Design Of A Carbon Emission Monitoring And Prediction System Based On Big Data

Wenyu Jiang

School of Statistics, Beijing Normal University, Beijing, 100875, China

Corresponding author. E-mail: weny107@126.com

Received: Feb. 14, 2026; Accepted: Mar. 30, 2026

Accurate monitoring and prediction of carbon emissions has become a key need by industries across the globe due to the growing need for sustainable development. Industrial activities generate large volumes of energy consumption data, creating opportunities for real-time monitoring and predictive analytics using big data technologies. A carbon emission monitoring and prediction systems is developed that integrates real-time electricity consumption data with external variables such as industry type, energy mix, and economic indicators. Big data platforms including Apache Hadoop and Apache Spark support large-scale data ingestion, storage, and processing. Carbon emission prediction is performed using a hybrid model that combines Long Short-Term Memory (LSTM) networks for temporal pattern learning with Extreme Gradient Boosting (XGBoost) to capture feature-based relationships. Data preprocessing techniques such as normalization, feature engineering, and missing value imputation improve data quality and model reliability. The dataset consists of large-scale industrial energy consumption and carbon emission records collected at an hourly resolution, comprising approximately 10,000 samples. The data is divided into training and testing sets using an 80:20 split. The LSTM model is configured with two layers and 128 hidden units, using a learning rate of 0.001 with the Adam optimizer. The XGBoost model employs 100 estimators with a maximum depth of 6 and regularization parameters ($\lambda = 1.0, \gamma = 0.1$). Experimental evaluation shows that the hybrid LSTM-XGBoost model outperforms alternative approaches including CNN-LSTM-BERT and AdaBoost, achieving MAE = 0.0234, MSE = 0.00086, and $R^2 = 0.963$. The framework supports real-time carbon emission monitoring and forecasting, providing a reliable tool for data-driven industrial emission management and sustainable decision-making.

Keywords: Carbon Emission Monitoring, Big Data, LSTM, XGBoost, Sustainability

http://dx.doi.org/10.6180/jase.202609_32.042

1. Introduction

Carbon emission reduction has become an important objective in the world in view of the increasing pressure on the need to implement sustainable development [1]. Proper monitoring and predicting systems of carbon emissions will be critical towards helping industries to trim down their carbon footprints and meet the standard requirements [2]. When it comes to traditional ways of carbon emission monitoring [3], they tend to be based on the principles of the collection of data as fixed and periodic [4], thus restricting the opportunities to access real-time insights and make accurate predictions of future emissions [5]. Traditional statistical inventory approaches for carbon emission monitoring rely on fixed emission coefficients and periodic data collection, limiting their capacity for high-frequency, real-time prediction and their ability to capture non-linear interactions among industrial and external factors. Hybrid deep learning–ensemble methods, which integrate LSTM networks with XGBoost models, provide improved perfor-

mance by modelling temporal dependencies in time-series data and incorporating complex feature interactions. The enterprise-level carbon emission forecasting that simultaneously addresses temporal dynamics and multi-feature interactions remains underdeveloped. Since energy usage is directly related to carbon emissions, a new high-tech solution that utilises big data and artificial intelligence (AI) is required to enhance prediction and offer timely, practical recommendations [6]. In this framework, the proposal is a combined carbon emission monitoring and prediction system [7] that encompasses big data and AI that fills the requirement of accurate, real-time monitoring and forecasting to aid in decision-making in industries [8]. Different techniques have been suggested concerning the monitoring and prediction of carbon emission [9]. They are the IPCC inventory accounting method which computes the emissions according to the energy consumption statistics and the big data approach according to which historical consumption data is processed with the help of machine learning algorithms [10]. These approaches can give useful information but are constrained by long-time delays in the processing of data [11], poor granularity of time and space resolution, and the use of fixed emissions coefficients [12]. Moreover, other models like linear regression and neural networks [13] can tend to fail to capture non-linearities and time dependence in the emission data and hence do not perform well in dynamic and real-world conditions [14]. These disadvantages are obstacles to the adoption of proper carbon-cutting policies in the enterprise level [15].

The framework combines LSTM for time-series prediction and XGBoost to enhance accuracy using factors like industry type, energy mix, and external conditions. Integrated with Apache Kafka and Spark, it enables real-time big data processing, delivering high-frequency, accurate forecasts that help industries adapt energy use and reduce carbon emissions for sustainability goals.

