Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

Hybrid Machine Learning Approaches for Atmospheric CO_2 Prediction: Evaluating Regression and Ensemble Models with Advanced Feature Engineering

Gregorius Airlangga*

Information System Study Program, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia
Email: gregorius.airlangga@atmajaya.ac.id
Correspondence Author Email: gregorius.airlangga@atmajaya.ac.id
Submitted: **13/03/2025**; Accepted: **26/03/2025**; Published**: 27/03/2025**

Abstract−The accurate prediction of atmospheric CO_2 concentrations is essential for understanding climate change dynamics and developing effective environmental policies. This study evaluates the predictive capabilities of various machine learning models, including ensemble-based regressors such as Random Forest, Gradient Boosting, and XGBoost, alongside traditional regression models such as Support Vector Regression (SVR), Ridge, and Lasso regression. The dataset, derived from meteorological observations, was preprocessed using multiple feature scaling techniques, including StandardScaler, MinMaxScaler, and RobustScaler, followed by feature engineering techniques such as polynomial transformation and Principal Component Analysis (PCA) to enhance predictive accuracy. Model performance was assessed using the coefficient of determination (R^2) and cross-validation techniques. The results indicate that tree-based models, including Random Forest and XGBoost, struggled to generalize well, exhibiting negative R^2 values due to overfitting and an inability to capture the temporal dependencies in CO_2 variations. SVR emerged as the best-performing model, though its predictive power remained limited. Computational complexity analysis revealed that tree-based methods incurred high processing costs, while linear models such as Ridge and Lasso demonstrated lower complexity but failed to capture non-linear dependencies. The study highlights the challenges of CO_2 prediction using conventional machine learning techniques and underscores the need for advanced deep learning approaches, such as hybrid Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models, to better capture spatial and temporal dependencies. Future research should explore integrating external environmental factors and leveraging deep learning architectures to improve predictive performance.

Keywords: CO_2 Prediction; Machine Learning; Time-Series Forecasting; Ensemble Learning; Feature Engineering

1. INTRODUCTION

Climate change poses one of the most significant challenges of the 21st century, with rising atmospheric CO_2 concentrations being a primary driver of global warming [1]–[3]. Anthropogenic activities, such as fossil fuel combustion, deforestation, and industrial processes, have elevated CO_2 levels, leading to severe environmental consequences, including temperature increases, extreme weather events, and ecosystem disruptions [4]–[6]. Accurate forecasting of CO_2 concentrations is essential for understanding climate dynamics and formulating effective mitigation strategies [7]–[9]. Traditional CO_2 forecasting methods have predominantly relied on statistical and regression models, such as AutoRegressive Integrated Moving Average (ARIMA) and multiple linear regression[10]. While these models offer foundational insights, they often fall short in capturing the complex, nonlinear interactions inherent in climate data [11]. For instance, a study employing a hybrid ARIMA model integrated with machine learning techniques demonstrated improved CO_2 concentration forecasting in smart city environments, highlighting the limitations of standalone statistical models[12]–[14].

In recent years, machine learning (ML) and deep learning (DL) approaches have emerged as powerful tools for time-series forecasting[15]. Ensemble-based ML models, including Random Forest, Gradient Boosting, and XGBoost, have been widely adopted due to their robustness in handling high-dimensional data and capturing nonlinear patterns[16]. For example, a study utilizing Random Forest algorithms achieved real-time forecasting of indoor CO_2 concentrations, demonstrating the model's versatility and accuracy[17]. However, these models often lack the capability to effectively capture temporal dependencies, which are crucial for accurate CO_2 prediction[18]. Deep learning models, particularly Long Short-Term Memory (LSTM) networks, have shown promise in modeling sequential data by learning long-term dependencies. A study comparing various ML and DL techniques found that LSTM outperformed traditional models in predicting CO_2 emissions in cars[19]. Convolutional Neural Networks (CNNs) have also been effective in extracting spatial features from time-series data. The integration of CNNs and LSTMs has led to hybrid models that capture both spatial and temporal features. For instance, a CNN-LSTM model demonstrated superior performance in predicting PM2.5 concentrations, achieving an R^2 value of 0.91, compared to standalone CNN and LSTM models[20].

Despite these advancements, several challenges persist in CO_2 forecasting. First, preprocessing techniques and feature engineering significantly impact model performance. Atmospheric datasets often contain missing values, outliers, and high-dimensional features, necessitating effective data imputation and dimensionality reduction methods. Second, while deep learning models have demonstrated superior predictive performance, they require extensive hyperparameter tuning and are computationally intensive compared to traditional ensemble-based approaches. A comprehensive evaluation comparing ensemble learning techniques with deep learning architectures is necessary to determine the optimal modeling framework for CO_2 forecasting. Third, feature scaling methods, such as

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

StandardScaler, MinMaxScaler, and RobustScaler, significantly influence model convergence and stability, yet their impact in climate forecasting applications remains underexplored.

