregular or complex temporal patterns. Nevertheless, domain-specific challenges such as emission variability across industrial, transportation, and energy sectors continue to complicate model selection. To address these limitations, recent studies have proposed hybrid architectures that combine the strengths of TCNs and LSTMs, with attention mechanisms introduced to emphasize significant temporal dependencies [6].

In parallel, systematic hyperparameter tuning such as adjustments to sequence length, dropout rate, and dilation factors has proven essential for enhancing forecasting performance, particularly in TCN-based models [7]. However, despite these advances, few studies have provided a comprehensive comparative evaluation of TCN and LSTM models for CO_2 emissions forecasting, especially when integrated with systematic hyperparameter optimization. This gap highlights the need for further research to establish robust benchmarks and identify the most effective modeling strategies in this domain.

A substantial research gap remains in comprehensively evaluating TCN, LSTM, and hybrid architectures across diverse CO_2 emission contexts, particularly when incorporating detailed HPO strategies [8, 9]. In response to these challenges, research into deep learning-based emission forecasting continues to attract attention, offering the potential to refine data-driven climate strategies, energy planning, and carbon market mechanisms. The paper [10] presents the results of research on the application of LSTM for time series forecasting in the industrial sector, demonstrating that LSTM models outperform traditional statistical approaches such as ARIMA in terms of predictive accuracy. This study builds a 2020 emission inventory for China's cement industry and, using an LSTM model, projects 2025–2035 emissions, showing that fuel and clinker substitution best reduce SO_2 and CO_2, while end-of-pipe controls are most effective for underscoring the co-benefits of low-carbon technologies

Shown in [11], the application of TCNs in large-scale time series datasets particularly in energy forecasting achieves superior computational efficiency compared to LSTM, making TCNs particularly valuable for structured emissions datasets. The study shows that TCN outperforms LSTM in seasonal energy forecasting by improving accuracy and efficiency. Despite these promising results, there remain unresolved issues related to the direct comparison between these models, especially in emissions forecasting. The reason for this may be objective difficulties associated with dataset heterogeneity, varying temporal structures, and a lack of standardized benchmarks for model comparison. Additionally, fundamental challenges in capturing long-term dependencies and short-term periodicity simultaneously still limit the practical application of individual models.

Hyperparameter optimization (HPO) emerges as a critical method to enhance the accuracy and efficiency of deep learning models. Research [5] comparing optimization techniques for CNN and LSTM found that adaptive methods such as Optuna and Tree-structured Parzen Estimator (TPE) yield better performance than conventional approaches like grid search. However, this study focused primarily on architecture-level optimization without targeting univariate emissions time series.

Moreover, several studies, such as [12], emphasize that TCN architectures mitigate the vanishing gradient problem commonly faced by RNN-based models, including LSTM, due to their use of dilated convolutions. Furthermore, [13] introduces a hybrid TCN-LSTM model with Savitzky-Golay filtering for wind power forecasting, achieving higher robustness than standalone models. However, this work does not fully examine model behavior under diverse emissions contexts or compare optimization techniques across architectures.

Nevertheless, while TCNs show theoretical superiority, their empirical effectiveness for CO_2 prediction remains insufficiently explored. Study [9, 11] affirms the advantages of LSTM in handling long-term dependencies, while [7] focuses on the integration of TCN and LSTM into hybrid architectures to combine their respective strengths. Yet, these studies fall short of providing a comprehensive performance assessment of each model under identical conditions using real-world CO_2 data.

Additionally, while [6] underscores the importance of tuning parameters such as dilation factors and dropout rates in optimizing TCN performance, it does not address how these configurations affect LSTM or hybrid models. A systematic benchmarking of the impact of HPO strategies – particularly random search and Bayesian optimization – on each model type is still lacking. Although prior research has shown that hyperparameter tuning can significantly improve accuracy [14], its specific influence on emissions forecasting has not been rigorously analyzed.

A notable contribution in this domain is made by [15], which investigates hybrid models such as CNN-LSTM and TCN-BiLSTM in multivariate environmental forecasting. The study confirms their capacity to model spatiotemporal dependencies effectively, but it employs a grid search approach for HPO suboptimal in high-dimensional search spaces. Moreover, the models are not tested in univariate CO_2 emissions scenarios, nor are adaptive optimization methods like random or Bayesian search examined in detail.

These considerations indicate the need for a comprehensive investigation into the relative performance of TCN, LSTM, and hybrid TCN-LSTM models for univariate CO_2 emissions forecasting, particularly emphasizing the impact of adaptive hyperparameter optimization techniques. Such research could yield valuable insights into configuring predictive models to support more effective climate-related decision-making.

3. The aim and objectives of the study

This study aims to evaluate and compare the performance of temporal convolutional networks (TCN), long short-term memory (LSTM), and hybrid TCN-LSTM models in predicting CO_2 emissions, with the incorporation of hyperparameter optimization. By developing a robust benchmarking framework, this research seeks to provide insights into model selection, thereby supporting more accurate forecasting and effective climate change mitigation strategies.

To achieve this aim, the following objectives are accomplished:

– to design and implement deep learning architectures for CO_2 emissions prediction;

– to identify the most accurate and efficient model for predicting CO_2 emissions through an empirical comparison of TCN, LSTM, and Hybrid TCN-LSTM approaches;

– to identify the optimal combination of hyperparameters that can improve the accuracy of CO_2 emissions predictions.

4. Materials and methods

The object of the study is CO_2 emission prediction using deep learning models. Specifically, the research focuses on evaluating and comparing the performance of tree architectures: temporal convolutional network (TCN), long short-term memory (LSTM), and hybrid TCN-LSTM model. The study to determine the most accurate and efficient model for forecasting CO_2 emissions while incorporating hyperparameter optimization (HPO) techniques (random search and Bayesian optimization) to enhance predictive capabilities.

The central hypothesis of this study is the hybrid TCN-LSTM model, when optimized with hyperparameter tuning (specifically Random Search), will outperform standalone TCN and LSTM models in predicting CO_2 emissions by leveraging the complementary strengths of both architectures. The key supporting arguments for hypothesis are: TCNs excel at capturing long-term dependencies and periodic patterns via dilated convolutions. LSTMs are effective at modeling complex temporal dynamics through recurrent memory mechanisms. The hybrid model integrates the strengths, addressing limitations of individual models (e. g., TCN's fixed receptive field vs. LSTM's sequential processing). HPO (e. g., learning rate, dropout rates) further refines model performance.

The assumption made in the study are:

– model complementarity. TCN and LSTM architecture are assumed to have strengths (TCN for local patterns, LSTM for long-term dependencies) the improve prediction accuracy when combined;

– hyperparameter optimization effectiveness. Random search and Bayesian optimization are assumed to significantly enhance model performance compared to default hyperparameters. Random search is assumed to avoid local optima traps better than Bayesian optimization in complex search spaces;

– data quality and relevance. The dataset from Our World in Data is assumed to be representative of real-world CO_2 emission patterns and sufficient for training robust models;

– generalization of results: The optimized models are assumed to generalize well to unseen data, given the use of rigorous evaluation metrics (MAE, RMSE, R^2, MAPE) and a 70:30 train-test split.

For clarity, the following abbreviations are used in this section: TCN – temporal convolutional network; LSTM – long short-term memory; HPO – hyperparameter optimization; MAE – mean absolute error; RMSE – root mean squared error; R^2 – squared; MAPE – mean absolute percentage error.

