COMPLETE
Journal of Computer, Electronic, and Telecommunication
ISSN 2723-4371, E-ISSN 2723-5912

Article

Application of Bayesian Optimization to the Temporal Fusion Transformer for Predicting Indonesia's Carbon Emissions

Daffa Ammar Mahendra[1], Faisal Muttaqin[2*], and Yisti Vita Via[3]

[1,2,3] Computer Science, Faculty of Computer Science, Veteran National Development University, East Java, Indonesia
* Correspondence: faisalmuttaqin.if@upnjatim.ac.id

Received: 29 August; Revised: 7 September 2026; Accepted: 12 October 2026; Published: 5 November 2026

Abstract: Carbon emissions are one of the main contributors to global climate change, highlighting the need for accurate prediction methods to support decision-making. This study aims to optimize the hyperparameters of the Temporal Fusion Transformer (TFT) using Bayesian Optimization for carbon emission forecasting in Indonesia. The dataset consists of annual carbon emission data from 1970 to 2024, with coal, oil, and gas emissions as input variables and CO_2 emissions as the target variable. The data were normalized using Min-Max Scaling and divided chronologically into training, validation, and testing sets. Bayesian Optimization was conducted using 20 trials for each random seed using a search space covering learning rate, hidden size, dropout rate, batch size, and epoch, with a sequence length of four time steps. Performance was evaluated using MAE, MAPE, and SMAPE across four random seeds. The results show that Bayesian Optimization reduced the average MAE from 52.3874 ± 2.4281 $MtCO_2$ to 46.7754 ± 5.5239 $MtCO_2$, while MAPE decreased from 7.13% ± 0.35% to 6.40% ± 0.83% and SMAPE from 7.47% ± 0.41% to 6.69% ± 0.82%. The model also captured the general trend of Indonesia's carbon emissions during the 2020-2024 testing period, indicating improved prediction performance.

Keywords: Temporal Fusion Transformer, Bayesian Optimization, Carbon Emissions Forecasting, Hyperparameter Optimization, Time Series.

1. Introduction

Climate change is one of the global challenges that has a significant impact on human life and environmental sustainability. One of its main causes is the increase in carbon dioxide (CO_2) concentrations in the atmosphere over the past few decades [1]. This rise in emissions is driven by various human activities, such as the use of fossil-fuel-based energy, industrial activities, the transportation sector, as well as economic growth and population growth [2]. This phenomenon has led to increasingly extreme climate changes, such as rising sea levels, melting polar ice caps, an increase in natural disasters, and ocean acidification, which disrupts the balance of ecosystems. Furthermore, atmospheric CO_2 concentrations have exceeded 400 ppm, largely driven by high fossil fuel consumption in tandem with industrial development and economic growth. This situation is also evident in many developing countries, including Indonesia, which still relies heavily on fossil fuel-based energy [3].

Carbon emissions in Indonesia have shown a steadily rising trend, particularly in the energy sector, where emissions have more than doubled over the past two decades. This situation indicates that the energy sector is one of the main contributors to the rise in carbon emissions in Indonesia [4]. In 2021, emissions from fuel combustion in Indonesia reached approximately 600 million metric tons of CO_2 (Mt CO_2), making Indonesia the ninth-largest emitter in the world. This increase is driven by the country's high reliance on fossil fuels, particularly coal, whose share rose from less than 10% in 2000 to nearly one-third of the total national energy supply in 2021. In addition, oil demand in the transportation sector has also increased by about 50% over the past two decades and accounts for nearly 40% of total end-user carbon emissions, primarily from land transportation [5].

This upward trend in emissions indicates that carbon emissions management requires an approach that not only focuses on monitoring current conditions but is also capable of forecasting future emissions trends. Therefore, a forecasting system is needed that can generate accurate estimates of carbon emissions to serve as the basis for developing emissions control policies and sustainable energy planning [6]. Prediction results can be used to support decision-making and evaluate the effectiveness of emission reduction strategies in achieving carbon neutrality targets [7]. However, the process of predicting carbon emissions is not a simple matter because the data used are time series and are influenced by various interacting factors. In addition to exhibiting strong temporal dependence, carbon emissions data also contain nonlinear relationships, noise, and potential missing values, thereby increasing the complexity of producing accurate estimates [8].

To address this complexity, various time series forecasting approaches have been developed, ranging from statistical methods to machine learning- and deep learning-based methods [9]. As the complexity of time-series data increases, more and more approaches are being developed that can capture temporal relationships and nonlinear patterns in multivariate data [10]. In line with these developments, Transformer-based architectures have increasingly been applied to time series forecasting problems because the attention mechanism allows the model to capture relevant temporal relationships from historical data [11].

One of the Transformer architecture extensions specifically designed for time series forecasting problems is the Temporal Fusion Transformer (TFT). The Temporal Fusion Transformer (TFT) is a deep learning model developed to address time series forecasting problems with long forecast horizons (multi-horizon forecasting) [12]. This model combines the Long Short-Term Memory (LSTM) architecture with a self-attention mechanism to learn short-term patterns while capturing long-term temporal relationships in the data. In addition to producing accurate predictions, the TFT also provides interpretability through the Variable Selection Network (VSN) and gating mechanisms, making it easier to identify the variables that most influence the prediction results [13]. Although it performs well, TFT's performance is heavily influenced by the hyperparameter configuration used. Manual hyperparameter tuning tends to be time-consuming and does not necessarily result in an optimal configuration. Therefore, an optimization method is needed that can determine hyperparameters more efficiently to improve model performance [14].

Based on these issues, this study applies Bayesian Optimization (BO) to optimize the hyperparameters of the Temporal Fusion Transformer (TFT) model for predicting Indonesia's carbon emissions. Although the combination of TFT and Bayesian Optimization has been explored in previous studies [13], [15], this study focuses on Indonesia's carbon emissions using coal, oil, and natural gas emissions as input variables. BO was chosen because it can systematically search for hyperparameter configurations by constructing a probabilistic model of the objective function, thereby identifying the most promising configurations [10]. This process allows for the search for hyperparameter configurations based on the evaluation results from previous experiments, so that promising configurations can be explored step by step [16]. Furthermore, model performance is evaluated using three metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE). MAE is used to measure the average absolute difference between predicted and actual values, while MAPE and SMAPE are used to assess the level of prediction error as a percentage, thereby enabling comparisons of model

performance across different data scales [17]. Through this evaluation, this study is expected to yield a more optimal approach to improving the accuracy of carbon emissions predictions and to provide a reference for applying hyperparameter optimization to deep learning models for time series forecasting problems.

2. Methods

2.1 Research Workflow

The research workflow was designed to provide an overview of the steps involved in developing a model to predict Indonesia's carbon emissions using TFT and Bayesian Optimization. Each step was carried out sequentially, from dataset collection to model performance evaluation. The research workflow used in this study is shown in Figure 1 below.