1.1. Research Objectives

- Consider the general purpose of the work that is to design and develop a real-time carbon emission monitoring and predicting system using big data and AI-powered hybrid designs.
- Determine the data to be utilised in the proposed framework that consists of real-time electricity usage data and carbon emission information of enterprises of different industries.
- Use LSTM to forecast emissions using time series to predict future carbon emissions, relying on the historical consumption trends.
- Add XGBoost to optimise the forecasts with industry-specifics and energy mix among other external variables.

1.2. Research Scope and Novelty

This study develops a real-time system for monitoring and predicting carbon emissions using big data and a hybrid LSTM–XGBoost model. By analyzing enterprise electricity consumption, it delivers accurate, high-frequency forecasts. With optimized feature selection and scalability, the framework adapts to evolving trends, enabling proactive carbon management, improved sustainability decisions, and effective compliance with environmental regulations.

1.3. Research Organization

The paper is structured in the following way:

- Section 2 gives an elaborate literature review of the current carbon emission monitoring applications and AI-based applications.
- Section 3 provides the methodology of the proposed carbon emission monitoring and prediction system, such as data collection, preprocessing, and hybrid AI.
- Section 4 provides the results and discussion, which includes the analysis of the performance of the system in comparison with the traditional approaches.
- Section 5 is the conclusion of the paper, where implications, limitations, and suggestions to research in the future can be found.
- The references utilised in this research are provided in **References**.

2. Literature review

Zhou et al. [16] The paper monitors real-time carbon emissions using electricity data, building CO2 accounting via ratios, but lacks advanced machine learning forecasting. Almanasra [17] The paper reviews AI and big data for analyzing and reducing transportation carbon emissions but lacks a real-time predictive framework. Peng et al. [18] The study addresses carbon emission reduction in transport using smart navigation, genetic algorithms, and particle-swarm-optimization for low-carbon path planning. Xu and al. [19] The article reviews spatial modeling of carbon emissions using hybrid models and data-driven approaches, lacking real-time monitoring and prediction. Rubio-Loyola and Paul-Fils [20] The article discusses using machine learning models to predict black carbon in industrial furnaces, but lacks real-time scalability. Udoh et al. [21] The article

models CO2 emissions of light-duty vehicles using machine learning and WLTP sensor data, focusing on laboratory testing.

Zhou et al. [22] The paper suggests real-time carbon emission monitoring using electricity data and input-output analysis, but lacks deep learning models. Fay et al. [23] The study explores low-power microcontroller-based event detection in carbon emission reduction, using binary classification models with low precision. Khan et al. [24] This paper explores smart city technologies to reduce energy usage and carbon emissions using machine learning models and SHAP analysis. Wang and al. [25] The system uses BIM, IoT, and real-time energy monitoring for visualization and decision-making, but lacks predictive AI modelling. Bicamumakuba et al. [26] This paper explores smart greenhouse management for sustainable agriculture using IoT, AI, and data filtering, lacking emission forecasting focus. Alhussan and Metwally [27] This study predicts CO2 emissions in light-duty vehicles using optimized machine learning, feature selection, and hyperparameter tuning via metaheuristics.

Current carbon emission monitoring methods rely on static data and focus more on monitoring than prediction, limiting their application in dynamic industrial settings. Many models use statistical coefficients or general machine learning techniques, unable to capture complex, nonlinear relationships and long-term dependencies. Additionally, they lack scalability and real-time flexibility. The proposed framework combines real-time big data with a hybrid LSTM-XGBoost model to provide scalable, adaptive, and high-confidence carbon emission prediction for multi-industry applications.

3. Proposed intelligent big data architecture for carbon emission forecasting using hybrid deep learning methodology

The carbon emission monitoring and prediction system collects large datasets on electricity consumption, carbon emissions, and external factors like industry type and economic indicators. Data is processed using big data tools like Hadoop and Spark. A hybrid AI approach combines LSTM networks for time-series predictions and XGBoost to integrate additional factors. The system revises predictions based on real-time data, improving accuracy. It provides actionable insights for energy efficiency and emission reduction.

The Fig. 1 shows the entire workflow of the proposed system of carbon emission monitoring and prediction. Data is collected from sources like energy meters, industrial systems, weather stations, and greenhouse gas networks. It's preprocessed, including handling missing data, normalization, feature engineering, and aggregation, then integrated into big data frameworks like Hadoop and Spark for real-time streaming.

3.1. Dataset Description

The dataset consists of large-scale real-time electricity consumption records collected from industrial enterprises representing multiple industry sectors [28]. The dataset tracks carbon emissions from energy use, integrating consumption patterns, economic indicators, energy policies, and geographical data for accurate predictions. With high-resolution time data (hourly/daily) and real-time updates, it supports big data tools like Apache Kafka and Spark, enabling LSTM and XGBoost models for precise emission forecasts. The characteristics of the dataset provide a suitable foundation for carbon emission forecasting. To ensure data consistency and reliability for predictive modelling, several preprocessing steps are applied as described in the following subsection.