In this study, the methodology outlined in Fig. 1 is designed to systematically develop and optimize an effective CO_2 emissions prediction model. Each phase is carefully structured to maximize prediction accuracy and computational efficiency:

1. Data collection and preprocessing. The dataset used in this study was obtained from the data published by Our World in Data [16]. The dataset spans the period from 1850 to 2022. After data cleaning and outlier removal using the Interquartile Range (IQR) method, it comprises 55 features and 47,415 instances. The next step is feature selection, in which random forest will be utilized as the estimator for selecting features. Recursive feature elimination (RFE) will be employed to systematically eliminate less important features based on their significance as determined by the random forest Regressor [17]. In this experiment, eight numerical features and one target variable were selected to enhance computational efficiency and prevent overfitting. This approach aims to balance model complexity with predictive accuracy. After feature selection, the data is divided into 70% for training and 30% for testing. This split ratio is commonly used in machine learning to balance training data and evaluation.

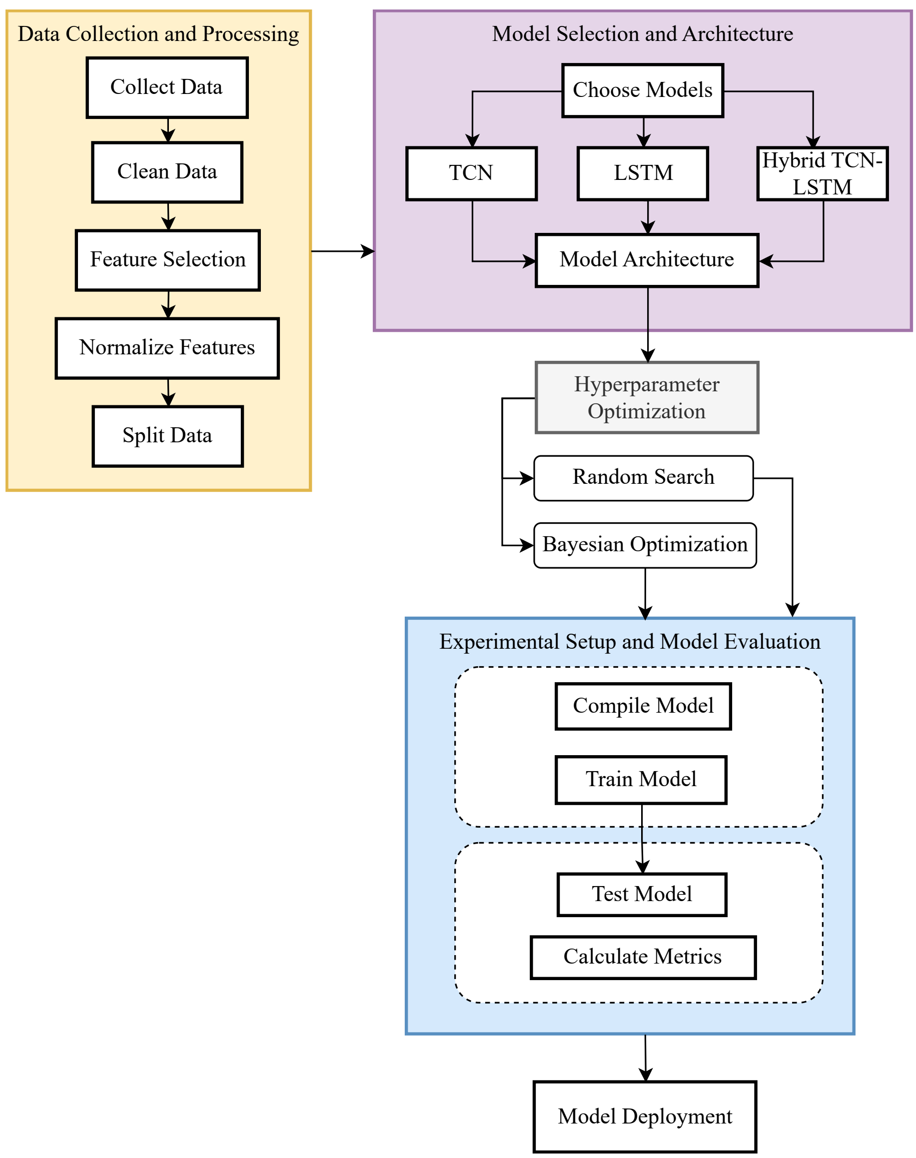

Fig. 1. Research method

2. Model selection and architecture. The deep learning models utilized in this study are described as follows:

2. 1. Temporal convolutional network (TCN) is a model based on convolutional neural networks (CNNs) that is specifically designed to process sequential or time series data [6, 17]. Unlike recurrent neural networks (RNNs), TCNs utilize one-dimensional convolutions along with dilated convolutions to effectively capture long-range temporal dependencies while addressing the vanishing gradient problem [9, 18]. Dilated convolution expands the filter's receptive field by introducing dilation, which refers to the spacing between filter elements. This technique enables the model to capture temporal information more effectively [19, 20].

For each TCN layer l with dilation d_l

$$H_{tcn}^{(l)} = Swish\left(W_{tcn}^{(l)} * d_l H_{tcn}^{(l-1)} + b_{tcn}^{(l)}\right), \tag{1}$$

where H is output of the l-th TCN layer, d_l is dilated convolution, and W, b is learnable weight and biases. Post processing are:

– batch normalization

$$\hat{H}_{tcn} = BatchNorm\left(H_{tcn}\right); \tag{2}$$

– flatten and dense layer

$$H_{dense} = Swish\left(W_{dense} \cdot Flatten\left(\hat{H}_{tcn}\right) + b_{dense}\right); \tag{3}$$

– output layer

$$\hat{y}_{tcn} = W_{out} \cdot H_{dense} + b_{out}. \tag{4}$$

The TCN architecture implemented in this study is presented in Table 1 below.

Table 1

Developing TCN architecture

Layer	Input Shape	Output Shape	Parameter
Input layer	(B, T, F)	(B, T, F)	T, F
TCN layer (4 Block)	(B, T, F)	(B, T, 128)	Kernel = 3
			Filters = 128
			Dilation = [1, 2, 4, 8]
			Activation = Swish
			Dropout = 25%
Batch normalization	(B, T, 128)	(B, T, 128)	–
Flatten	(B, T, 128)	(B, T, 128)	–
Dense layer	(B, T, 128)	(B, 64)	Units = 64
			Activation = Swish
			Dropout =2 0%
Output layer	(B, 64)	(B, 1)	Units = 1
			Activation = Linear

2. 2. Long short-term memory (LSTM) is a type of recurrent neural network (RNN) specifically designed to manage long-term dependencies in sequential data. It employs memory cells and gates to regulate the flow of information during the processing of sequences [21]. LSTM networks enable the capture of relationships between distant elements in a time sequence [22].

For each time step t are:

– input gate

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right); \tag{5}$$

– forget gate

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right); \tag{6}$$

– output gate

$$o_t = \sigma\left(w_o \cdot \left[h_{t-1}, x_t\right] + b_o\right); \tag{7}$$

– candidate cell gate

$$\tilde{c}_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right); \tag{8}$$

– cell state update

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t; \tag{9}$$

– hidden state

$$h_t = o_t \odot \tanh\left(c_t\right); \tag{10}$$

– output

$$\hat{y}_{lstm} = W_{out} . h_T + b_{out}, \tag{11}$$

where h_T is the final hidden state. The LSTM architecture utilized in this study is detailed in Table 2 below.