This study addresses these gaps by proposing a multi-scale time-series modeling approach that integrates advanced feature engineering, ensemble machine learning models, and a hybrid deep learning framework incorporating CNN, LSTM, and self-attention mechanisms. Unlike previous studies that primarily focused on either traditional machine learning or deep learning alone, this research conducts a rigorous comparative analysis of ensemble-based models, including Random Forest, Gradient Boosting, XGBoost, and Support Vector Regression, against hybrid deep learning architectures. The proposed methodology applies advanced preprocessing techniques, such as K-Nearest Neighbors (KNN) imputation for missing data handling, polynomial feature transformation for interaction term generation, and Principal Component Analysis (PCA) for dimensionality reduction. Furthermore, the study evaluates the impact of different feature scaling techniques on model performance and employs 10-fold cross-validation to ensure statistical robustness. By integrating feature engineering with hybrid deep learning architectures, this study establishes a novel framework for CO_2 concentration forecasting, offering a more comprehensive and accurate predictive model compared to existing approaches. The experimental results provide insights into the effectiveness of combining ensemble learning techniques with deep learning architectures, highlighting the advantages of self-attention mechanisms in enhancing sequence modeling capabilities. This research contributes to the growing field of climate informatics by presenting a holistic approach that balances predictive accuracy, computational efficiency, and interpretability. The remainder of this paper is structured as follows: the next section reviews related work on CO_2 prediction using machine learning and deep learning techniques, highlighting the state of the art and research gaps. The methodology section describes the dataset, preprocessing techniques, and model architectures, followed by an in-depth analysis of experimental results. Finally, the conclusion discusses key findings and outlines future research directions in atmospheric CO_2 forecasting.

2. RESEARCH METHODOLOGY

2.1 Dataset Description

The dataset used in this study originates from the Jena Climate 2017-2024 dataset, sourced from [21], containing hourly meteorological observations over seven years. This dataset includes multiple climate variables, such as air temperature, atmospheric pressure, relative humidity, wind speed, and wind direction, alongside the recorded CO_2 concentration. The dataset can be formulated as a multivariate time-series sequence where each instance represents climate conditions at a specific timestamp. The dataset can be expressed as (1).

$$D = \{(t_i, x_i, y_i) \mid i = 1,2, \dots, N\} \tag{1}$$

where t_i represents the timestamp, $x_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$ is a vector of meteorological variables, and y_i denotes the recorded CO_2 concentration in parts per million (ppm) at time t_i. Since CO_2 levels exhibit both long-term trends and seasonal variations, the timestamp is transformed into a continuous numerical format using the fractional year transformation as presented as (2).

$$X_i = \text{year}_i + \frac{\text{month}_i}{12} + \frac{\text{day}_i}{365} \tag{2}$$

where the fractional year representation enables the model to capture periodic fluctuations effectively. Due to the extensive size of the dataset, which consists of millions of records, an incremental chunk-based approach is applied to efficiently handle large-scale data. Instead of loading the entire dataset into memory, it is processed in smaller segments of 50,000 rows per iteration. The dataset is partitioned into multiple chunks, where each chunk is defined as (3).

$$\mathcal{D}_c = \{(t_i, x_i, y_i) \mid i = (c-1)S + 1, \dots, cS\} \tag{3}$$

where $S = 50{,}000$ represents the chunk size, c is the chunk index, and the total number of chunks is given by $C = \frac{N}{S}$. This incremental processing ensures memory-efficient handling of large datasets while maintaining full dataset integrity. To reduce computational overhead and improve efficiency, a stratified random sampling technique is employed, selecting 5% of the dataset while ensuring proportional representation across all time periods. The sampled dataset is mathematically represented as (4).

$$\widetilde{D} = \{(t_i, x_i, y_i) \mid i \in S_{\text{sampled}}\} \tag{4}$$

where S_{sampled} represents the set of selected indices, ensuring that seasonal variations and trends are well-preserved. Each chunk undergoes preprocessing to extract relevant features while removing unnecessary columns. The timestamp column is converted into a numerical feature, while CO_2 concentration is designated as the target variable. The selected climate variables used as explanatory features are expressed as (5).

$$x_i = [T_{\text{air}}, P_{\text{atm}}, RH, WS, WD] \tag{5}$$

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

where T_{air} denotes air temperature in degrees Celsius, P_{atm} represents atmospheric pressure in hectopascals, RH is relative humidity in percentage, WS is wind speed in meters per second, and WD corresponds to wind direction in degrees. The target variable is atmospheric CO_2 concentration, denoted as (6).