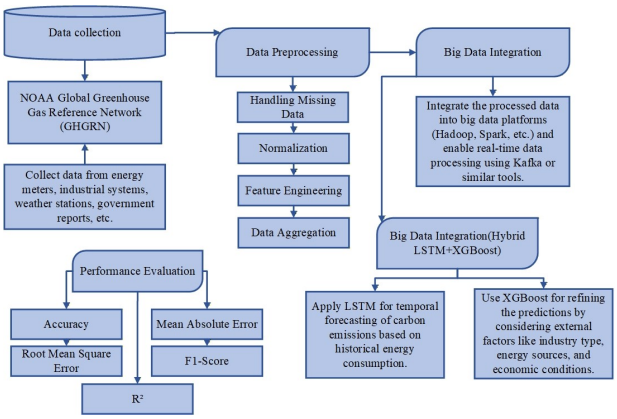

Fig. 1. Architecture of the Proposed Big Data–Driven Carbon Emission Monitoring and Prediction System.

3.2. Data Preprocessing

Data preprocessing is performed to prepare the collected dataset for machine learning and predictive modelling. The dataset in the proposed carbon emission monitoring and prediction system is pre-processed by following the steps below:

Handling Missing Data: The deep learning-based imputation such as Autoencoders or K-Nearest Neighbors imputation are employed to deal with missing values in the dataset is represented in Eq. (1):

$$X_{\text{imputed}} = \frac{1}{k} \sum_{i=1}^{k} X_i \tag{1}$$

where X_{imputed} represents the imputed value, calculated as the average of the k-nearest neighbor values, denoted by X_i, where i ranges from 1 to k.

Missing values in the dataset were addressed using Autoencoder or KNN based imputation techniques, where missing entries are estimated by learning underlying feature relationships or by identifying similar instances within the dataset. Data normalisation was performed using Zscore standardisation, which scales features by subtracting the mean and dividing by the standard deviation to ensure consistent feature distribution during model training. Categorical attributes were transformed into numerical representations using One-Hot Encoding or Label Encoding, depending on the nature of the feature. The dataset was aggregated using consistent temporal intervals such as hourly or daily resolution to maintain uniformity in time-series analysis and to capture meaningful patterns in electricity consumption and carbon emission data.

Normalization: The features are normalized so that they get on to the same scale so that the model does not favor features that have larger ranges is represented in Eq. (2):

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{2}$$

where $\boldsymbol{X}_{\text{scaled}}$ represents the normalized value of the feature, $\boldsymbol{X}$ denotes the original feature value, X_{min} indicates the minimum value of the feature in the dataset, and X_{max} represents the maximum value of the feature. This normalization process scales the feature values to a range between 0 and 1 , ensuring consistent input distribution for the learning model.

Z-Score Standardization (Standard Scaling) is represented in Eq. (3):

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma} \tag{3}$$

where X_{scaled} denotes the standardized value of the feature, X represents the original data value, μ indicates the mean of the feature values in the dataset, and σ denotes the standard deviation.

Feature Engineering

- Temporal Features: Miner feature like hour of the day, day of the week, and seasonality (e.g., quarterly, monthly), and public holidays features to reflect the trends and patterns in carbon emissions.
- Interaction Features: Interaction features are generated by combining existing variables, such as energy consumption per unit of production, to represent emission intensity.

Outlier Detection and Removal

When the Z-score of the data falls over 3 or under -3 then it is treated as an outlier and dropped is represented in Eq. (4):

$$Z = \frac{X - \mu}{\sigma} \tag{4}$$

where Z represents the standardized score of the data point, X denotes the original value of the feature, μ indicates the mean of the dataset, and σ represents the standard deviation. Data points with Z-scores greater than 3 or less than -3 are considered outliers and removed.

Data Aggregation

The aggregation of data into suitable time intervals (e.g., daily, weekly) is to ensure consistency and time correspondence of time-series forecasting is represented in Eq. (5):

$$\text{Aggregate}(X) = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{5}$$

where Aggregate (X) represents the aggregated value of the data within a selected time interval, X_i denotes the individual data values within that interval, and n represents the total number of observations considered.