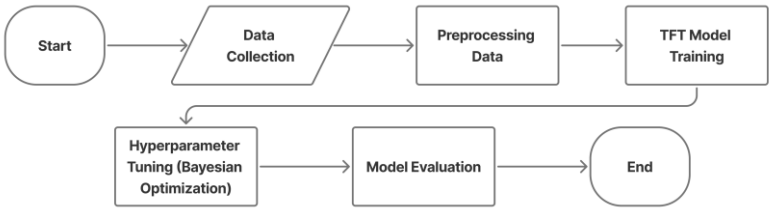


Figure 1. Research Workflow Diagram

The research workflow in Figure 1 illustrates the steps involved in developing a carbon emissions prediction model using the TFT model. The research process begins with the collection of a carbon emissions dataset, which will serve as the basis for developing the carbon emissions prediction model. The acquired dataset then undergoes a preprocessing stage to prepare the data before it is used in the model training process. Next, the TFT model is built using the preprocessed data, followed by hyperparameter optimization using Bayesian Optimization to obtain the model configuration that delivers the best performance. The final stage of the research involves evaluating the prediction results using the MAE, MAPE, and SMAPE metrics as a basis for assessing the model's performance.

2.2 Research Dataset

This study uses a carbon emissions dataset obtained from the Our World in Data website, covering the observation period from 1970 to 2024. The dataset consists of 13,926 data points covering various countries and annual observation periods, and includes various indicators related to carbon emissions from the energy use sector, economic activities, and other contributing factors. As an illustration, a portion of the dataset used in this study is shown in Table 1.

Table 1. Research Dataset Summary

Country	Year	Co2	Coal_co2	Oil_co2	Gas_co2	…
Afghanistan	1970	1.670	0.437	0.672	0.217	…
Afghanistan	1971	1.894	0.359	0.747	0.440	…
Afghanistan	1972	1.530	0.191	0.627	0.300	…
…	…	…	…	…	…	…
Zimbabwe	2024	13.701	8.459	4.855	-	…

Table 1 provides a summary of the dataset, which consists of carbon emissions data from various countries organized by observation year. Each data point contains information on the country, year, total carbon emissions (CO2), and carbon emissions from coal (Coal_co2), oil (Oil_co2), and natural gas (Gas_co2). In addition to these attributes, the dataset also includes several other attributes

indicated by (...), so overall, the dataset provides sufficiently comprehensive information to support the modeling process.

2.3 Preprocessing Data

The preprocessing stage is a crucial step in data processing aimed at improving data quality before it is used in the modeling process. This process is carried out to reduce data inconsistencies, retain relevant attributes, and prepare the data structure to meet the needs of machine learning models. In this study, the preprocessing stage includes feature selection, data cleaning, and dataset splitting. These three steps are performed to produce representative data that can support the TFT model's training process in learning carbon emission patterns more effectively.

1. Feature Selection

Feature selection is performed to identify attributes relevant to the research objectives so that the modeling process can focus on information that is correlated with the predicted variables. This step also helps reduce data complexity and improve the efficiency of the model training process. The variables used in this study are shown in Table 2.

Table 2. Research Variables

Variable	Category
Co2	Target
Coal_co2	Input
Oil_co2	Input
Gas_co2	Input

As shown in Table 2, this study uses one target variable, namely CO_2, and three input variables consisting of coal (Coal_co2), oil (Oil_co2), and natural gas (Gas_co2). These three input variables were selected because they represent the primary sources of carbon emissions from the energy sector in Indonesia; thus, they are expected to provide relevant information for studying patterns of change in total carbon emissions [18].

2. Data Splitting

After the research variables were determined during the feature selection stage, the dataset was subsequently divided into three parts, training data, validation data, and test data. For time-series data, the division was performed chronologically based on the temporal sequence to preserve the relationships between observations while minimizing the risk of data leakage, which could lead to biased evaluation results [16]. In this study, the training data were used to build a model based on historical carbon emission patterns, while the validation data were used in the hyperparameter tuning process using Bayesian optimization. The test data were used to evaluate the model's ability to predict Indonesia's carbon emissions during the test period. Details of the dataset split used in this study are presented in Table 3.

Table 3. Dataset Breakdown Table

Data Type	Year Range	Coverage	Total
Data Training	1970-2014	Global	**5.985**
Data Validation	2015-2019	Global	**665**
Data Testing	2020-2024	Indonesia	**5**

Table 3 shows the division of the dataset into training, validation, and test sets, along with the observation periods for each. The training period covers the years 1970-2014, followed by the validation period from 2015 to 2019 and the testing period from 2020 to 2024. This temporal division is applied to simulate real-world forecasting conditions and prevent future data leakage. Unlike typical independent data, time-series data exhibit temporal dependence, meaning the order of observations is not exchangeable and must not be randomized. The boundary years were selected to ensure that the validation period remains relatively close to the testing period. Using more recent observations for validation allows the hyperparameter optimization process to consider temporal patterns that are more representative of the conditions leading up to the testing period, while still maintaining a chronological separation between training, validation, and testing data. This separation allows hyperparameter optimization during the validation period and final evaluation during the testing period to be conducted using information from preceding time periods [19]. The training and validation sets include observations from various countries, including Indonesia, with Indonesia's data from 1970–2014 included in training and 2015–2019 in validation. Testing was then conducted exclusively on Indonesia for 2020–2024. Based on the total observations in Table 3, the training, validation, and testing sets comprise approximately 89.93%, 9.99%, and 0.08%, respectively.

3. Data Cleaning

Data cleaning is one of the key steps in preprocessing, aimed at improving data quality before it is used in the modeling process. This process is carried out to identify and correct issues in the data such as missing values, format inconsistencies, and data structure discrepancies thereby enhancing the reliability of the analysis results [20]. In this study, the data cleaning process involved checking for missing values, standardizing data formats and units, and sorting the data chronologically.

The handling of missing values began by removing countries with no data for the main variables and eliminating attributes with a missing value proportion exceeding 50%. Next, remaining empty values were filled using linear interpolation for each country, followed by the forward-fill and backward-fill methods to complete values at the beginning and end of the time series [21]. This series of processes resulted in a more complete dataset, with the number of observations reduced from 13.926 to 7.645.

After the missing value handling process was completed, all variables were standardized into the same format and units, and the data was sorted chronologically based on the year of observation. This step was taken to preserve the characteristics of the time series data so that the temporal relationships between observations would be maintained during the model training and evaluation processes.