Table 2

Developing LSTM architecture

Layer	Input shape	Output shape	Parameter
Input layer	(B, T, F)	(B, T, F)	T, F
LSTM layer	(B, T, F)	(B, 96)	Units = 96
			Dropout = 20%
			Return sequences = False
Dense layer	(B, 96)	(B, 64)	Units = 64
			Activation = Relu
			Dropout = 20%
Output layer	(B, 64)	(B, 1)	Units = 1
			Activation = Linear

2. 3. The Hybrid TCN-LSTM model integrates the TCN efficiency in capturing local patterns and long-term dependencies with the LSTM network's capability to model complex temporal relationships [22, 23]. The formulation is:

$$H_{tcn} = TCN\left(X\right), \tag{12}$$

$$H_{lstm} = LSTM\left(X\right), \tag{13}$$

$$H_{lstm\_broadcast} = Repeat\left(H_{lstm}, T\right), \tag{14}$$

$$H_{concat} = Concat\left(\left[H_{tcn}, H_{lstm\_broadcast}\right]\right). \tag{15}$$

Dense layer and output:

$$H_{dense} = Swish\left(W_{dense} \cdot H_{concat} + b_{dense}\right), \tag{16}$$

$$\hat{y}_{hybrid} = W_{out} \cdot H_{dense} + b_{out}, \quad (17)$$

where H_{tcn} is TCN path, H_{lstm} is LSTM path, $H_{lstm\_broadcast}$ is broadcast LSTM output, and H_{concat} is concatenation. This formulation aligns with the architecture and parameters to be developed in this study, ensuring accurate replication of the model for CO_2 emission prediction. The hybrid TCN-LSTM architecture utilized in this study is detailed in Table 3 below.

Swish activation will be utilized to enhance the architecture. It is a non-linear function and a contemporary alternative to ReLU, often providing improved performance [24, 25]

$$f(x) = x \cdot \sigma(x), \quad (18)$$

where

$$\sigma(x) = \frac{1}{1+e^{-x}}, \quad (19)$$

is the sigmoid function. Swish is the product of the input (x) and the sigmoid function of that input.

3. Hyperparameter optimization (HPO). HPO aims to identify the optimal combination of hyperparameters for model training. This study employs two methods: Random search, which randomly selects hyperparameters from a specified range, and Bayesian optimization, which predicts the best combination based on previous results [14]. Bayesian optimization employs a probabilistic model, specifically a Gaussian process, to predict the optimal hyperparameter combinations based on prior results [5]. The research [26, 27] indicates the hyperparameter optimization, even without employing formal methods such as Bayesian optimization, still plays a crucial role in enhancing the performance of hybrid model. The hyperparameter configuration utilized in this study is presented in Table 4 below.

Table 4

Hyperparameter optimization (HPO)

Model	Hyperparameter	Search space
TCN	tcn_nb_filter	{64, 128, 192, 256}
	tcn_kernel_size	{3, 5}
	tcn_dropout_rate	{0.1, 0.2, 0,3, 0.4, 0.5}
	dilations_rate	{1, 2, 4, 8}
	dense_units	{32, 64, 96, 128}
	dense_l2_reg	{0.0001 – 0.01}
	learning_rate	{0.0001 – 0.001}
	activation_tcn	‘swish’
	activation_dense	‘relu’
	epoch	50 (tuning), 100(final training)
LSTM	lstm_units	{32, 64, 96, 128}
	dropout_rate	{0.1, 0.2, 0,3, 0.4, 0.5}
	dense_units	{32, 64, 96, 128}
	dense_l2_reg	{0.0001 – 0.01}
	learning_rate	{0.0001 – 0.001}
	activation_lstm	‘tanh’
	activation_dense	‘relu’
	epoch	50 (tuning), 100(final training)
Hybrid TCN-LSTM	tcn_nb_filters	{64, 128, 192, 256}
	tcn_kernel_size	{3, 5}
	tcn_dropout_rate	{0.1, 0.2, 0,3, 0.4, 0.5}
	dilations_rate	{1, 2, 4, 8}
	lstm_units	{32, 64, 96, 128}
	lstm_dropout_rate	{0.1, 0.2, 0,3, 0.4, 0.5}
	dense_units	{32, 64, 96, 128}
	dense_l2_reg	{0.0001 – 0.01}
	learning_rate	{0.0001 – 0.001}
	activation_tcn	‘swish’
	activation_lstm	‘tanh’
	activation_dense	‘relu’
	epoch	50 (tuning), 100(final training)

Table 3

Developing hybrid TCN-LSTM architecture

Layer	Input shape	Output shape	Parameter
Input layer	(B, T, F)	(B, T, F)	T, F
TCN layer	(B, T, F)	(B, T, 64)	Filters = 64
			Kernel = 5
			Dilation = [1, 2, 4, 8]
LSTM Layer	(B, T, F)	(B, 32)	Units = 32
			Dropout = 20%
Concatenation	(B, T.64), (B, 32)	(B, T.64 + 32)	–
Dense layer	(B, T.64 + 32)	(B, 32)	Units = 32
			Activation = Swish
			Dropout = 20%
Output layer	(B, 32)	(B, 1)	Units = 1
			Activation = Linear

Table 4 presents the configuration used to evaluate model performance. The TCN model includes parameters such as the number of filters, kernel size, dropout rate, and the number of units in the dense layer, while maintaining a fixed dilation rate and utilizing the ‘swish’ activation function. The LSTM model is characterized by its number of units and dropout rate, employing the ‘tanh’ activation function. The hybrid TCN-LSTM model integrates hyperparameters from both models. The hyperparameter search is conducted over 50 epochs, while the final training of the model is performed over 100 epochs.

4. Experimental setup and model evaluation. The experiment was conducted on a Windows operating system using Python, with the environment set up on Google Colab. Model training and testing was performed utilizing the A100 GPU runtime. The model was implemented using the Keras, TensorFlow, Keras-TCN, and Keras-Tuner libraries. Four evaluation metrics were employed in this study: MAE, RMSE, R^2, and MAPE. The formulations of these metrics are as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \quad (20)$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}, \quad (21)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \qquad (22)$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \qquad (23)$$

in the equation, y_i represents the actual CO_2 emission data, $yhat_i$ donates the predicted value, and n indicates the number of samples. This metric evaluates the model's performance and predictive accuracy.

5. Model development. This model is designed to predict CO_2 emissions using a deep learning-based time series approach, implemented through six systematic stages as outlined in the following algorithm. Following the preprocessing and feature selection steps described in the previous section, the normalized dataset is used to train and evaluate three models:

I) a temporal convolutional network (TCN) with a kernel size of 3, 128 filters, and dilation rates of (1, 2, 4, 8);

II) a Long short-term memory (LSTM) network with 96 units and 20% dropout;

III) a hybrid TCN-LSTM architecture, where a TCN layer (kernel size = 5, 64 filters) operates in parallel with an LSTM layer (32 units).

The outputs of both layers are concatenated and passed through a dense layer consisting of 32 units with a Swish activation function to yield the final output.

5. Results of evaluating deep learning models for CO_2 emissions prediction

5. 1. Proposed model architectures

Before introducing the deep learning architectures, it is crucial to outline the data preparation and preprocessing steps, including feature correlation analysis, historical CO_2 trend examination, and data partitioning. These steps ensure models are built on relevant features and long-term emission patterns. Fig. 2 presents the feature correlation heatmap, while Fig. 3 illustrates the historical trends of the CO_2 emissions over time.