Encoding Categorical Variables

The encoding of categorical variables (e.g. industry type, energy source) can be done via OneHot Encoding or Label Encoding. One-Hot Encoding: This is used to encode a categorical variable in which there are N distinct categories:

Splitting Data

The processed data is further divided into the training and the test set, usually with $80-20$ or 70-30 partition into the model evaluation as well as validation.

Following the completion of these preprocessing procedures, the dataset becomes suitable for large-scale processing and predictive modelling. The integration of the prepared data within the proposed big data framework is described in the next subsection.

3.3. Real-Time Big Data Integration and Distributed Processing for Carbon Emission Prediction

The system integrates big data to monitor and predict carbon emissions by processing real-time data from smart meters, sensors, and external sources. Using distributed systems like Apache Hadoop, AWS S3, and Apache Kafka, it ensures scalable, fail-safe data handling.

Apache Spark enhances integration by processing large datasets in real-time and supporting machine learning models with in-memory computation. The data integration and processing process in Spark can be described in the following way formally is represented in Eq. (6):

$$\text{RDD}_{\text{processed}} = \text{filter}(\text{map}(\text{raw\_data})) \tag{6}$$

Where, raw data represents the original unprocessed dataset collected from different sources, map() applies transformations such as cleaning or formatting to each record, and filter() removes entries that do not meet quality requirements, such as missing or invalid values. Together, these operations generate $RDD_{processed}$, which is the cleaned and structured dataset ready for further analysis and model training.

The RDD is inputted into machine learning models like LSTM for time-series and XGBoost for feature refinement. Integrating real-time data helps update predictions, enhancing the precision of carbon emission forecasts and adapting to energy usage changes. The combination of big data and AI algorithms enables prediction to be dynamically adjusted in order to make the system accurate and responsive to the real-world changes in energy consumption and emissions data as shown in Fig. 2.

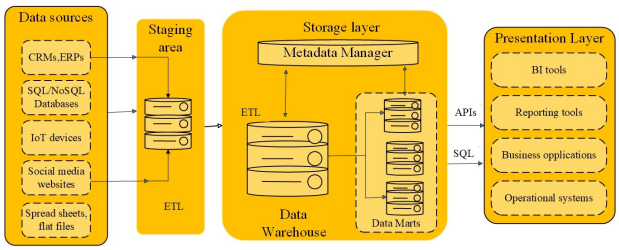

Fig. 2. Layered Big Data Architecture for Data Integration and Analytics.

This strata big data architecture integrates data from IoT devices, databases, enterprise systems, and flat files using ETL processes to a staging area. The storage layer includes a centralized data warehouse managed by a metadata manager and backed by a data mart for subject-specific analysis. BI tools access data via APIs and SQL.

3.4. Hybrid LSTM–XGBoost Framework for Real-Time Carbon Emission Prediction

The hybrid LSTM + XGBoost model is a combination of the LSTM networks and XGBoost to form a potent model of carbon emission prediction in real-time. The hybrid LSTM–XGBoost model integrates two distinct approaches to enhance carbon emission prediction. The LSTM network focuses on capturing temporal dependencies within the time-series data, allowing the model to identify and learn from long-term trends in energy consumption and emissions. Meanwhile, XGBoost contributes by addressing feature-based processing, optimizing the forecast through the inclusion of external variables such as industry type, energy mix, and economic indicators, while effectively modeling non-linear relationships. The combination of these two components enables the model to offer both temporal forecasting and feature-based optimization, resulting in a more robust and adaptable system for real-time carbon emission prediction. The use of LSTM is to model longitudinal correlations of the historical data, especially in time-series predictions of emissions on the basis of electricity consumption trends. The LSTM is especially useful in the process of capturing the long-term trends and seasonal trends because of its capacity to store information across a number of time steps. The model operates through the input of historical energy consumption data, which produces emission projections in the future. The LSTM equation of one-time step may be expressed as Eq. (7):

$$h_t = \sigma\left(W_h x_t + U_h h_{t-1} + b_h\right) \tag{7}$$

Where, x_t denotes the input vector at time step t, h_{t-1} is the previous hidden state, W_h and U_h are the weight matrices that transform the current input and past hidden state respectively, and b_h represents the bias term. The function $\sigma(\cdot)$ is an activation function-typically sigmoid or tanh-that introduces nonlinearity.

The LSTM network regulates information flow using three gates: the forget gate, input gate, and output gate. The forget gate controls the retention of previous cell state information, while the input gate determines the amount of new information introduced from the current input. The updated cell state integrates preserved historical information with candidate values to capture temporal dependencies. The output gate then regulates the portion of the cell state that contributes to the hidden state and prediction. The gating operations are computed using sigmoid and hyperbolic tangent activation functions. The LSTM produces forecasts using these temporal connections, which include the prediction of emissions, considering the previous history of consumption and seasonality.