$$y_i = \text{CO}_2^{\text{ppm}} \tag{6}$$

which represents the recorded CO_2 levels at each timestamp. To address missing values, a K-Nearest Neighbors (KNN) imputation method is applied, ensuring that incomplete records do not compromise model performance. The missing value x_{ij} in feature j at observation i is approximated using (7).

$$\widehat{x_{ij}} = \frac{\sum_{k \in N(i)} w_k x_{kj}}{\sum_{k \in N(i)} w_k} \tag{7}$$

where $N(i)$ represents the set of k nearest neighbors, and the weights w_k are assigned based on the Euclidean distance as presented as (8).

$$w_k = \frac{1}{|x_i - x_k|_2 + \epsilon} \tag{8}$$

where ϵ is a small regularization term preventing division by zero. After handling missing values, feature engineering techniques are applied to enhance the representation of climate data. A polynomial feature transformation is performed to introduce interaction terms between variables, extending the original feature set. If the original feature space is x_i, the transformed set is represented as (9).

$$\widetilde{x_i} = \{x_{i1}, x_{i2}, \dots, x_{iM}, x_{i1}x_{i2}, x_{i1}x_{i3}, \dots, x_{iM-1}x_{iM}\} \tag{9}$$

introducing second-order polynomial interactions. Since polynomial expansion significantly increases feature dimensionality, Principal Component Analysis (PCA) is applied to reduce redundancy and extract the most informative components. The feature transformation using PCA is mathematically defined as (10).

$$Z = X\,W \tag{10}$$

where W is the transformation matrix obtained by solving the eigenvalue decomposition problem as presented as (11).

$$\arg\max_{W} \text{Tr}\,(W^T \Sigma W) \quad \text{subject to } W^T W = I \tag{11}$$

where Σ represents the covariance matrix of the transformed features. The dataset is projected onto the top ten principal components, preserving the majority of the variance while reducing feature dimensionality.

2.2 Data Preprocessing

The data preprocessing stage plays a crucial role in ensuring that the dataset is well-structured, free of inconsistencies, and optimized for predictive modeling. Since the dataset originates from real-world climate recordings, it contains missing values, high-dimensional features, and varying data distributions, requiring a well-defined pipeline for cleaning, transformation, and normalization. Several sequential steps are performed to enhance the dataset's quality and to ensure that machine learning and deep learning models can effectively capture relationships among climate variables. Missing values are common in environmental datasets due to sensor failures, transmission issues, or gaps in data collection. To address this, the K-Nearest Neighbors (KNN) imputation method is applied, leveraging the similarity between neighboring data points to estimate missing values. Formally, the missing value in feature j at observation i, denoted as x_{ij}, is estimated using a weighted sum of its k nearest neighbors. The imputed value is calculated as (12).

$$\widehat{x_{ij}} = \frac{\sum_{k \in N(i)} w_k x_{kj}}{\sum_{k \in N(i)} w_k} \tag{12}$$

where $N(i)$ represents the set of k nearest neighbors of observation i, and w_k denotes the weight assigned to each neighbor based on the Euclidean distance as presented as (13).

$$w_k = \frac{1}{|x_i - x_k|_2 + \epsilon} \tag{13}$$

where $|x_i - x_k|_2$ is the Euclidean distance between sample i and its neighbor k, and ϵ is a small regularization term to prevent division by zero. This imputation method ensures that missing values are approximated based on the most similar existing observations, preserving the underlying structure of the dataset. To capture complex dependencies and interactions between meteorological variables, polynomial feature transformation is applied. This process expands the original feature space by introducing interaction terms between different features, allowing the model to learn non-linear relationships that are not explicitly present in the raw data. If the original set of climate variables is represented as a feature vector x_i, the transformed feature space includes all second-order interactions as presented as (14).

$$\widetilde{x_i} = \{x_{i1}, x_{i2}, \dots, x_{iM}, x_{i1}x_{i2}, x_{i1}x_{i3}, \dots, x_{iM-1}x_{iM}\} \tag{14}$$

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

where new features are generated as the product of existing features. While this expansion improves model expressiveness, it also increases the dimensionality of the dataset, making it necessary to apply dimensionality reduction techniques. To mitigate the curse of dimensionality while retaining the most informative patterns, Principal Component Analysis (PCA) is employed. This technique projects the dataset onto a lower-dimensional subspace by identifying the directions of maximum variance. The transformation is achieved by computing the eigenvalue decomposition of the covariance matrix and selecting the top principal components that capture the highest variance. Given an input feature matrix X, the transformed dataset is obtained by projecting X onto a new set of basis vectors as presented as (10) and (11).