This study develops a predictive model based on long-term structural drivers of CO_2 emissions (Fig. 3), such as energy use and economic growth, and does not explicitly consider short-term disruptions. A key limitation is the exclusion of the Covid-19 effect, which temporarily reduced global emissions by about 5–7% in 2020 due to reduced transport and industry. Since emissions rebounded to near pre-pandemic levels by 2021, the long-term trajectory remained largely unchanged. Thus, while the short-term impact of Covid-19 is acknowledged, the model emphasizes underlying trends aligned with Intergovernmental Panel on Climate Change (IPCC) projections. Using the data foundation, three architectures are proposed for predicting CO_2 emissions: TCN, LSTM, and a Hybrid TCN-LSTM model. As shown in Fig. 4, these models allow for a comparative evaluation of different deep learning strategies in capturing sequential dependencies and structural patterns in emissions data.

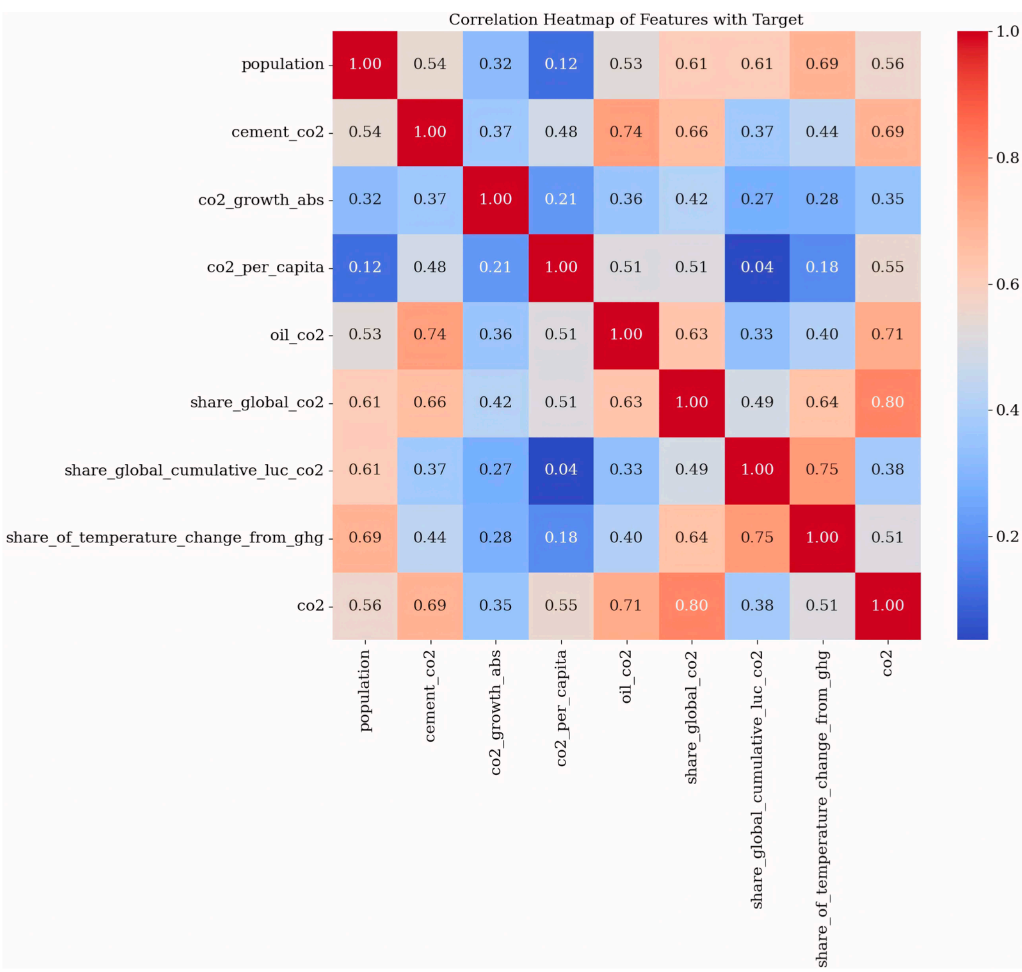

Fig. 2. Heatmap correlation

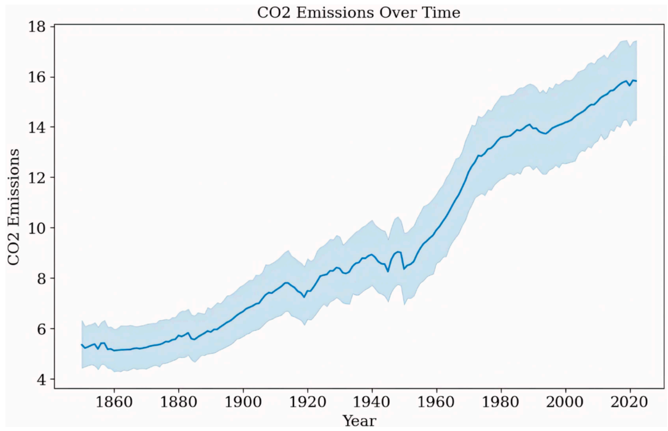

Fig. 3. Historical trends in CO_2 emission over time

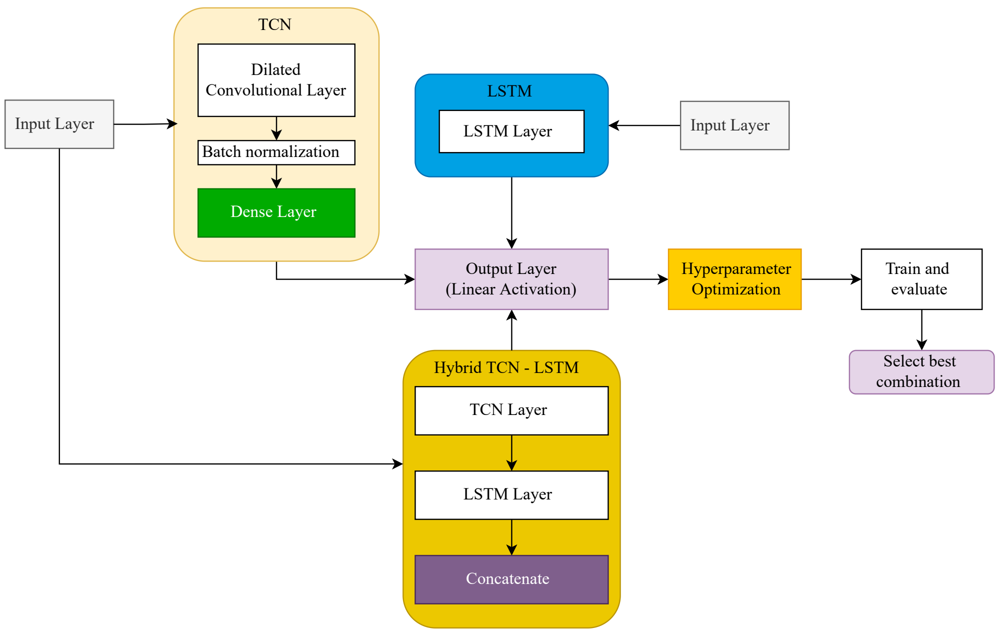

Fig. 4. Proposed model architectures for CO_2 emissions forecasting

Hyperparameter optimization is performed using both random search and Bayesian optimization, targeting the minimization of mean squared error (MSE) via the Adam optimizer, with learning rates ranging from 0.0001 to 0.001. The search space includes the number of TCN filters (64–256), LSTM units (32–128), and dropout rates (0.1–0.4). Each model is trained for 100 epochs with early stopping (patience = 15) and model checkpointing to prevent overfitting.