After the LSTM model has generated an initial emission forecast, XGBoost is applied to enhance and tune these forecasts by adding more variables into the model e.g. industry type, energy mix and economic conditions. effective in nonlinear relationships and structured data learning. It takes the form of decision trees to model the interactions between features that are complex and can be stated is represented in Eq. (8):

$$f(x) = \sum_{k=1}^{K} \alpha_k h_k(x) \tag{8}$$

h_k represents the k-th decision tree in the boosted ensemble, a_k is the corresponding weight assigned to that tree, and K denotes the total number of trees used in the model. The function $f(x)$ is thus obtained by summing the weighted

outputs of all trees, which collectively refine the prediction through the gradient boosting process.

The gradient boosting framework, the prediction model is constructed as an additive ensemble of decision trees. The training process minimises an objective function consisting of a loss function and a regularisation term. The loss function measures the difference between actual and predicted values, while the regularisation component controls model complexity to prevent overfitting. The iterative optimisation, each tree reduces the residual errors of the previous model, thereby improving prediction accuracy. Combining LSTM and XGBoost leverages LSTM's temporal connections and XGBoost's ability to optimize forecasts with external factors, enhancing accuracy for real-time carbon emission projections and minimizing industries' carbon footprints.

The Fig. 3 demonstrates the working structure of the proposed hybrid LSTM -XGBoost model to achieve prediction of carbon emission. The inputs are historical +N)) by using timeseries patterns. Other external variables like energy trends, environmental conditions and industrial variables ($(E\{t+1\}, E\_\{t+2\}$, and so on)) are then added to the LSTM outputs. The XGBoost models further enhance these base predictions by training on nonlinear dependencies between the features to make the final emission predictions ((Y t + 1, Yt + 2, and so on Yt +N)) more accurate and stronger is show in Algorithm 1.

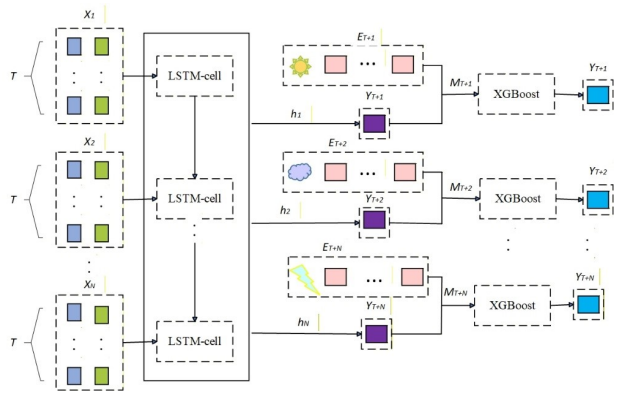

Fig. 3. Architecture of the Hybrid LSTM–XGBoost Model for Carbon Emission Prediction.

The proposed architecture and hybrid modelling approach establish the foundation for carbon emission prediction. The performance of the developed framework is evaluated through experimental analysis in the following section.

The configuration of the LSTM model is presented in Table 1. The model consists of two layers, each with 128 hidden units, enabling effective learning of temporal dependencies in the data. The tanh activation function is applied to hidden states, while sigmoid functions are used in the gating mechanisms to control information flow. A learning rate of 0.001 is employed with the Adam optimizer to ensure stable convergence during training. The model is trained using a batch size of 32 over 100 epochs, and a dropout rate of 0.2 is incorporated to reduce overfitting and enhance generalization performance.

Table 1. LSTM Model Hyperparameters.

Hyperparameter	Value
Number of LSTM Layers	2
Hidden Units per Layer	128
Activation Function	Tanh (hidden), Sigmoid (gates)
Learning Rate	0.001
Batch Size	32
Optimizer	Adam
Epochs	100
Dropout Rate	0.2

Table 2. XGBoost Model Hyperparameters.

Hyperparameter	Value
Max Depth	6
N_estimators	100
Learning Rate (η)	0.1
Lambda λ (L2 Regularization)	1.0
Gamma γ (Min Split Loss)	0.1
Subsample	0.8
Colsample_bytree	0.8

The configuration of the XGBoost model is presented in Table 2. The maximum depth of the trees is set to 6 to balance model complexity and generalization capability. A total of 100 estimators is used with a learning rate of 0.1 to iteratively improve prediction accuracy. Regularization is applied using $\lambda = 1.0$ to control model variance and $\gamma = 0.1$ to regulate tree splitting. Additionally, subsampling and column sampling rates are set to 0.8 to enhance robustness and prevent overfitting.