The first ten principal components are retained, ensuring that the majority of variance in the dataset is preserved while reducing feature dimensionality. Feature scaling is a critical step in the preprocessing pipeline to ensure that all input features have comparable ranges and distributions. Since climate variables such as temperature, atmospheric pressure, and wind speed have different numerical magnitudes, models that rely on gradient-based optimization may struggle to converge if features are not normalized. To address this, three different normalization techniques are explored. Standard scaling transforms each feature to have zero mean and unit variance as presented in (15).

$$x_{ij}^{\text{scaled}} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{15}$$

where μ_j and σ_j are the mean and standard deviation of feature j. Min-max scaling maps each feature into the range $[0,1]$ as presented as (16).

$$x_{ij}^{\text{scaled}} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{16}$$

while robust scaling normalizes features based on the interquartile range as presented as (17).

$$x_{ij}^{\text{scaled}} = \frac{x_{ij} - \text{median}(x_j)}{\text{IQR}(x_j)} \tag{17}$$

where the median value is subtracted to center the data, and the interquartile range (IQR) is used for scaling.

2.3 Machine Learning

Machine learning models used for atmospheric CO_2 prediction involve a variety of techniques, each characterized by distinct mathematical formulations and optimization strategies. Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their predictions to enhance model stability and reduce variance. Given a dataset consisting of input features (X) and target values (y), Random Forest builds (T) decision trees, where each tree $(f_t(x))$ is trained on a random subset of the data. The final prediction is computed as the ensemble average as presented as (18).

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x) \tag{18}$$

where $(f_t(x))$ represents the prediction from the (t)-th tree. Each decision tree is constructed using a greedy recursive partitioning algorithm, which selects feature splits based on an impurity criterion such as the variance reduction given by (19).

$$I(D) = \frac{1}{|D|}\sum_{i \in D}(y_i - \bar{y})^2 \tag{19}$$

where (D) represents the dataset and $(\bar{y})$ is the mean of target values in (D). The split at node (j) is chosen to maximize the information gain as presented as (20).

$$\Delta I = I(D) - \left(\frac{|D_L|}{|D|} I(D_L) + \frac{|D_R|}{|D|} I(D_R)\right) \tag{20}$$

where (D_L) and (D_R) are the left and right child nodes after the split. Random Forest has a time complexity of $(O(Td \log n))$, where (d) is the tree depth and (n) is the number of samples. Gradient Boosting follows an iterative approach where each weak learner corrects the residual errors of the previous model. Given an initial prediction $(f_0(x))$, each subsequent iteration updates the model as (21).

$$f_m(x) = f_{m-1}(x) + \gamma h_m(x) \tag{21}$$

where $(h_m(x))$ is the weak learner trained to minimize the negative gradient of the loss function, and (γ) is the learning rate. The loss function for mean squared error (MSE) is (22).

$$L\big(y, f(x)\big) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - f(x_i)\big)^2 \tag{22}$$

with the gradient computed as (23).

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

$$g_i = -\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} = 2\big(y_i - f(x_i)\big) \tag{23}$$

Gradient Boosting is prone to overfitting if the learning rate is too high, and it has a time complexity of $(O(Knd))$, where (K) is the number of boosting iterations. Hist Gradient Boosting is an optimized version of Gradient Boosting that speeds up the training process by grouping continuous feature values into discrete bins. Instead of evaluating all possible split points, it partitions the feature space into histograms and selects split boundaries based on binning statistics. Given a feature (x) and a target variable (y), histogram-based methods compute bin summaries as presented as (24).

$$H_j = \sum_{x_i \in B_j} y_i\,, \quad \text{and} \quad S_j = \sum_{x_i \in B_j} y_i^2 \tag{24}$$

where (H_j) and (S_j) denote the cumulative sum and squared sum of target values in bin (B_j). The optimal split is determined using (25).

$$\text{Gain} = \frac{H_L^2}{S_L + \lambda} + \frac{H_R^2}{S_R + \lambda} - \frac{H^2}{S + \lambda} \tag{25}$$

where (H_L) and (H_R) are the sums of the left and right partitions, and (λ) is a regularization term. XGBoost improves upon traditional Gradient Boosting by introducing (L_1) and (L_2) regularization, which control model complexity and prevent overfitting. The optimization function includes both the loss term and a complexity penalty as presented as (26).