2.4 Temporal Fusion Transformer

The Temporal Fusion Transformer (TFT) is a deep learning architecture developed to address multivariate time series prediction problems by combining Long Short-Term Memory (LSTM) networks and a multi-head attention mechanism [12]. The combination of these two mechanisms enables the model to effectively learn temporal relationships from historical data, making it suitable for predicting carbon emissions influenced by multiple emission source variables. For the carbon emissions forecasting task, the TFT architecture was trained on annual data using coal, oil, and natural gas emissions as input variables. Although the number of covariates is relatively limited, these variables represent the primary emission sources directly related to the forecasting target. The TFT's temporal modeling capabilities allow the model to utilize historical information from these variables to capture changes in carbon emissions over time.

In this study, the TFT model was applied to predict carbon emissions using preprocessed data. The steps for applying the TFT model in this study are shown in Figure 2. The process begins with data normalization using the Min-Max Scaling method, followed by sequence formation using the sliding window technique; the data is then fed into the TFT model, which consists of a Variable Selection Network, an LSTM Encoder-Decoder, and Multi-head Attention. During the training

process, the training loss and validation loss values are evaluated at each epoch. Once the early-stopping criteria are met, the best model is saved and used to make predictions on the test data; the prediction results are then evaluated using the MAE, MAPE, and SMAPE metrics.

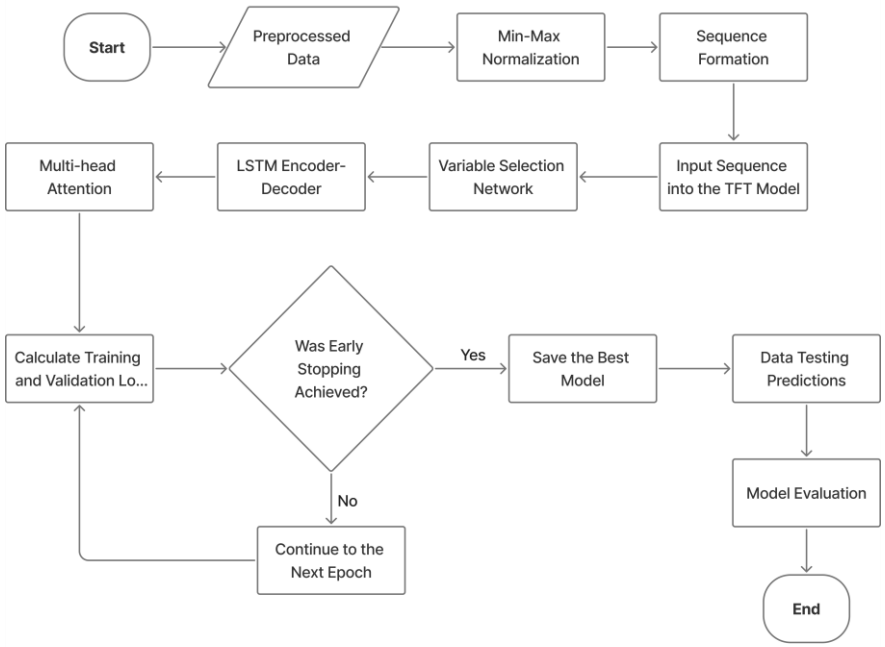


Figure 2. TFT Model Flowchart

Before entering the training phase, all numerical variables are normalized using the Min-Max Scaling method to ensure they have a uniform range of values [6]. This normalization process is expressed in Equation (1).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

with :

- x = original value,
- x' = normalized value,
- x_{min}= minimum value,
- x_{max} = maximum value.

Next, the data is organized into a sequence using a sliding window technique with a length of 4 time steps, so that each sample consists of four historical data points to predict the carbon emission value for the next period. The input variables in each sequence consist only of observations from the four preceding time steps. Thus, when predicting carbon emissions for a given year, the coal, oil, and natural gas emission values from the prediction year are not included as input variables. This ensures that the model uses only historical information available prior to the prediction period and avoids information leakage from the prediction year.

To preserve the temporal structure across countries, sequence formation was performed separately for each country. The data for each country were first ordered chronologically and then transformed into four-time-step sequences using a sliding window. Therefore, observations from different countries were not combined within the same sequence. The resulting sequences from different countries were subsequently combined and used to train a single TFT model with shared parameters. During training, a mini-batch may contain sequences from different countries, but each sequence remains associated with a single country. This allows the model to learn shared temporal patterns from the global training data while maintaining the temporal continuity of each country's observations. During the training process, TFT utilizes a multi-head attention mechanism to calculate

the importance of each time step for the prediction process [12]. This mechanism is expressed in Equation (2).

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

with :

- Q = query,
- K = key,
- V = value,
- T = transpose,
- $\sqrt{d_k}$ = dimension key.

The model training process is performed using training data, while validation data is used during the hyperparameter optimization process using Bayesian Optimization. The optimal configuration resulting from the optimization is then used as the final model to generate predictions on the test data. Details of the initial hyperparameters used in the TFT model are presented in Table 4.

Table 4. TFT Hyperparameter Configuration

Hyperparameter	Value
Learning Rate	0.0007
Hidden Units	64
Dropout Rate	0.4
Batch Size	64
Sequence Length	4
Epoch	85

The initial hyperparameter configuration used in the TFT model is shown in Table 4. The model uses a learning rate of 0.0007, a hidden layer size of 64, a dropout rate of 0.40, and a batch size of 64. Additionally, the sequence length is set to 4 years to provide historical context for the prediction process using annual data. This configuration is then used as the baseline model before hyperparameter optimization is performed using Bayesian Optimization.

2.5 Bayesian Optimization

Determining the appropriate hyperparameter values is one of the key factors in achieving optimal performance in deep learning models. Using inappropriate parameter values can cause the learning process to be ineffective, thereby reducing prediction accuracy. Therefore, this study employs Bayesian Optimization as a hyperparameter search method that utilizes the evaluation results from previous experiments to determine the hyperparameter configuration for subsequent experiments [13].

Unlike approaches that evaluate parameter combinations independently, Bayesian Optimization uses the results of previous evaluations to guide the selection of subsequent hyperparameter configurations. This allows the search process to focus on configurations that show greater potential based on the observed validation results. The workflow for hyperparameter optimization using Bayesian Optimization in this study is shown in Figure 3.