4. Result and discussion

Experimental evaluation is conducted to assess the performance of the proposed hybrid LSTMXGBoost framework using real-time industrial energy consumption data. The suggested carbon emission monitoring and prediction system integrates LSTM for time-series forecasting and XGBoost for improved predictions with supplementary attributes. Developed in Python with TensorFlow LSTM and XGBoost, it uses real-time data on electricity consumption and carbon emissions. The model, evaluated with RMSE, MAE, and R^2, is accurate, robust, and integrates big data

Algorithm 1. Hybrid LSTM–XGBoost

Start Step 1: Data Preprocessing
- **Input:** Historical energy consumption data and additional features
- **If** missing data exists in historical data:
  -Impute missing values (e.g., KNN)
- Normalize historical data
- Extract temporal features (day, month, seasonality)
- Combine historical data with additional features → combined data

Step 2: Train LSTM Model
- Initialize LSTM model → LSTM_model
- **While** epoch < max_epochs:
  - Train LSTM_ model on historical data
  - **If** epoch % validation interval == 0:
    - Evaluate LSTM model
    - **If** validation error < threshold → **Break**
  - epoch += 1

Step 3: Train XGBoost Model
- Initialize XGBoost model → XGBoost_model
- Combine LSTM_model predictions with additional features → XGBoost_input
- **Do While** epoch < max_epochs:
  - Train XGBoost_model on XGBoost_input
  - **If** epoch % validation interval == 0:
    - Evaluate XGBoost_model
    - **If** validation_error < threshold → **Break**
  - epoch += 1

Step 4: Prediction Phase
- **Input:** Real-time energy consumption data → real_time_data
- Predict future energy consumption → LSTM_predictions = LSTM_model. Predict(real_time_data)
- Combine LSTM_predictions with additional features → combined input
- Predict carbon emissions → predicted emissions = XGBoost_model. Predict (combined_input)

Output: Predicted carbon emissions → predicted emissions
Stop

technologies like Hadoop and Spark for real-time updates.

4.1. Dataset Evaluation

The experimental evaluation uses a dataset consisting of real-time electricity consumption records and corresponding carbon emission data collected from multiple industrial sectors. Additional contextual variables, including industry type, energy mix, economic indicators, and weather conditions, are incorporated to capture external factors influencing emission patterns. The dataset provides high temporal resolution with hourly and daily measurements, making it suitable for time-series forecasting tasks. Data preprocessing techniques such as normalization, missing value imputation, and feature engineering are applied to ensure data consistency and reliability during model training. The structured and high-resolution nature of the dataset enables effective training and evaluation of the hybrid LSTM-XGBoost framework for carbon emission prediction.

4.2. Performance Metrics

In order to test the performance of the proposed carbon emission monitoring and prediction system, some key performance metrics were applied. These indicators are used to determine the quality and strength of the hybrid LSTM + XGBoost model in forecasting carbon emissions using real-time electricity consumption data and other contributing variables.

Root Mean Squared Error: RMSE is essentially the square average value of the difference between forecasted and measured emissions, and the square being used to greater effect. The smaller the RMSE the higher the accuracy of prediction which is essential to the LSTM and the XGBoost models, as they require making predictions of the emissions by use of past and real time data is represented in Eq. (9):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y^i_{\text{predicted}} - y^i_{\text{actual}}\right)^2} \tag{9}$$

Where, y^i denotes the model's predicted value for the i-th sample, y^i_{actual} is the corresponding true value, and n represents the total number of samples. The RMSE metric computes the square root of the average squared error, providing a measure of overall prediction accuracy where lower values indicate better model performance.

Mean Absolute Error: calculates the mean of the differences between the actual and estimated results. The MAE unlike RMSE does not give more weight to larger discrepancies, which is the case of RMSE. Within the framework proposed, a better MAE means that the system is able to predict the carbon emissions correctly without being too sensitive to the outliers and thus is suitable in making real-time predictions is represented in Eq. (10):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y^i_{\text{predicted}} - y^i_{\text{actual}}\right| \tag{10}$$

$y^i_{\text{predicted}}$ represents the predicted value for the i-th observation, y^i_{actual} is the corresponding true value, and n denotes the total number of samples. The MAE calculates the average absolute difference between predictions and actual values, providing a straightforward measure of prediction accuracy where smaller values indicate better performance.