Model performance is evaluated on the testing set using four standard metrics: mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R^2), and mean absolute percentage error (MAPE). Among the tested architectures, the hybrid TCN-LSTM model demonstrates the highest predictive accuracy, benefiting from the ability to capture both long-term and short-term dependencies. The trained models are saved for future deployment in CO_2 emission forecasting tasks. The integration of hybrid modeling with systematic optimization contributes to the robustness and generalizability of the proposed approach.

Algorithm: CO_2 Emissions Prediction
Input: Time series dataset D
Output: Optimized model for time series prediction

#Step 1: Data Proprocessing
Load dataset D
Handle missing value:
Remove columns with more than 50% missing value
Drop columns with more than 90% missing value
Remove outlier with IQR method
Feature selection using Random Forest Regressor and RFE
Normalize the dataset using MinMaxScaler
Split the dataset into training and testing sets: (Xtrain, ytrain), (Xtest, ytest): 70:30
#Step 2: Define Model Architecture
#Model TCN
Define Input layer with shape (T, F) (T, F), where T is the time steps and F is the number of features
Add TCN Layer:
Dilated Convolution with kernel size=3, 128 filters, and dilation rates= [1,2,4,8]
Swish activation and Dropout (25%)
Apply Batch Normalization
Add Dense Layer (64 units, Swish activation, Dropout 20%)
Define Output Layer (1unit, linear activation)
#Model LSTM
Define Input Layer

```
Add LSTM Layer:
        LSTM units= 96
        Dropout rate= 0.2
Add Dense Layer (64 units, Relu activation, Dropout 20%)
Define Output Layer (1 units, linear activation)
#Model Hybrid TCN-LSTM
Define Input Layer with shape (T, F) (T, F)
Apply TCN Layer:
        Filters= 64, kernel size= 5, dilation rates= [1,2,4,8]
Apply LSTM Layer:
        LSTM units= 32, Dropout rate= 0.2
Concatenate TCN and LSTM outputs
Add Dense Layer (32 units, Swish activation, Dropout 20%)
Define Ouput Layer (1 units, linear activation)
Step 3: Hyperparameter Optimization (HPO)
Define Search Space
        TCN filters: {64, 128, 192, 256}
        Kernel size: {3, 5}
        Dropout rate: {0.1, 0.2, 0.3, 0.4}
        LSTM units: {32, 64, 96, 128}
        Dense units: {32. 64, 96, 128}
        Dense L2 reg: {0.0001 – 0.01}
        Learning rate: {0.0001 – 0.001}
Apply Optimization Method
Random Search
        Randomly select hyperparameters
        Train the model and evaluate MAE and RMSE
        Select the best hyperparameter combination
        Epoch= 50
Bayesian Optimization
        Define objective function based on loss function (MSE)
        Use Gaussian Process to estimate the best hyperparameter combination
        Train the model iteratively, refining search
        Epoch= 50
Step 4: Model Compilation and Training
Compile the model using Adam optimizer with learning rate optimized from HPO
Use Mean Squared Error (MSE) as the loss function
Use Mean Absolute Error (MAE) as the evaluation metric
Train the model using
        Earnly Stopping (patience= 15)
        Model Checkpointing (save best model)
        Epoch= 100
Step 5: Model Evaluation
Test model on testing set: ytest = model (Xtest)
Calculate evaluation metrics:
        MAE
        RMSE
        R2
        MAPE
Step 6: Model Deployment
Save the best-performing model.
```

5. 2. Evaluation of model performance in CO_2 emission prediction

The performance evaluation indicates that the Hybrid TCN-LSTM model, optimized using Random Search, outperformed both the standalone TCN and LSTM models. As shown in Table 5, it achieved the lowest MAE (1.0269), RMSE (2.9392), and MAPE (4.4725), along with the highest R^2 score of 0.9305. These results demonstrate superior accuracy and an enhanced ability to capture the variability of CO_2 emissions.

The exceptional performance can be attributed to the complementary strengths of the TCN and LSTM architectures: TCN effectively captures long-term dependencies through dilated convolutions, while LSTM adeptly manages complex and dynamic temporal patterns. These findings underscore the hybrid model as the most accurate and efficient for predicting CO_2 emissions, providing both high accuracy and adaptability to fluctuations in the data. Nevertheless, the individually optimized TCN and LSTM models also exhibited competitive performance.

Table 5

Model evaluation matrix

Model	HPO	MAE	RMSE	R^2	MAPE
TCN baseline	–	1.1776	3.0966	0.9228	6.4031
LSTM baseline	–	1.3650	3.3398	0.9102	5.5878
TCN	Random search	1.0923	2.9765	0.9287	4.5854
TCN	Bayesian optimization	1.2268	3.0329	0.9260	5.4411
LSTM	Random search	1.3331	3.1920	0.9180	6.4702
LSTM	Bayesian optimization	1.2346	3.0961	0.9228	5.5534
Hybrid TCN-LSTM	Random search	1.0269	2.9392	0.9305	4.4725
Hybrid TCN-LSTM	Bayesian optimization	1.0723	2.9657	0.9292	5.1870

For instance, the TCN model optimized using random search reduced its MAE from 1.1776 (baseline) to 1.0923, while the LSTM model optimized with Bayesian Optimization decreased its MAE from 1.3650 to 1.2346. However, both models still lagged behind the hybrid model in terms of prediction accuracy. This underscores the significant potential of the hybrid approach for large-scale CO_2 emissions forecasting.

5. 3. The influence of hyperparameter optimization

Hyperparameter optimization is crucial for enhancing the performance of deep learning models, as demonstrated in this study. All models, including TCN, LSTM, and Hybrid, exhibited significant improvements after tuning compared to their baseline versions. These results underscore that optimal hyperparameter settings substantially enhance CO_2 prediction accuracy, particularly in complex time series scenarios such as emission data. One of the key findings from the hyperparameter optimization (Table 6) process is that certain parameters, such as the learning rate, dropout rate, and L2 regularization for dense layers, significantly impact model performance.

For instance, in the hybrid TCN-LSTM model optimized through random search, higher dropout rates (tcn_dropout_rate = 0.1 and lstm_dropout_rate = 0.5) enhanced the model's robustness against noise in the dataset. Additionally, utilizing an optimal learning rate, such as 0.000539539 in the hybrid model, facilitated more efficient learning. These results emphasize that the careful selection of hyperparameters can greatly improve model generalization, thereby producing more reliable predictive models. Performance validation, as illustrated in Fig. 5, demonstrates that the comparison between training loss and validation loss confirms the model's ability to learn effectively without overfitting, thereby maintaining stability and accuracy.