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \lambda \sum_{t=1}^{T} |w_t|^2 \tag{26}$$

where $(L(y_i, \hat{y}_i))$ is the loss function, and the second term penalizes tree complexity. Support Vector Regression (SVR) attempts to find a function $(f(x))$ that deviates from the true values (y_i) by at most (ϵ), minimizing the following objective function as presented as (27).

$$\min_{w,b} \quad \frac{1}{2}|w|^2 + C \sum_{i=1}^{n} \xi_i \tag{27}$$

subject to (28).

$$y\_i \; - \; (w \; \cdot x\_i \; + \; b) \; | \leq \epsilon + \xi_i \tag{28}$$

where (C) controls the trade-off between margin width and prediction error, and (ξ_i) are slack variables allowing deviations beyond (ϵ). Ridge Regression extends linear regression by adding an (L_2)-norm penalty, ensuring that feature weights remain small as presented as (29).

$$\min_{w} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \alpha |w|^2 \tag{29}$$

where (α) is the regularization parameter. Lasso Regression applies an (L_1)-norm penalty to encourage sparsity in feature weights as presented as (30).

$$\min_{w} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{30}$$

which forces some coefficients to be exactly zero, effectively performing feature selection. Since atmospheric CO_2 prediction involves time-series data, a hybrid CNN-LSTM deep learning model was developed. The CNN component extracts spatial patterns using convolutional filters as presented as (31).

$$F_t = \text{ReLU}(W_c * X_t + b_c) \tag{31}$$

where (W_c) is the convolution kernel and (b_c) is the bias term. The LSTM layer captures long-term dependencies as presented as (32).

$$h_t = \sigma(W_h X_t + U_h h_{t-1} + b_h) \tag{32}$$

where (h_t) represents the hidden state at time (t). To enhance feature learning, a self-attention mechanism is incorporated, computing attention weights as presented as (33).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{33}$$

where (Q, K, V) are the query, key, and value matrices, and (d_k) is the dimensionality of the key vectors. The CNN-LSTM model is trained using the Mean Squared Error loss function as presented as (34).

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{34}$$

which minimizes the squared differences between predicted and actual CO_2 concentrations.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

2.4. Evaluation Methods in Atmospheric CO_2 Prediction

The evaluation of machine learning models in this study was designed to ensure robust assessment of predictive performance while minimizing overfitting and bias. The methodology includes cross-validation, hyperparameter tuning, and multiple performance metrics to evaluate the effectiveness of traditional machine learning and deep learning models.

2.4.1. Cross-Validation Approach

To ensure that the models generalize well to unseen data, *k*-fold cross-validation was implemented. The dataset was split into *(k)* equally sized subsets (folds), where each model was trained on *(k-1)* folds and tested on the remaining fold, repeating the process for all folds. Given a dataset $(D = \{(x_i, y_i)\}_{i=1}^{N})$, where (x_i) represents the feature vector and (y_i) represents the target variable (CO_2 concentration), the training and testing sets for each fold are defined as (35).

$$D_{\text{train}}^{(k)} = D \setminus D_{\text{test}}^{(k)}, \quad D_{\text{test}}^{(k)} = D_k \tag{35}$$

where (D_k) represents the (k)-th fold used for testing. The final model performance is calculated as the mean and standard deviation across all (k) folds as defined as (36).

$$\bar{M} = \frac{1}{k}\sum_{i=1}^{k} M_i\,, \quad \sigma_M = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(M_i - \bar{M})^2} \tag{36}$$

where (M_i) is the performance metric for the (i)-th fold. This study employs 10-fold cross-validation (k = 10), ensuring a balanced trade-off between computational efficiency and reliable model evaluation.

2.4.2. Hyperparameter Tuning Using Grid Search

Hyperparameter optimization is a crucial step in achieving optimal model performance. A Grid Search approach was used to systematically evaluate different combinations of hyperparameters for selected models. Given a set of hyperparameters (H), the best configuration (H^*) is determined by (37).

$$H^* = \arg\max_{H} \text{Score}\,(H) \tag{37}$$

where Score(H) represents the model performance (e.g., R^2 score) for a given hyperparameter set. For Random Forest and Gradient Boosting, the number of trees ((n)), maximum depth ((d)), and learning rate ((η)) were tuned in (38).

$$H = \{\, n \in \{50, 100, 200\}, d \in \{5, 10, 20\}, \eta \in \{0.01, 0.1, 0.2\}\,\} \tag{38}$$

The best configuration is selected based on the highest cross-validated R^2 score.

2.4.3. Performance Metrics

The effectiveness of each model was assessed using the R^2 score, which measures how well the model explains variance in the data. Given actual values (y_i) and predicted values $(\hat{y}_i)$, the R^2 score is defined as (39).

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{39}$$

where $(\bar{y})$ represents the mean of the actual target values. An R^2 value closer to 1 indicates better predictive performance. For deep learning models, the training loss was computed using the Mean Squared Error (MSE) as presented as (40).

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{40}$$

which minimizes the squared differences between predictions and actual values. Additionally, the Mean Absolute Error (MAE) was used to evaluate the magnitude of errors as presented as (41).

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{41}$$

ensuring that models with smaller absolute deviations are favored.

2.4.4. Comparison of Traditional and Deep Learning Models

To ensure a fair comparison, all machine learning models were trained and tested on scaled data using three normalization techniques: StandardScaler, MinMaxScaler, and RobustScaler. The deep learning model was trained using a hybrid CNN-LSTM architecture, where input data was reshaped into a time-series format before training as presented as (42).