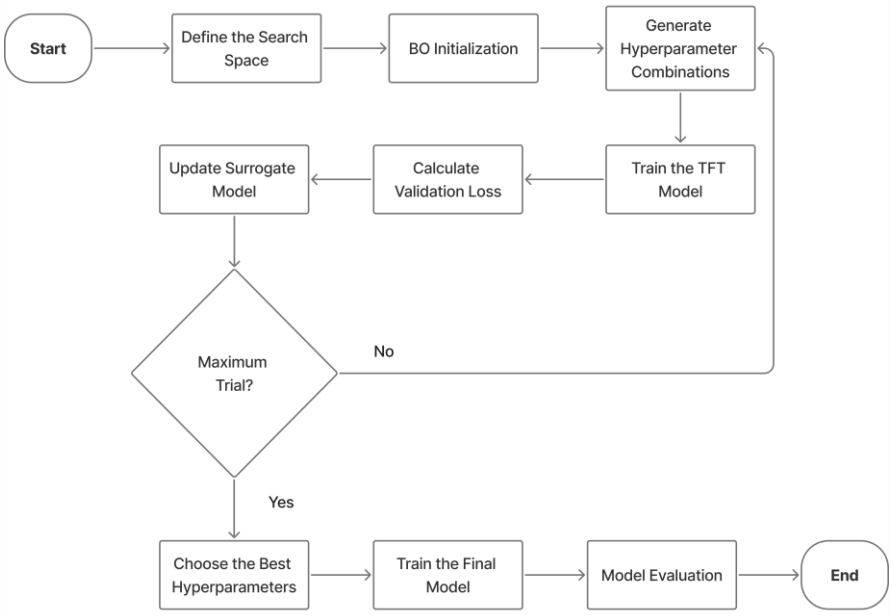


Figure 3. Bayesian Optimization Diagram

As shown in Figure 3, hyperparameter optimization in this study was performed iteratively using Bayesian Optimization. The process began with defining the search space; Bayesian Optimization then generated hyperparameter combinations that were evaluated by training the TFT model. The validation loss values obtained were used to update the surrogate model, allowing for a more optimal selection of the next hyperparameter combination. This cycle was repeated until the specified number of trials was reached. Next, the best hyperparameter combination is used to train the final model before evaluation using the MAE, MAPE, and SMAPE metrics.

In this study, the optimization process focused on four main hyperparameters: learning rate, hidden size, dropout, batch size, and epoch. These four parameters were selected because they influence the model's ability to learn historical carbon emission patterns [15]. During the optimization process, each parameter combination is evaluated using validation data, and the configuration with the lowest validation loss is selected as the best configuration. Mathematically, the optimization objective is expressed in Equation (3).

$$\theta^* = argmin_\theta f(\theta) \tag{3}$$

In Equation (3), θ represents the optimized combination of hyperparameters, while $f(\theta)$ is the objective function used as the basis for evaluating model performance. The optimal combination of hyperparameters is obtained by minimizing the value of this objective function [6]. In this process, Tree-structured Parzen Estimator (TPE) is used to model the relationship between hyperparameter configurations and objective function values based on the results of previous experiments, thereby determining which configurations to evaluate in the next experiment. The TPE mechanism for determining the next hyperparameter candidate is explained through the separation of observations based on objective function values and the modeling of the probability density function in Equation (4) below.

$$x_{next} = \arg\max_{x \in \mathcal{X}} \frac{p(x|\mathcal{D}^{(l)})}{p(x|\mathcal{D}^{(g)})} \tag{4}$$

The TPE algorithm works by dividing the hyperparameter observation history into two groups based on the quantile threshold value γ (top-quantile), the group with the best performance ($\mathcal{D}^{(l)}$) and the group with poorer performance ($\mathcal{D}^{(g)}$). The selection of the next hyperparameter candidate

(x_{next}) is performed by maximizing this density ratio [22]. In this study, hyperparameter optimization was performed using Optuna with four seed values 42, 123, 100, and 234 each consisting of 20 trials. The number of trials was determined based on computational efficiency considerations. Each trial evaluated combinations of hyperparameters based on validation loss values; the configuration with the lowest validation loss was then used for evaluation for each seed. The evaluation results from all seeds were subsequently summarized as mean and standard deviation.

Table 5. Hyperparameter Optimization Range

Hyperparameter	Range of Values
Learning Rate	0.0001 – 0.01
Hidden Units	64 – 256
Dropout Rate	0.1 – 0.4
Batch Size	32, 64, 128
Epoch	100 – 300

The hyperparameter ranges in Table 5 are used as the search space in the Bayesian optimization process. Each hyperparameter is assigned a sufficiently wide range of values so that the optimization process can explore various parameter combinations that have the potential to improve model performance. The learning rate, hidden units, dropout rate, batch size, and epoch were selected because they influence the training process and the model's ability to learn data patterns. Through this exploration process, a hyperparameter configuration with the lowest validation loss was obtained, which was subsequently used as the final model.

2.6 Model Evaluation

To measure the accuracy of the prediction results, this study uses three evaluation metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE). These three metrics were selected because they provide insights into prediction errors from different perspectives. MAE is used to measure the average absolute difference between predicted and actual values; MAPE indicates the magnitude of the error as a percentage of the actual value; and SMAPE is used to address MAPE's limitations when both actual and predicted values approach zero, thereby yielding a more stable evaluation.

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i | \quad (5)$$

Equation (4) is used to calculate the average absolute difference between the actual value (y_i) and the predicted value ($\hat{y}_i$), where (n) represents the number of observations. A smaller MAE value indicates a lower level of prediction error.

- Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \frac{|yi-\hat{y}i|}{yi} \quad (6)$$

Equation (5) calculates the mean absolute percentage error between the actual values (y_i) and the predicted values ($\hat{y}_i$) across the entire set of observed data (n). The smaller the MAPE value, the better the model's predictive performance.

- Symmetric Mean Absolute Percentage Error (SMAPE)

$$sMAPE = \frac{100\%}{m}\sum_{i=1}^{m} \frac{|Xi-Yi|}{\frac{(|Xi|+|Yi|)}{2}} \quad (7)$$

Equation (6) is a modification of MAPE that uses the average of the actual values (y_i) and the predicted values ($\hat{y}_i$) as the denominator, thereby yielding a more balanced measure of error. A smaller SMAPE value indicates a higher level of prediction accuracy.

3. Results

This section presents the results of implementing the Temporal Fusion Transformer (TFT) model for predicting Indonesia's carbon emissions, as well as the impact of Bayesian Optimization on the model's performance. The analysis is based on training results, evaluation using error metrics, and visualizations of predictions compared to actual data.

3.1. Results of the TFT Model Training

As an initial step, the TFT model was built using an initial hyperparameter configuration to generate a baseline model. The resulting predictions were then evaluated using the MAE, MAPE, and SMAPE metrics as a basis for comparison with the model optimized using Bayesian Optimization.