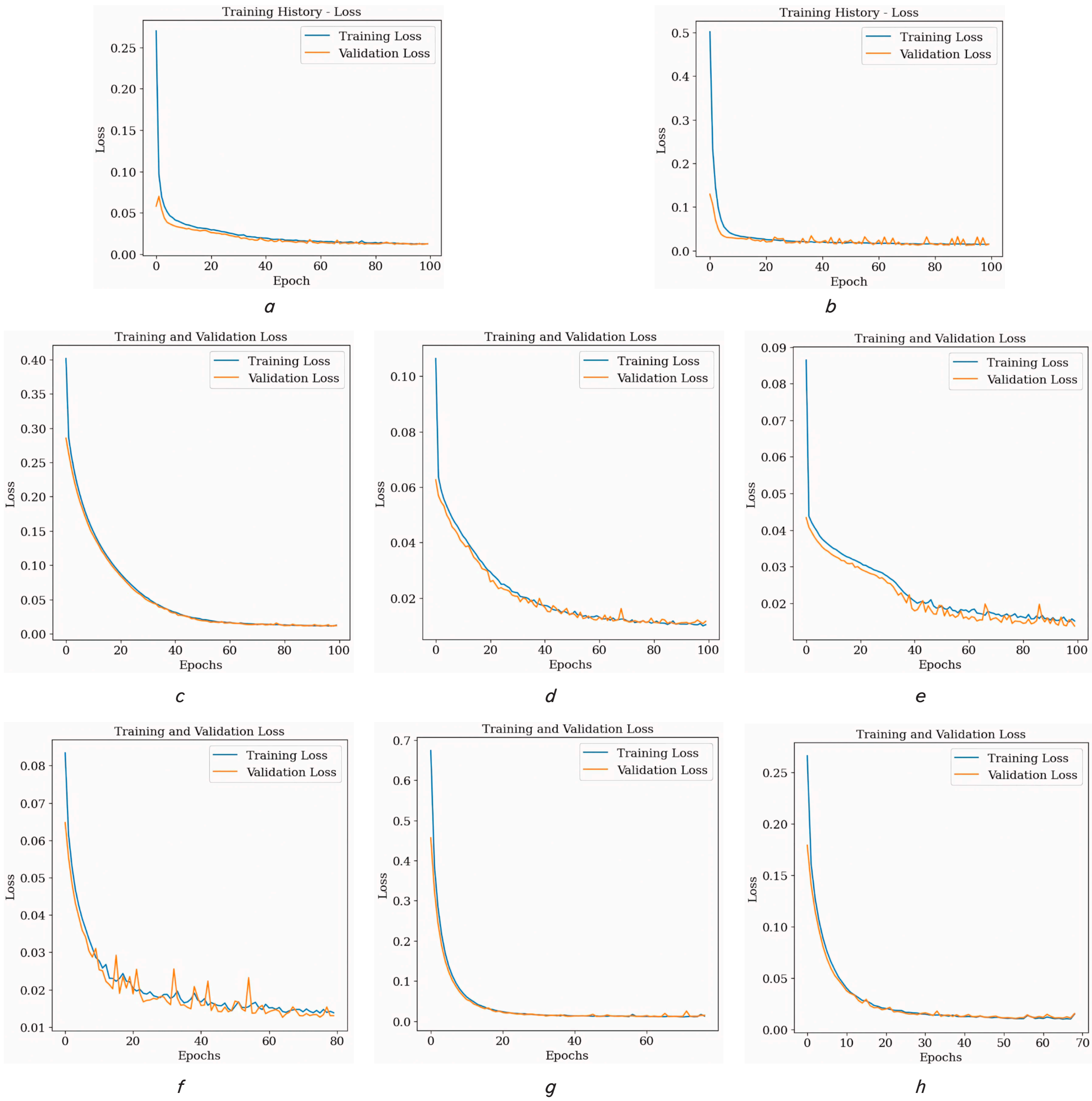

Fig. 5. Performance validation result: *a* – temporal convolutional network baseline; *b* – long short-term memory baseline; *c* – temporal convolutional network random search; *d* – temporal convolutional network Bayesian optimization; *e* – long short-term memory random search; *f* – long short-term memory Bayesian optimization; *g* – hybrid random search; *h* – hybrid Bayesian optimization

Table 6

Hyperparameter optimization performance

Model	HPO	Best Val MAE	Total elapsed time	Best hyperparameter
1	2	3	4	5
TCN baseline	–	0.0804	00h 11m 02s	–
LSTM baseline	–	0.0509	00h 13m 12s	–
TCN	Random search	0.0475	00h 07m 39s	tcn_nb_filters = 128
				tcn_kernel_size = 3
				tcn_dropout_rate = 0.2
				dense_units = 96
				dense_l2_reg = 0.004823781
				learning_rate = 0.000491589

Continuation of Table 6

1	2	3	4	5
TCN	Bayesian optimization	0.0692	00h 07m 38s	tcn_nb_filters = 128
				tcn_kernel_size = 3
				tcn_dropout_rate = 0.2
				dense_units = 32
				dense_l2_reg = 0.000856437
				learning_rate = 0.000159418
LSTM	Random search	0.0465	00h 05m 43s	lstm_units = 96
				dropout_rate = 0.4
				dense_units = 32
				dense_l2_reg = 0.000183332
				learning_rate = 0.000869153
LSTM	Bayesian optimization	0.0483	00h 05m 36s	lstm_units = 128
				dropout_rate = 0.1
				dense_units = 128
				dense_l2_reg = 0.000609870
				learning_rate = 0.000549713
Hybrid TCN-LSTM	Random search	0.0438	00h 13m 23s	tcn_nb_filters = 192
				tcn_kernel_size = 3
				tcn_dropout_rate = 0.1
				lstm_units = 64
				lstm_dropout_rate = 0.5
				dense_units = 128
				dense_l2_reg = 0.003467461
				learning_rate = 0.000539539
Hybrid TCN-LSTM	Bayesian optimization	0.0455	00h 13m 35s	tcn_nb_filters = 192
				tcn_kernel_size = 5
				tcn_dropout_rate = 0.3
				lstm_units = 128
				lstm_dropout_rate = 0.4
				dense_units = 32
				dense_l2_reg = 0.000133802
				learning_rate = 0.000533017

More specifically, the Fig. 5 presents eight training-validation loss trajectories across different model configurations namely TCN, LSTM, and a hybrid TCN-LSTM under two distinct hyperparameter optimization strategies: random search and Bayesian optimization. Models in Fig. 5, *a*, *b*, *g*, *h*, exhibit stable and effective training performance. In contrast, Fig. 5, *e*, *f*, indicate potential overfitting or high sensitivity to validation data. Meanwhile, Fig. 5, *c*, *d*, demonstrate consistent learning with slower convergence rates.

Thus, it can be concluded that the hybrid TCN-LSTM approach, which incorporates adaptive hyperparameter optimization primarily through random search yields the most stable and accurate CO_2 prediction results compared to either a single model or the lack of optimization.

6. Discussion of results evaluating deep learning models for CO_2 emissions prediction

This study demonstrates that systematic data preprocessing combined with well-designed deep learning architectures (Fig. 4) enhances the reliability of CO_2 emission forecasts. By emphasizing long-term structural drivers such as energy use (Fig. 3), the models align with IPCC projections while acknowledging, but not modeling, short-term anomalies like Covid-19. Comparative results show that the hybrid TCN-LSTM consistently outperforms single models by capturing both short- and long-term dependencies. Moreover, hyperparameter tuning through random search and Bayesian optimization improves accuracy and prevents overfitting, making the hybrid model robust, adaptable for real-time applications, and highly relevant for supporting data-driven climate mitigation and net-zero strategies.

The hybrid TCN-LSTM model delivers more accurate predictions and effectively manages data complexity. However, training time is a critical factor. As shown in Table 6, the hybrid model requires a longer training duration (13 minutes and 23 seconds) compared to individual models like TCN or LSTM (under 8 minutes). Nevertheless, the significant improvement in accuracy justifies the extended training time as a reasonable trade-off. This highlights the balance between accuracy and computational efficiency when selecting deep learning models for CO_2 emission prediction. Evaluation results confirm that the Hybrid TCN-LSTM model is most suitable for datasets characterized by complex and dynamic temporal patterns.