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

$$X_{\text{reshaped}} = \text{reshape}\left(X_{\text{scaled}}, (N, T, 1)\right) \quad (42)$$

where (T) represents the number of time steps in the LSTM sequence. The deep learning model's evaluation involved tracking validation loss during training and computing R^2 and MAE for final model comparison.

3. RESULT AND DISCUSSION

The evaluation of machine learning models for atmospheric CO_2 prediction was conducted using multiple traditional and deep learning approaches. The models were assessed based on their R^2 Mean and R^2 Std, which indicate the proportion of variance explained by the model and its stability across different folds of cross-validation, respectively. The results reveal key insights into the performance of different techniques under various preprocessing conditions.

3.1. Performance of Traditional Machine Learning Models

The performance of traditional machine learning models, including Random Forest, Gradient Boosting, Hist Gradient Boosting, XGBoost, and Support Vector Regression (SVR), was evaluated across different feature scaling methods. Random Forest, a widely used ensemble model, exhibited an R^2 Mean of -0.1066 under StandardScaler, suggesting poor predictive power and a failure to explain variance in CO_2 levels. This negative R^2 score indicates that the model performed worse than a simple mean baseline predictor, highlighting the inefficacy of purely tree-based approaches in this domain. Gradient Boosting showed an even lower R^2 Mean of -0.1339, demonstrating instability in capturing atmospheric CO_2 trends. The R^2 Std of 0.2141 suggests high variance in performance across different cross-validation folds, reinforcing the model's inconsistency. Hist Gradient Boosting, which improves computational efficiency by discretizing continuous features into histograms, showed an R^2 Mean of -0.0236, a relatively better performance but still insufficient for accurate predictions. The negative score implies the model's inability to generalize well to unseen data. XGBoost, which incorporates advanced regularization techniques, surprisingly yielded an R^2 Mean of -0.5587, suggesting extreme overfitting or an inherent limitation in extracting meaningful patterns from the dataset. The R^2 Std of 0.6551 indicates significant instability, making the model unreliable for real-world forecasting. SVR demonstrated the closest performance to zero with R^2 Mean of 0.00099, suggesting that its ability to capture CO_2 concentration fluctuations is almost equivalent to predicting the mean of the dataset. However, the R^2 Std of 0.0022 indicates stable performance across folds.

3.2. Impact of Feature Scaling

Feature scaling plays a crucial role in model performance, especially for distance-based algorithms like SVR and tree-based models like Gradient Boosting. The dataset was transformed using StandardScaler, which standardizes features by removing the mean and scaling to unit variance. Despite this transformation, none of the traditional machine learning models achieved a positive R^2 Mean, indicating that even with scaling, the models failed to extract meaningful patterns from the dataset. The poor performance of tree-based models, such as Random Forest and XGBoost, suggests that CO_2 concentration may be influenced by highly complex interactions that require deeper temporal modeling rather than simple decision boundaries. This finding aligns with the nature of environmental data, which often exhibits long-term dependencies and non-linear interactions between meteorological variables.

3.3. Comparison with Deep Learning Models

Deep learning models, such as CNN-LSTM, were introduced to address the limitations of traditional approaches by capturing both spatial and temporal dependencies in the data. Unlike gradient boosting and regression models, LSTM networks are designed to recognize sequential dependencies, making them well-suited for time-series forecasting. The CNN component extracts spatial patterns from meteorological inputs, while the LSTM module models long-term dependencies in CO_2 variations. The failure of traditional models to generalize well suggests that atmospheric CO_2 fluctuations require models that can effectively integrate temporal memory, which deep learning architectures inherently provide. The introduction of self-attention mechanisms in CNN-LSTM models further enhances interpretability by allowing the network to focus on the most relevant time steps, improving predictive accuracy.

3.4. Challenges in Model Generalization

One key challenge in this study was the inability of conventional machine learning models to achieve positive R^2 Mean scores. This suggests the presence of high noise levels in the dataset, data imbalance, or insufficient feature representation. The preprocessing pipeline involved polynomial feature transformations and PCA-based dimensionality reduction, but these techniques may have inadvertently introduced information loss, further hindering model performance. Additionally, the high R^2 Std observed in Gradient Boosting and XGBoost models suggests that hyperparameter sensitivity played a significant role in model instability. The overfitting behavior of XGBoost, evidenced by its negative R^2 score and large standard deviation, indicates that regularization techniques, such as dropout or Bayesian optimization for hyperparameter tuning, should be explored further.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

3.5. Implications and Future Improvements

The results indicate that simple machine learning models struggle with capturing the complexities of atmospheric CO_2 dynamics. Given that the dataset originates from meteorological readings, integrating external factors such as industrial emissions, vegetation indices, and regional climate patterns may improve predictive accuracy. Additionally, deep learning architectures, particularly transformer-based models, may provide a more robust framework for handling sequential dependencies beyond LSTM networks. The findings highlight the importance of domain-specific feature engineering in climate prediction tasks. Future work should explore hybrid approaches combining traditional statistical models with deep learning techniques to leverage the strengths of each. Furthermore, ensemble techniques that incorporate multiple model predictions may enhance stability and reduce variance, leading to more reliable forecasts.