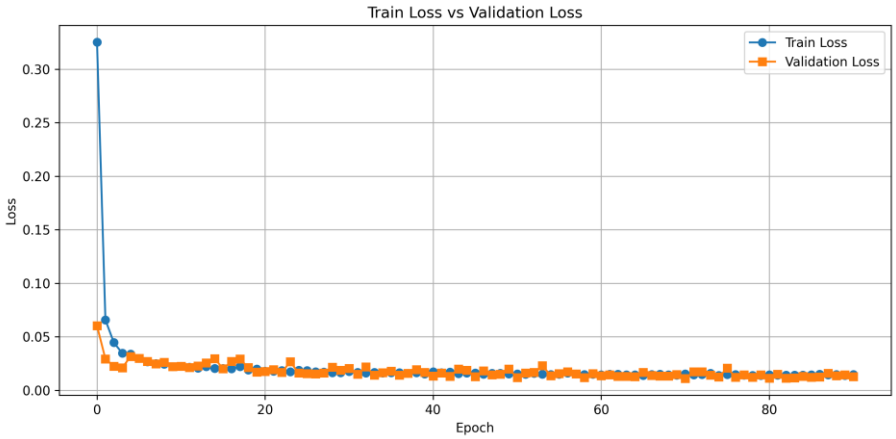


Figure 4. TFT Model Training Chart

To observe the model's learning process during training, the changes in training loss and validation loss are visualized in Figure 4. The figure shows the training results of the TFT baseline model with seed 42, which yielded the best evaluation results. Both curves declined quite sharply in the early epochs, then tended to stabilize with minor fluctuations until the end of the training process. The relatively similar patterns of change in training loss and validation loss indicate that the model's learning process proceeded quite stably and showed no strong signs of overfitting.

In addition to being observed through the training curves, the model's performance was also quantitatively evaluated using the MAE, MAPE, and SMAPE metrics on the test data. The evaluation results for the TFT baseline model are presented in Table 6.

Table 6. Results of the TFT Model Evaluation

Evaluation Metrics	Seed 42	Seed 100	Seed 123	Seed 234
MAE	49.7622	51.8702	55.6306	52.2867
MAPE	6.81%	7.02%	7.63%	7.05%
SMAPE	7.09%	7.35%	8.05%	7.37%

Based on the evaluation results in Table 6, the results show that the baseline TFT model maintains relatively stable performance across all four seed values, despite variations in prediction error between runs. The lowest error was consistently obtained with seed 42, while seed 123 produced the highest error across all three metrics. This pattern suggests that random initialization

has a certain influence on the model's prediction performance. However, the overall variation remains limited, indicating that the baseline model is not particularly sensitive to the choice of seed. Therefore, these results provide a consistent benchmark for evaluating the impact of Bayesian Optimization on the TFT model.

3.2. Results of Hyperparameter Optimization Using Bayesian Optimization

After obtaining the results of the baseline model evaluation, a hyperparameter optimization process was conducted to identify a parameter configuration better suited to the characteristics of Indonesia's carbon emissions data. The optimization focused on four hyperparameters: learning rate, hidden layer size, dropout, and batch size. The search process consisted of 20 trials, with each trial evaluating a different combination of hyperparameters based on the objective function value until the configuration with the best performance was identified.

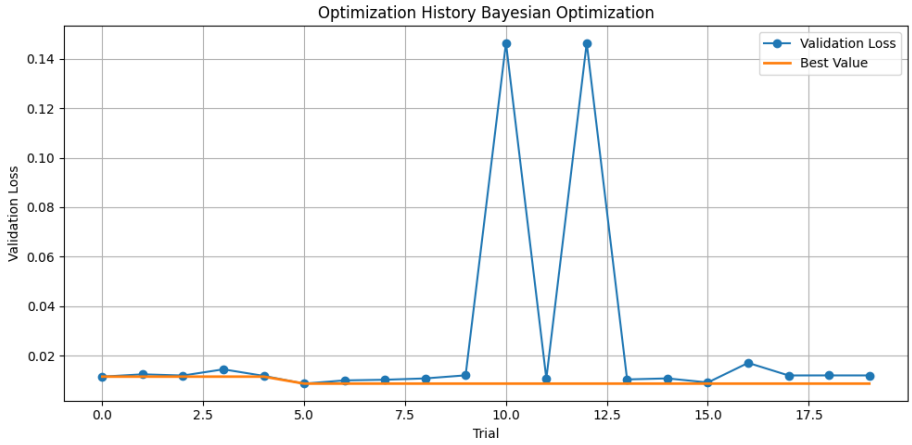


Figure 5. Optimization History Chart

The optimization process shown in Figure 5 presents the results obtained using an initial value of 42. The validation loss values varied across the evaluated trials, with most trials producing values within a relatively narrow range. However, substantial increases in validation loss can be observed at trials 10 and 12, indicating that the corresponding hyperparameter combinations resulted in considerably poorer validation performance. In contrast, the validation loss decreased to its lowest value at trial 5 and remained relatively stable in the subsequent trials, although several fluctuations were still observed. The best value therefore remained at the level obtained in trial 5, as none of the subsequent trials produced a lower validation loss. Based on this result, the hyperparameter configuration corresponding to the lowest validation loss was selected for the optimized TFT model.

Table 7. Bayesian Optimization Hyperparameter Configurations

Hyperparameter	**Bayesian Optimization**			
	Seed 42	**Seed 100**	**Seed 123**	**Seed 234**
Learning Rate	0.00211	0.000107	0.000103	0.000478
Hidden Size	256	128	64	64
Dropout Rate	0.1554	0.2065	0.2375	0.1524
Batch Size	32	32	32	32
Epoch	279	299	242	266

The optimization results in Table 7 show that Bayesian Optimization produced different hyperparameter configurations across the four seeds. Compared with the initial TFT settings, the learning rate decreased across all seeds, ranging from 0.000103 to 0.00211, while the dropout rate decreased to values between 0.1524 and 0.2375. The batch size consistently changed from 64 to 32, whereas the hidden size varied between 64, 128, and 256, indicating that the preferred model capacity differed across optimization runs. The number of epochs also increased from the initial 85 epochs, ranging from 242 to 299 epochs. This variation suggests that the optimization process identified different combinations of hyperparameters according to the random initialization, rather than converging to a single configuration. The resulting configurations were then used for model training and evaluation for each corresponding seed, with the training and validation loss presented in Figure 6.

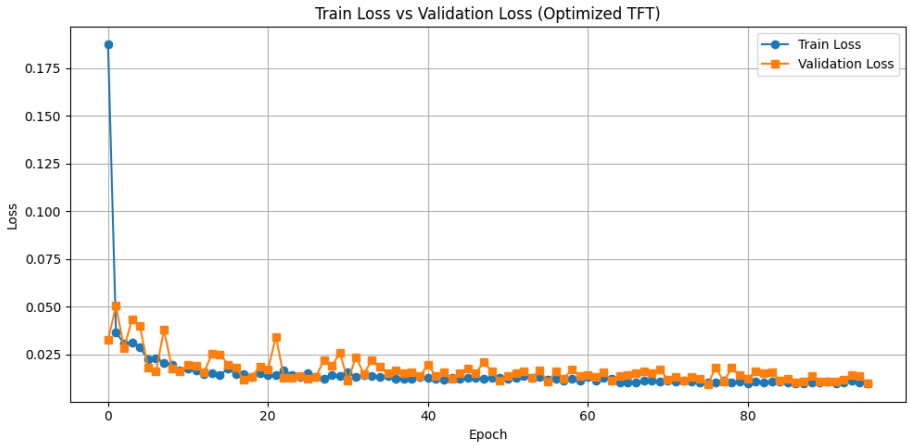


Figure 6. Optimized TFT Model Training Chart

As shown in Figure 6, the training and validation loss values decrease sharply in the early epochs and then converge to a relatively stable range. Although the validation loss value fluctuates slightly, it does not show a sustained increase as the training loss decreases, indicating no strong signs of overfitting. This suggests that the TFT model optimized using seed 42 has reached a stable training phase. The model was then evaluated on the test data using MAE, MAPE, and SMAPE, with the results presented in Table 8.