The choice of hyperparameter optimization methods significantly impacts model performance. Table 5 demonstrates that Random Search is more effective for hybrid models, achieving the lowest MAE (1.0269) due to its comprehensive exploration of the search space. The optimal hyperparameter configuration (e. g., tcn_dropout_rate = 0.1, and learning_rate = 0.000539) in Table 6 was attained using Random Search with best val MAE 0.0438. These findings underscore that hyperparameter optimization not only enhances accuracy but also ensures the model's practical applicability, such as in predicting CO_2 emissions for climate change mitigation. The decision between Random Search and Bayesian Optimization should align with the specific characteristics of the model and dataset. Unlike Grid Search as used in [15], which is constrained by its discrete and exhaustive nature, Random Search offers greater flexibility and robustness in high-dimensional, non-context search spaces, reducing the risk of local optima and improving convergence on optimal configurations.

This study demonstrates superior CO_2 emission prediction accuracy compared to prior research. The hybrid TCN-LSTM model optimized with Random Search achieves a MAPE of 4.47% and an R^2 of 0.9305, outperforming [19] (waterlogging depth prediction, MAE 2.22) and [13] (wind power prediction, 2.00% reduction in nMAE). This advantage stems from integrating TCN (capturing periodic patterns via dilated convolutions) and LSTM (modeling complex temporal dependencies), further enhanced by systematic hyperparameter tuning. Unlike [13], which relied on external filters for data denoising, this study proves that hyperparameter optimization alone suffices for stable CO_2 data. In contrast to [15], which applied hybrid models and conventional grid search to structured multivariate datasets, the present study addresses the more volatile and irregular nature of univariate CO_2 emissions using an adaptive, fine-grained hybrid TCN-LSTM architecture, resulting in superior predictive precision, broader generalization capacity, and enhanced relevance to domain-specific forecasting challenges.

This research addresses a literature gap by directly comparing TCN, LSTM, and Hybrid TCN-LSTM models for CO_2 prediction a rarely explored focus in prior work. Unlike [13, 19], which employed simple or parallel hybrid architectures, this study examines TCN-LSTM feature fusion through concatenation and evaluates the efficacy of Random Search versus Bayesian optimization. Results reveal that Random Search effectively navigates complex hyperparameter spaces to avoid local optima. This methodological rigor positions the study as a reference for developing adaptive, data-driven predictive models. In this way, the proposed framework not only enhances model accuracy, but also advances the methodological landscape of environmental forecasting by integrating optimization strategies directly into the architectural design.

The findings support the design of effective environmental policies through precise and stable CO_2 emission predictions. Despite requiring longer training times (13.38 minutes), the hybrid model is recommended for large-scale implementation to advance climate change mitigation. The proven accuracy and methodological robustness highlight the necessity of compromising computational efficiency for superior results. Thus, this study not only offers technical solutions but also provides actionable recommendations to inform data-driven policies in global efforts to reduce carbon emissions.

Accurate CO_2 emission predictions are essential for data-driven climate mitigation. This study optimizes TCN, LSTM, and a Hybrid TCN-LSTM model to enhance both accuracy and efficiency when working with large datasets. Through empirical analysis, it provides methodological insights for improving emission prediction systems and informs environmental policy. The architecture and optimization paradigm proposed herein are adaptable across domains, and serve as a scalable foundation for predictive modeling in support of sustainability goals.

This study is limited by its focus on univariate CO_2 emissions data, which restricts the generalizability of the models to more complex, multivariate environmental datasets. Additionally, the reproducibility of results may be affected by inherent randomness in model training and optimization, such as initialization variance and stochastic search processes. The stability of the proposed models under shifting data distributions or extreme events remains untested, highlighting a need for broader validation across varied emission scenarios.

A notable disadvantage lies in the high computational cost of the hybrid TCN-LSTM model, which requires significantly longer training time than its individual counterparts. While this is justified by improved accuracy, it poses challenges for deployment in time-sensitive or resource-constrained applications. The proven effectiveness of the hybrid TCN-LSTM model makes it a reliable tool for forecasting CO_2 emissions, enabling policymakers to design proactive, data-driven mitigation strategies. Furthermore, the success of random search offers an efficient and replicable method for tuning complex models in other domains like energy demand forecasting. By providing a scalable framework for large datasets, this study contributes to more informed decision-making and robust climate governance.

Future research should advance from univariate to multivariate CO_2 forecasting by incorporating sectoral and regional variables. Building on the effectiveness of random search, further exploration of advanced optimization methods such as evolutionary algorithms and reinforcement learning is recommended. Integrating explainable AI will improve transparency and policy relevance. Key challenges include increased model complexity, overfitting risks, and data harmonization, all of which must be addressed to ensure robust, scalable, and interpretable climate prediction systems.

7. Conclusion

1. This study introduces three architectures for CO_2 emission forecasting: TCN, LSTM, and hybrid TCN-LSTM. The TCN uses dilated convolutions and batch normalization to capture periodic patterns, while the LSTM models long-term dependencies through recurrent memory. The hybrid model integrates both strengths via concatenation, addressing short-term periodicity and long-term trends simultaneously. To enhance robustness and predictive accuracy, hyperparameter optimization using random search and Bayesian optimization is embedded into the architecture design, refining key parameters such as filter size, kernel size, dropout rate, and learning rate. This ensures that the proposed models are not only methodologically rigorous but also practical for real-world applications.

2. This study confirms that the hybrid TCN-LSTM model, optimized using random search, is the most effective architecture for predicting CO_2 emissions. It achieves the lowest mean absolute error (MAE) of 1.0269, root mean square error (RMSE) of 2.9392, mean absolute percentage error (MAPE) of 4.47%, and the highest coefficient of determination (R^2) of 0.9305. The model's strength lies in its ability to integrate TCN for capturing local temporal patterns with LSTM networks for modeling long-term dependencies. This hybrid model outperforms standalone TCN

and LSTM architectures and scales effectively to large datasets, although it necessitates a longer training time. These findings underscore the effectiveness of deep learning hybridization for CO_2 emission prediction and support data-driven strategies for climate change mitigation.

3. This study demonstrates that random search is more effective than Bayesian Optimization for tuning the Hybrid TCN-LSTM model, particularly in complex, non-linear search spaces. The optimal configuration (tcn_nb_filters = 192, learning_rate = 0.000539, tcn_dropout_rate = 0.1, lstm_dropout_rate = 0.5) minimizes prediction errors (MAE = 1.0269) and enhances model generalization. Random search excels in broad exploration, effectively avoiding traps associated with local minima. These findings advocate for the use of random search in hybrid models and other complex architectures for CO_2 emission prediction, thereby improving accuracy while providing an efficient and replicable framework.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, whether financial, personal, authorship or otherwise, that could affect the study and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Manuscript has associated data in a data repository.

Use of artificial intelligence

The authors have used artificial intelligence technologies within acceptable limits to provide their own verified data, to support language refinement, and which is described in the research methodology section.

Acknowledgments

We want to express our deepest gratitude to everyone who contributed to the completion of this research. First and foremost, we would like to thank Universitas Mikroskil, and Universitas Sumatera Utara for their financial and logistical support.