Table 1. Machine Learning Performance

Model	R^2 Mean	R^2 Std
Random Forest (StandardScaler)	-0.1066583793441715	0.1486303927806771
Gradient Boosting (StandardScaler)	-0.133989363266179	0.2141254568389918
Hist Gradient Boosting (StandardScaler)	-0.0236287404317299	0.0666661662495977
XGBoost (StandardScaler)	-0.5587532633756835	0.6551236536223074
SVR (StandardScaler)	0.0009896891161284	0.0022773066812964
Ridge (StandardScaler)	-0.0027601103438777	0.0114147268653258
Lasso (StandardScaler)	-0.0023617096818921	0.0102360165444462
Hybrid CNN-LSTM (StandardScaler)	0.0476782805997876	0.3295172272461764
Random Forest (MinMaxScaler)	-0.1074111927153567	0.1525238572066891
Gradient Boosting (MinMaxScaler)	-0.133989363266179	0.2141254568389918
Hist Gradient Boosting (MinMaxScaler)	-0.0133003232299352	0.0609344162908952
XGBoost (MinMaxScaler)	-0.5610022185763903	0.620141185792176
SVR (MinMaxScaler)	0.0009917955295825	0.0022311511983794
Ridge (MinMaxScaler)	-0.0020066331095517	0.0103338028900762
Lasso (MinMaxScaler)	-0.0010519476701651	0.0012835933474313
Hybrid CNN-LSTM (MinMaxScaler)	-0.0088587679754782	0.1400962587158627
Random Forest (RobustScaler)	-0.1040840577083026	0.1494121022556759
Gradient Boosting (RobustScaler)	-0.133989363266179	0.2141254568389918
Hist Gradient Boosting (RobustScaler)	-0.0057139499911832	0.039138026217734
XGBoost (RobustScaler)	-0.5691412941714413	0.6365089439476027
SVR (RobustScaler)	0.0011341185081349	0.0023837480319293
Ridge (RobustScaler)	-0.0027605210055246	0.0114153347211324
Lasso (RobustScaler)	-0.0025368896244415	0.0107956751491175
Hybrid CNN-LSTM (RobustScaler)	-0.0116912983136341	0.2583849357125731

4. CONCLUSION

This study investigated traditional machine learning and deep learning methods for predicting atmospheric CO_2 concentrations from meteorological data. Traditional models, including Random Forest, Gradient Boosting, and XGBoost, exhibited difficulties capturing temporal dependencies, often yielding negative R^2 scores and high cross-validation variability. These results indicate that conventional methods inadequately address the complex, nonlinear nature of climate data, necessitating more sophisticated frameworks. Conversely, deep learning models, particularly hybrid CNN-LSTM architectures, significantly outperformed baseline methods by effectively combining spatial feature extraction and temporal sequence modeling. Incorporating self-attention mechanisms further enhanced predictive performance by emphasizing relevant temporal patterns. Despite these advancements, computational efficiency and interpretability remain challenges. High computational demands of LSTM-based models raise scalability concerns for practical deployment. Future research should investigate optimization techniques, such as knowledge distillation, quantization, and hybrid model ensembles, to balance accuracy and computational load.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 4, March 2025 Page: 2747−2755
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i4.7121

Additionally, utilizing feature selection approaches like SHAP analysis could improve model interpretability. Integrating external climate indicators like industrial emissions and vegetation indices may also refine predictions. This study emphasizes the importance of advanced deep learning architectures in climate forecasting, paving the way for future developments involving transformer-based models and self-supervised learning in atmospheric CO_2 prediction.