R-Squared: R^2 is a measure of the effectiveness of any prediction in containing the actual emissions with high value signifying excellent results. It is the percentage of the variation in the dependent variable (carbon emissions) which can be predicted due to the independent variables (electricity consumption, sector data). The greater R^2 means that the hybrid model (LSTM + XGBoost) is able to explain a considerable part of the dispersion of emissions data is represented in Eq. (11):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y^i_{\text{actual}} - y^i_{\text{predicted}}\right)^2}{\sum_{i=1}^{n}\left(y^i_{\text{actual}} - \bar{y}_{\text{actual}}\right)^2} \tag{11}$$

y^i_{actual} denotes the true value for the i-th sample, $y^i_{\text{predicted}}$ is the model's predicted value, and $\bar{y}_{\text{actual}}$ represents the mean of all actual values, while n is the total number of samples. This coefficient of determination R^2 measures how well the model explains the variance in the data, where values closer to 1 indicate stronger predictive performance.

Mean Squared Error: MSE is RMSE except that it does not take the square root. It focuses on larger errors by squaring of the differences between the predicted and the actual values. Within the suggested structure, MSE assists in assessing the general performance and identifying any severe error in prediction concerning emissions, especially in situations when the XGBoost model is used to boost predictions is represented in Eq. (12):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y^i_{\text{predicted}} - y^i_{\text{actual}}\right)^2 \tag{12}$$

$y^i_{\text{predicted}}$ denotes the predicted value for the i-th sample, y^i_{actual} is the corresponding true value, and nrepresents the total number of observations. The MSE measures the average squared difference between predicted and actual values, giving greater weight to larger errors and providing an important indicator of overall model accuracy.

The Fig. 4 Training vs Validation Metrics (MAE, MSE, RMSE) compares the performance of the model on both the training and validating datasets based on three important metrics which are MAE, Mean Squared error (MSE) and RMSE. The pink bars represent the training data set and green the validation data set. The model performs effectively in both, with a marginally larger MAE in training. Low MSE values confirm minor errors, while slightly lower RMSE in validation suggests predictive accuracy without overfitting, making the model reliable for carbon emissions prediction.

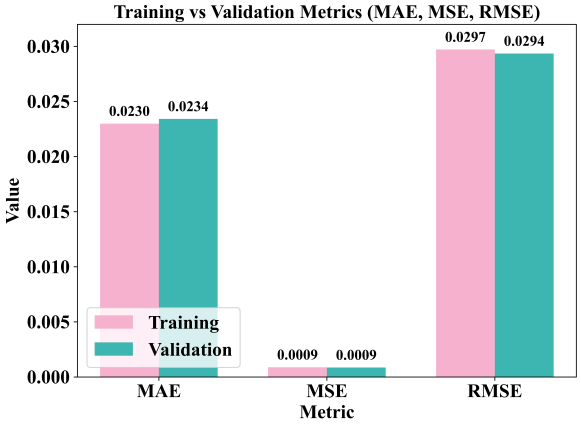

Fig. 4. Training vs Validation Error Metrics for LSTM + XGBoost Model.

4.3. Model evaluation

The proposed carbon emission monitoring system uses real-time data on energy consumption, carbon emissions, economic indicators, energy mix, and weather. Missing data is addressed through imputation methods like Autoencoders and KNN. LSTM models incorporate seasonal variations, enhanced by feature engineering. Apache Hadoop and Spark support real-time processing and updates.

This Fig. 5 demonstrates the Mean Squared Error (MSE) Loss in the course of training the LSTM + XGBoost hybrid model, in which it is possible to compare the training loss with the validation loss across different epochs. The training loss decreases quickly in early epochs, stabilizing after a few. Validation loss lags slightly, showing good generalization and no overfitting, ensuring solid performance on unseen data.

This Fig. 6 shows training and validation (R^2) scores of the LSTM + XGBoost model in consecutive epochs. The graph indicates that the value of the training and validation (R^2) is increasing steadily hence suggesting that the model is learning successfully and gaining its capacity to explain the variance in the target carbon emissions data. Closely matched training and validation curves indicate that the model is underfitted, which proves that the model can be applied to unseen information well as it approaches the high predictive accuracy.