Table 8. Results of the Evaluation of the Optimized TFT Model

Evaluation Metrics	Seed 42	Seed 100	Seed 123	Seed 234
MAE	39.2249	48.9778	52.2223	46.6768
MAPE	5.25%	6.66%	7.21%	6.48%
SMAPE	5.56%	6.95%	7.51%	6.73%

The results in Table 8 illustrate the performance obtained after applying the optimized hyperparameter configuration to the four random initial values. The differences between runs indicate that the optimization process does not produce identical predictive behavior for different random initializations. Nevertheless, the resulting models maintain comparable error rates across all evaluation metrics, indicating that the optimized configuration remains applicable to repeated runs. The overall impact of this optimization is further analyzed by comparing these results with the performance of the baseline TFT in the following subsection.

3.3. Model Performance Comparison

To evaluate the impact of Bayesian Optimization on model performance, we compared the evaluation results of the baseline TFT model with those of the optimized TFT model. The comparison

was conducted using three evaluation metrics: MAE, MAPE, and SMAPE. A visualization of the comparison results is shown in Figure 7.

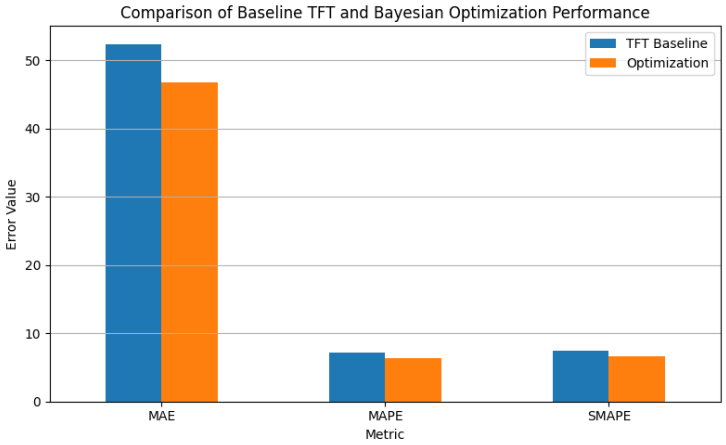


Figure 7. Model Performance Comparison

As shown in Figure 7, all three evaluation metrics show lower average error values after applying Bayesian Optimization to the TFT model. The average MAE decreased from 52.3874 to 46.7754, while the average MAPE decreased from 7.13% to 6.40% and the average SMAPE decreased from 7.47% to 6.69%. These results indicate that the optimized TFT configuration achieved lower prediction errors than the baseline model across the four evaluated seeds. The largest reduction was observed in the MAE, followed by the MAPE and SMAPE, with improvements of 10.71%, 10.21%, and 10.42%, respectively. The corresponding standard deviations and performance improvements are presented in Table 9.

Table 9. Comparison of Model Evaluations

Model	MAE	MAPE	SMAPE
TFT Baseline	52.3874 ± 2.4281	7.13% ± 0.35%	7.47% ± 0.41%
TFT + BO	46.7754 ± 5.5239	6.40% ± 0.83%	6.69% ± 0.82%
Improvement	10.71%	10.21%	10.42%

The comparison in Table 9 shows that the application of Bayesian Optimization improved the average prediction performance of the TFT model across all four seeds. The average MAE decreased from 52.3874 to 46.7754, corresponding to an improvement of 10.71%, while MAPE and sMAPE decreased by 10.21% and 10.42%, respectively. The reductions across all three metrics indicate that the optimized TFT model produced lower prediction errors than the baseline model across the four evaluated seeds. These results suggest that Bayesian Optimization can improve TFT prediction accuracy through hyperparameter optimization. However, the higher standard deviations observed in the optimized model indicate greater variation in performance across different random seeds compared with the baseline model.

3.4. Analysis of Model Prediction Results

In addition to being evaluated using quantitative metrics, the model's performance was also analyzed by comparing actual values with predicted results on the test data. This visualization aims to provide an overview of the model's ability to track trends in Indonesia's carbon emissions following the hyperparameter optimization process. The model's predictions compared to the actual data are shown in Figure 8.

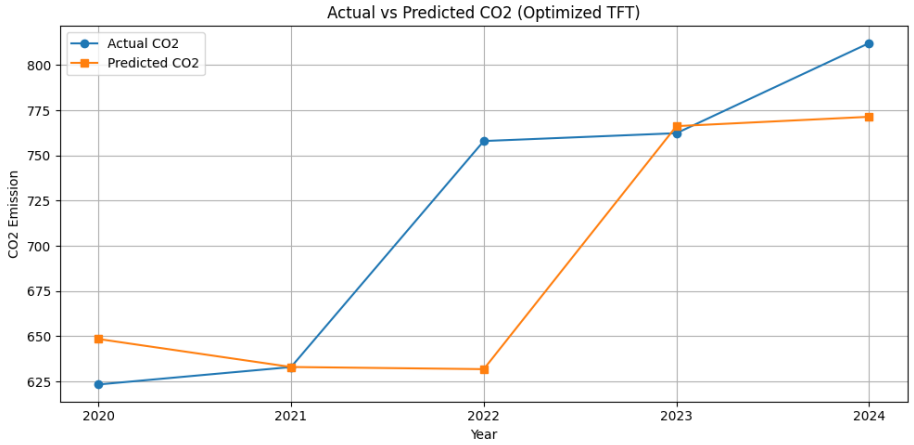


Figure 8. Actual Results and Optimization Model Predictions

Based on Figure 8, the model's predictions generally tracked the trend in carbon emissions during the testing period, although there were discrepancies between actual and predicted values in some years. In 2020, the model tended to overpredict, with a predicted value of 648.46 $MtCO_2$ compared with the actual value of 623.30 $MtCO_2$. In 2021, the predicted value of 632.97 $MtCO_2$ was relatively close to the actual value of 632.95 $MtCO_2$. The most striking discrepancy was observed in 2022, when actual emissions increased sharply to 758.02 $MtCO_2$, while the model predicted only 631.78 $MtCO_2$. This indicates that the model was not yet fully capable of capturing the substantial increase in emissions that occurred in that year. In 2023, the prediction was again close to the actual value, with 766.27 $MtCO_2$ predicted compared with 762.36 $MtCO_2$ actual. However, the gap increased again in 2024, when the actual value reached 812.22 $MtCO_2$ while the model predicted 771.43 $MtCO_2$.