References

1. DeAngelo, J., Azevedo, I., Bistline, J., Clarke, L., Luderer, G., Byers, E., Davis, S. J. (2021). Energy systems in scenarios at net-zero CO_2 emissions. Nature Communications, 12 (1). https://doi.org/10.1038/s41467-021-26356-y
2. Zhao, Y., Liu, L., Wang, A., Liu, M. (2023). A novel deep learning based forecasting model for carbon emissions trading: A comparative analysis of regional markets. Solar Energy, 262, 111863. https://doi.org/10.1016/j.solener.2023.111863
3. Wei, X., Xu, Y. (2023). Research on carbon emission prediction and economic policy based on TCN-LSTM combined with attention mechanism. Frontiers in Ecology and Evolution, 11. https://doi.org/10.3389/fevo.2023.1270248
4. Chen, C., Guo, J., Zhang, L., Wu, X., Yang, Z. (2024). Robust multi-scale time series prediction for building carbon emissions with explainable deep learning. Energy and Buildings, 312, 114159. https://doi.org/10.1016/j.enbuild.2024.114159
5. Hanifi, S., Cammarono, A., Zare-Behtash, H. (2024). Advanced hyperparameter optimization of deep learning models for wind power prediction. Renewable Energy, 221, 119700. https://doi.org/10.1016/j.renene.2023.119700
6. Yin, H., Yin, Y., Li, H., Zhu, J., Xian, Z., Tang, Y. et al. (2025). Carbon emissions trading price forecasting based on temporal-spatial multidimensional collaborative attention network and segment imbalance regression. Applied Energy, 377, 124357. https://doi.org/10.1016/j.apenergy.2024.124357
7. Gong, M., Zhang, Y., Li, J., Chen, L. (2024). Dynamic spatial–temporal model for carbon emission forecasting. Journal of Cleaner Production, 463, 142581. https://doi.org/10.1016/j.jclepro.2024.142581
8. Chen, J., Cui, Y., Zhang, X., Yang, J., Zhou, M. (2024). Temporal Convolutional Network for Carbon Tax Projection: A Data-Driven Approach. Applied Sciences, 14 (20), 9213. https://doi.org/10.3390/app14209213
9. Rafi, M. A., Rodrigues, G. N., Mir, M. N. H., Bhuiyan, M. S. M., Mridha, M. F., Islam, M. R., Watanobe, Y. (2025). A Hybrid Temporal Convolutional Network and Transformer Model for Accurate and Scalable Sales Forecasting. IEEE Open Journal of the Computer Society, 6, 380–391. https://doi.org/10.1109/ojcs.2025.3538579
10. Liu, X., Yang, L., Du, J., Zhang, H., Hu, J., Chen, A., Lv, W. (2024). Carbon and air pollutant emissions forecast of China's cement industry from 2021 to 2035. Resources, Conservation and Recycling, 204, 107498. https://doi.org/10.1016/j.resconrec.2024.107498
11. Shaikh, A. K., Nazir, A., Khalique, N., Shah, A. S., Adhikari, N. (2023). A new approach to seasonal energy consumption forecasting using temporal convolutional networks. Results in Engineering, 19, 101296. https://doi.org/10.1016/j.rineng.2023.101296
12. Li, Q., Ren, X., Zhang, F., Gao, L., Hao, B. (2024). A novel ultra-short-term wind power forecasting method based on TCN and Informer models. Computers and Electrical Engineering, 120, 109632. https://doi.org/10.1016/j.compeleceng.2024.109632
13. Liu, S., Xu, T., Du, X., Zhang, Y., Wu, J. (2024). A hybrid deep learning model based on parallel architecture TCN-LSTM with Savitzky-Golay filter for wind power prediction. Energy Conversion and Management, 302, 118122. https://doi.org/10.1016/j.enconman.2024.118122
14. Tahir, M. F., Yousaf, M. Z., Tzes, A., El Moursi, M. S., El-Fouly, T. H. M. (2024). Enhanced solar photovoltaic power prediction using diverse machine learning algorithms with hyperparameter optimization. Renewable and Sustainable Energy Reviews, 200, 114581. https://doi.org/10.1016/j.rser.2024.114581

15. Mahmoud, A., Mohammed, A. (2024). Leveraging Hybrid Deep Learning Models for Enhanced Multivariate Time Series Forecasting. Neural Processing Letters, 56 (5). https://doi.org/10.1007/s11063-024-11656-3
16. Ritchie, H., Rosado, P., Roser, M. (2023). CO2 and Greenhouse Gas Emissions. Our World in Data. https://ourworldindata.org/co2-and-greenhouse-gas-emissions
17. Barzani, A. R., Pahlavani, P., Ghorbanzadeh, O., Gholamnia, K., Ghamisi, P. (2024). Evaluating the Impact of Recursive Feature Elimination on Machine Learning Models for Predicting Forest Fire-Prone Zones. Fire, 7 (12), 440. https://doi.org/10.3390/fire7120440
18. Wang, M., Qin, F. (2024). A TCN-Linear Hybrid Model for Chaotic Time Series Forecasting. Entropy, 26 (6), 467. https://doi.org/10.3390/e26060467
19. Yao, J., Cai, Z., Qian, Z., Yang, B. (2023). A noval approach based on TCN-LSTM network for predicting waterlogging depth with waterlogging monitoring station. PLOS ONE, 18 (10), e0286821. https://doi.org/10.1371/journal.pone.0286821
20. Dong, J., Liu, X., Su, R., Xu, H., Yu, T. (2025). TCN-Transformer Deep Network with Random Forest for Prediction of the Chemical Synthetic Ammonia Process. ACS Omega, 10 (2), 2269–2279. https://doi.org/10.1021/acsomega.4c09634
21. Kong, F., Song, J., Yang, Z. (2022). A novel short-term carbon emission prediction model based on secondary decomposition method and long short-term memory network. Environmental Science and Pollution Research, 29 (43), 64983–64998. https://doi.org/10.1007/s11356-022-20393-w
22. Xie, T., Huang, Z., Tan, T., Chen, Y. (2024). Forecasting China's agricultural carbon emissions: A comparative study based on deep learning models. Ecological Informatics, 82, 102661. https://doi.org/10.1016/j.ecoinf.2024.102661
23. Bi, J., Zhang, X., Yuan, H., Zhang, J., Zhou, M. (2022). A Hybrid Prediction Method for Realistic Network Traffic With Temporal Convolutional Network and LSTM. IEEE Transactions on Automation Science and Engineering, 19 (3), 1869–1879. https://doi.org/10.1109/tase.2021.3077537
24. Ankalaki, S., Thippeswamy, M. N. (2023). A novel optimized parametric hyperbolic tangent swish activation function for 1D-CNN: application of sensor-based human activity recognition and anomaly detection. Multimedia Tools and Applications, 83 (22), 61789–61819. https://doi.org/10.1007/s11042-023-15766-3
25. Sunkari, S., Sangam, A., P., V. S., M., S., Raman, R., Rajalakshmi, R., S., T. (2024). A refined ResNet18 architecture with Swish activation function for Diabetic Retinopathy classification. Biomedical Signal Processing and Control, 88, 105630. https://doi.org/10.1016/j.bspc.2023.105630
26. Harumy, T. H. F., Zarlis, M., Lydia, M. S., Efendi, S. (2023). A novel approach to the development of neural network architecture based on metaheuristic protis approach. Eastern-European Journal of Enterprise Technologies, 4 (4 (124)), 46–59. https://doi.org/10.15587/1729-4061.2023.281986
27. Harumy, T. H. F., Br Ginting, D. S., Manik, F. Y., Alkhowarizmi, A. (2024). Developing an early detection model for skin diseases using a hybrid deep neural network to enhance health independence in coastal communities. Eastern-European Journal of Enterprise Technologies, 6 (9 (132)), 71–85. https://doi.org/10.15587/1729-4061.2024.313983