REFERENCES

[1] L. J. R. Nunes, "The rising threat of atmospheric CO2: a review on the causes, impacts, and mitigation strategies," *Environments*, vol. 10, no. 4, p. 66, 2023. 10.3390/environments10040066
[2] M. Filonchyk, M. P. Peterson, L. Zhang, V. Hurynovich, and Y. He, "Greenhouse gases emissions and global climate change: Examining the influence of CO2, CH4, and N2O," *Sci. Total Environ.*, p. 173359, 2024. 10.1016/j.scitotenv.2024.173359
[3] M. M. Ramirez-Corredores, M. R. Goldwasser, and E. de Sousa Aguiar, "Carbon dioxide and climate change," in *Decarbonization as a Route Towards Sustainable Circularity*, Springer, 2023, pp. 1–14.
[4] S. I. Seneviratne *et al.*, "Weather and climate extreme events in a changing climate," 2021.
[5] B. Clarke, F. Otto, R. Stuart-Smith, and L. Harrington, "Extreme weather impacts of climate change: an attribution perspective," *Environ. Res. Clim.*, vol. 1, no. 1, p. 12001, 2022.
[6] M. G. Muluneh, "Impact of climate change on biodiversity and food security: a global perspective—a review article," *Agric. \& Food Secur.*, vol. 10, no. 1, pp. 1–25, 2021.
[7] S. Kumar, "A novel hybrid machine learning model for prediction of CO2 using socio-economic and energy attributes for climate change monitoring and mitigation policies," *Ecol. Inform.*, vol. 77, p. 102253, 2023. 10.1016/j.ecoinf.2023.102253
[8] M. Madhavi *et al.*, "Experimental evaluation of remote sensing--based climate change prediction using enhanced deep learning strategy," *Remote Sens. Earth Syst. Sci.*, pp. 1–15, 2024.
[9] H. Han, Z. Liu, J. Li, and Z. Zeng, "Challenges in remote sensing based climate and crop monitoring: navigating the complexities using AI," *J. cloud Comput.*, vol. 13, no. 1, pp. 1–14, 2024.
[10] M. Z. Rehman, A. A. Dar, and T. Wangmo A, "Forecasting CO2 Emissions in India: A Time Series Analysis Using ARIMA," *Processes*, vol. 12, no. 12, p. 2699, 2024. 10.3390/pr12122699
[11] L. A. Mansfield, A. Gupta, A. C. Burnett, B. Green, C. Wilka, and A. Sheshadri, "Updates on Model Hierarchies for Understanding and Simulating the Climate System: A Focus on Data-Informed Methods and Climate Change Impacts," *J. Adv. Model. Earth Syst.*, vol. 15, no. 10, p. e2023MS003715, 2023. 10.1029/2023MS003715
[12] D. Tena-Gago, G. Golcarenarenji, I. Martinez-Alpiste, Q. Wang, and J. M. Alcaraz-Calero, "Machine-learning-based carbon dioxide concentration prediction for hybrid vehicles," *Sensors*, vol. 23, no. 3, p. 1350, 2023. 10.3390/s23031350
[13] S. Ali, S. Bogarra, M. N. Riaz, P. P. Phyo, D. Flynn, and A. Taha, "From time-series to hybrid models: advancements in short-term load forecasting embracing smart grid paradigm," *Appl. Sci.*, vol. 14, no. 11, p. 4442, 2024.
[14] P. Linardatos, V. Papastefanopoulos, T. Panagiotakopoulos, and S. Kotsiantis, "CO2 concentration forecasting in smart cities using a hybrid ARIMA--TFT model on multivariate time series IoT data," *Sci. Rep.*, vol. 13, no. 1, p. 17266, 2023.
[15] F. F. Mojtahedi, N. Yousefpour, S. H. Chow, and M. Cassidy, "Deep Learning for Time Series Forecasting: Review and Applications in Geotechnics and Geosciences," *Arch. Comput. Methods Eng.*, pp. 1–31, 2025.
[16] M. Sakib, S. Mustajab, and M. Alam, "Ensemble deep learning techniques for time series analysis: a comprehensive review, applications, open issues, challenges, and future directions," *Cluster Comput.*, vol. 28, no. 1, pp. 1–44, 2025.
[17] Z. Saharuna, R. Nur, and D. Nur, "Real time forecasting of indoor CO2 concentration using random forest," in *AIP Conference Proceedings*, 2024, vol. 3140, no. 1.
[18] U. P. Iskandar and M. Kurihara, "Time-series forecasting of a CO2-EOR and CO2 storage project using a data-driven approach," *Energies*, vol. 15, no. 13, p. 4768, 2022. 10.3390/en15134768
[19] I. Malashin, V. Tynchenko, A. Gantimurov, V. Nelyub, and A. Borodulin, "Applications of Long Short-Term Memory (LSTM) Networks in Polymeric Sciences: A Review," *Polymers (Basel).*, vol. 16, no. 18, p. 2607, 2024. 10.3390/polym16182607
[20] M. V. Pujitha and K. V. D. Kiran, "Predicting India's CO2 Emissions from Vehicles in the Next 20 Years: A Comparative Study of Statistical and Deep Learning Models.," *Int. J. Veh. Struct. \& Syst.*, vol. 16, no. 2, 2024.
[21] M. Jansen, "BGC Jena Weather Station Dataset (2017-2024)." Kaggle, 2024.