Overall, the visualization in Figure 8 shows that the model was able to capture the general pattern of changes in carbon emissions during the testing period, particularly the upward movement observed after 2022. However, the differences between actual and predicted values in several years indicate that the model's ability to capture relatively sharp changes remains limited. This provides a basis for examining the prediction errors in greater detail for each year. Therefore, further analysis was conducted on the absolute prediction error values shown in Figure 9.

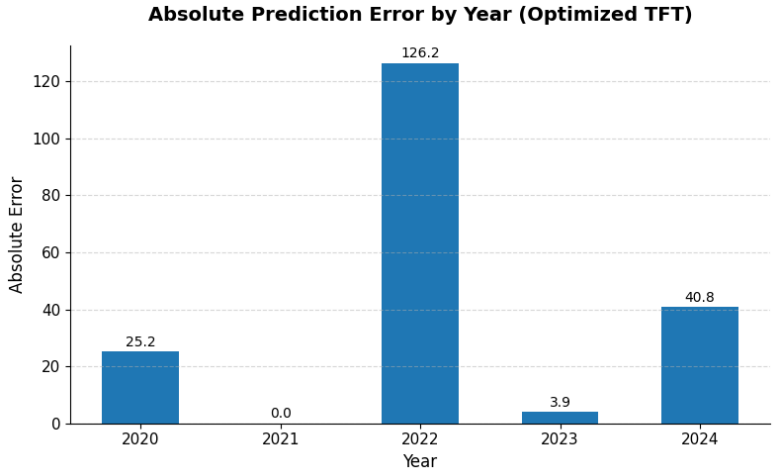


Figure 9. Graph of the Difference Between Predicted and Actual Values

As shown in Figure 9, the absolute prediction error varies across the test years, with the largest error occurring in 2022. The model recorded an error of 0.0 $MtCO_2$ in 2021, indicating that the prediction was very close to the actual value, while the error increased sharply to 126.2 $MtCO_2$ in 2022. This value is approximately 3.22 times the MAE of 39.2249 $MtCO_2$ obtained by the optimized

TFT model on seed 42. The large error in 2022 indicates that the model had difficulty capturing the sharp increase in CO_2 emissions during that year. Based on the test data, emissions increased from 632.95 $MtCO_2$ in 2021 to 758.02 $MtCO_2$ in 2022, representing an increase of approximately 19.8%. Such a substantial change may be difficult to capture using the limited input variables employed in this study, which consist of historical CO_2 emissions and emissions from coal, oil, and natural gas. After 2022, the error decreased considerably to 3.9 $MtCO_2$ in 2023, before increasing again to 40.8 $MtCO_2$ in 2024. This variation indicates that the model was able to produce predictions close to the actual values in some years, but its performance remained sensitive to substantial changes in the emission pattern.

4. Discussion

The results show that applying Bayesian Optimization improves the performance of the TFT model. The average MAE decreased from 52.3874 to 46.7754, while MAPE and sMAPE decreased from 7.13% to 6.40% and from 7.47% to 6.69%, respectively. Previous studies have explored various approaches for configuring and optimizing TFT and related Transformer-based models [6], [13], [14]. The results of these studies and the present study are summarized in Table 10.

Table 10. Comparison of Previous Studies and the Present Study

Study	Application	Model & Optimization	Evaluation Results
Nazir et al. [14]	Household energy consumption	TFT + trial-and-error	MAE = 1.50; SMAPE = 26.46%
Stefenon et al. [13]	Dam water-level forecasting during flood events	AutoTFT; automated hyperparameter tuning	RMSE = 2.78×10^{-3}; RMSPE = 3.23×10^{-4}; MAE = 2.38×10^{-3} *(P = 3)*
Alhussan & Metwally [6]	Vehicle CO_2 emissions	TFT + FbOA	MAE = 3.13×10^{-5}; RMSE = 9.23×10^{-4}; R^2 = 0.9747
This Study	Indonesia carbon emissions	TFT + Bayesian Optimization	MAE = 46.7754 ± 5.5239; MAPE = 6.40% ± 0.83%; SMAPE = 6.69% ± 0.82%

Table 10 shows that different optimization strategies have been applied across TFT-based forecasting studies. Manual parameter configuration was used in Nazir et al. [14], automated hyperparameter tuning was applied in Stefenon et al. [13], and the Football Optimization Algorithm was used in Alhussan and Metwally [6]. In this study, Bayesian Optimization was used to determine the hyperparameter configuration of TFT for Indonesia's carbon emission forecasting. However, the reported error values should not be interpreted as a direct ranking of model performance because the studies use different datasets, prediction targets, measurement units, forecasting horizons, and experimental settings. The comparison therefore provides contextual information on how different optimization approaches have been applied to TFT-based forecasting problems rather than a direct performance comparison.

The prediction results also show that the model still struggles when there are fairly sharp changes in emissions. This was most evident in 2022, when actual emissions increased by approximately 19.8% from 632.95 $MtCO_2$ in 2021 to 758.02 $MtCO_2$, while the absolute error reached

126.2 $MtCO_2$, about 3.22 times the MAE of the optimized model at seed 42. Subsequently, the error decreased to 3.9 $MtCO_2$ in 2023 before rising again to 40.8 $MtCO_2$ in 2024. This pattern suggests that the model can better track changes that remain relatively close to historical patterns but faces challenges when emissions increase substantially over a short period. This limitation may be related to the information used as input, which in this study was limited to historical emissions from coal, oil, and natural gas.

4.1. Limitations

This study has several limitations that should be taken into account when interpreting the results. The test data consists of only five annual observations for Indonesia during the 2020-2024 period consequently, the MAE, MAPE, and SMAPE values are quite sensitive to errors in each year and cannot yet reflect the model's performance over a longer period. Additionally, the model uses only coal, oil, and natural gas emissions as input variables, so other factors that could potentially influence emissions are not accounted for. This study also focuses on comparing the TFT before and after Bayesian optimization without involving other forecasting models as comparators. Evaluations using a longer test period, the inclusion of relevant variables, and the use of rolling-origin or walk-forward methods could be considered in future research.

5. Conclusion

This study applies Bayesian Optimization to optimize the hyperparameters of the Temporal Fusion Transformer for predicting Indonesia's carbon emissions. The model uses coal, oil, and natural gas emissions as input variables, while the optimization process is conducted to obtain a hyperparameter configuration based on evaluation results using validation data. Testing with four random seeds shows that the optimized TFT model produces a lower average error compared to the baseline model. The average MAE decreased from 52.3874 ± 2.4281 to 46.7754 ± 5.5239, while MAPE and SMAPE decreased from 7.13% ± 0.35% to 6.40% ± 0.83% and from 7.47% ± 0.41% to 6.69% ± 0.82%, respectively. Based on these results, Bayesian Optimization can be used to tune TFT hyperparameters and improve prediction performance in the context of this study. However, the results obtained are still limited by the dataset and variable configuration used, particularly since the testing period covered only five years. Future development could focus on using a longer testing period, adding relevant variables, and applying time-based evaluation schemes such as rolling-origin or walk-forward to obtain a more comprehensive picture of the model's performance.

References

1. M. Anugrah Putra and S. Andryana, "Perbandingan Algoritma Arima Dan Lstm Dalam Peramalan Tingkat Konsentrasi Co2 Emisi Atmosfer Untuk Masa Mendatang," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 3, pp. 4150–4157, 2025, doi: 10.36040/jati.v9i3.13511.
2. S. Elmi, R. Yanti, Mardainis, and H. Asnal, "Prediksi Emisi Co2 Di Indonesia Menggunakan Pendekatan Hybrid Arima Dan LSTM," *J. Ilm. BETRIK*, vol. 16, no. 02, pp. 95–108, 2025.
3. A. Setiyono, A. Dharmawan, and Z. Furqon, "Prediksi Faktor Penyebab Emisi CO 2 pada Kendaraan Menggunakan Gradient Boosting Regression," vol. 09, no. 04, pp. 243–247, 2023.
4. Parid Pakaya, Fitryane Lihawa, and Dewi Wahyuni K. Baderan, "Efektivitas Ruang Terbuka Hijau Publik dalam Menyerap Emisi Karbon Dioksida untuk Mendukung Keberlanjutan Lingkungan Perkotaan," *Hidroponik J. Ilmu Pertan. Dan Teknol. Dalam Ilmu Tanam.*, vol. 1, no. 3, pp. 54–75, Nov. 2024, doi: 10.62951/hidroponik.v1i3.199.
5. International Energy Agency, "An Energy Sector Roadmap to Net Zero Emissions in Indonesia." Accessed: Mar. 26, 2026. [Online]. Available: https://www.iea.org/reports/an-

energy-sector-roadmap-to-net-zero-emissions-in-indonesia

6. A. A. Alhussan and M. Metwally, "Enhanced CO2 Emissions Prediction Using Temporal Fusion Transformer Optimized by Football Optimization Algorithm," *Mathematics*, vol. 13, no. 10, pp. 1–47, 2025, doi: 10.3390/math13101627.
7. K. Calvin *et al.*, *IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland.* 2023. doi: 10.59327/IPCC/AR6-9789291691647.
8. X. Wen and W. Li, "Time Series Prediction Based on LSTM-Attention-LSTM Model," *IEEE Access*, vol. 11, no. May, pp. 48322–48331, 2023, doi: 10.1109/ACCESS.2023.3276628.
9. B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 379, no. 2194, 2021, doi: 10.1098/rsta.2020.0209.
10. M. A. Iqbal, J. M. Gil, and S. K. Kim, "Attention-Driven Hybrid Ensemble Approach with Bayesian Optimization for Accurate Energy Forecasting in Jeju Island's Renewable Energy System," *IEEE Access*, vol. 13, no. January, pp. 7986–8010, 2025, doi: 10.1109/ACCESS.2025.3526943.
11. S. M. Al-selwi, M. Fadzil, and S. Jadid, "RNN-LSTM : From applications to modeling techniques and beyond — Systematic review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. May, pp. 1–32, 2024, doi: 10.1016/j.jksuci.2024.102068.
12. B. Lim, S. Arık, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecast.*, vol. 37, no. 4, pp. 1748–1764, 2021, doi: 10.1016/j.ijforecast.2021.03.012.
13. S. F. Stefenon, L. O. Seman, L. S. A. da Silva, V. C. Mariani, and L. dos S. Coelho, "Hypertuned temporal fusion transformer for multi-horizon time series forecasting of dam level in hydroelectric power plants," *Int. J. Electr. Power Energy Syst.*, vol. 157, no. November 2023, p. 109876, 2024, doi: 10.1016/j.ijepes.2024.109876.
14. A. Nazir, A. K. Shaikh, A. S. Shah, and A. Khalil, "Forecasting energy consumption demand of customers in smart grid using Temporal Fusion Transformer (TFT)," *Results Eng.*, vol. 17, no. January, p. 100888, 2023, doi: 10.1016/j.rineng.2023.100888.
15. J. Xu, C. Wu, Y.-F. Li, G. Danoy, and P. Bouvry, "A Unified Hyperparameter Optimization Pipeline for Transformer-Based Time Series Forecasting Models," 2025, [Online]. Available: http://arxiv.org/abs/2501.01394
16. J. Zuo, M. He, Z. Yu, and J. Wang, "Short-term passenger flow prediction of urban rail transit on Bayesian Optimization- Bidirectional Long Short-Term Memory with Causal Temporal Pattern Attention," *Int. J. Transp. Sci. Technol.*, no. xxxx, 2025, doi: 10.1016/j.ijtst.2025.01.008.
17. C. Li, M. Li, and Z. Qiu, "A long-term dependable and reliable method for reactor accident prognosis using temporal fusion transformer," *Front. Nucl. Eng.*, vol. 3, no. February, pp. 1–21, 2024, doi: 10.3389/fnuen.2024.1339457.
18. Regina Citra Kurnia Pangestu and Anak Agung Ketut Ayuningsasi, "Pengaruh Konsumsi Energi Sektor Industri, Rumah Tangga, dan Transportasi terhadap Emisi Karbon di Indonesia," *Inisiat. J. Ekon. Akunt. dan Manaj.*, vol. 3, no. 4, pp. 297–311, 2024, doi: 10.30640/inisiatif.v3i4.3154.

19. R. F. Barber, A. Pananjady, and G. Tech, "Predictive inference for time series : why is split conformal effective despite temporal dependence ?," pp. 1–22, 2026.
20. A. M. Sharifnia and D. K. Thapa, "A Primer of Data Cleaning in Quantitative Research : Handling Missing Values and Outliers," pp. 970–975, 2026, doi: 10.1111/jan.16908.
21. F. Falchi, M. Girardi, G. Gurioli, N. Messina, C. Padovani, and D. Pellegrini, "Deep learning and structural health monitoring: Temporal Fusion Transformers for anomaly detection in masonry towers," *Mech. Syst. Signal Process.*, vol. 215, Jun. 2024, doi: 10.1016/j.ymssp.2024.111382.
22. S. Watanabe, "Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance," pp. 1–70.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).