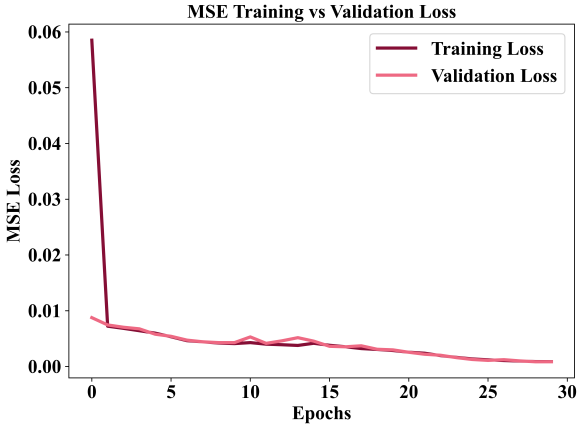

Fig. 5. MSE Training vs Validation Loss.

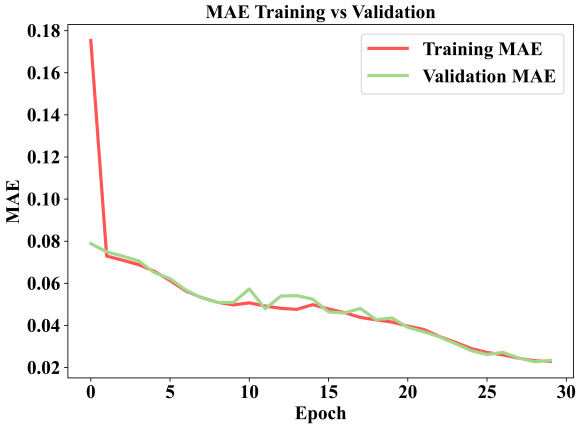

Fig. 7. MAE Training vs Validation.

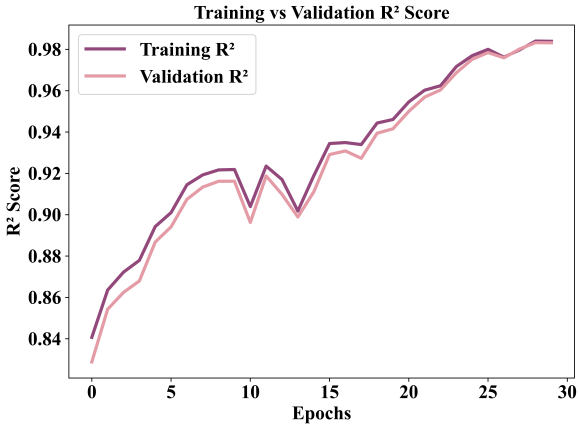

Fig. 6. Training vs Validation R^2 Score.

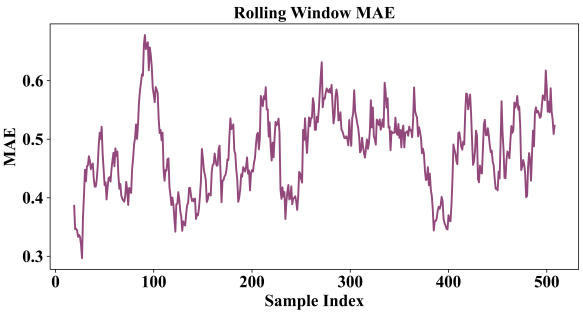

Fig. 8. Rolling Window MAE.

This Fig. 7 shows the comparison of the MAE of the Training and Validation datasets after 30 epochs. The red line represents Training MAE, and the green line represents Validation MAE. Both decrease and converge over time, indicating the model reduces errors and generalizes well without overfitting.

This Fig. 8 shows the Rolling Window MAE on a 500 data sample. The MAE is then plotted with the Sample Index and the error trend is observed as the model is predicted in a sliding window. The MAE variations reflect prediction accuracy changes due to data pattern shifts. The rolling window method averages errors to show consistent model performance and identify high prediction error periods.

4.3.1. Model Performance and Validation Results

The next section will contain a detailed analysis of the proposed carbon emission monitoring and prediction system. It will have several performance measurements and visualizations to evaluate the accuracy and reliability of the predictions made by the model. The findings indicate the usefulness of the hybrid LSTM + XGBoost model in projecting the carbon emissions, both in training and validation datasets.

This Fig. 9 represents the Cumulative Absolute error with time which is plotted against the Sample Index. Cumulative error shows the progressive increase in error as the sample index grows, tracking the model's forecast accuracy. It helps monitor errors over time, indicating areas for model improvements.

This Fig. 10 shows the QQ plot (Quantile-Quantile plot) will be a plot of the residuals of the model to determine whether they are normally distributed. The ranked residuals are plotted against the theoretical quantiles of a standard normal distribution. Points align with the red ideal line, indicating normality, validating model predictions and adjustments.

The Fig. 11 shown in this Residual Plot compares the observed and the expected value in the model. The plot shows residuals versus predicted values, with a red dashed

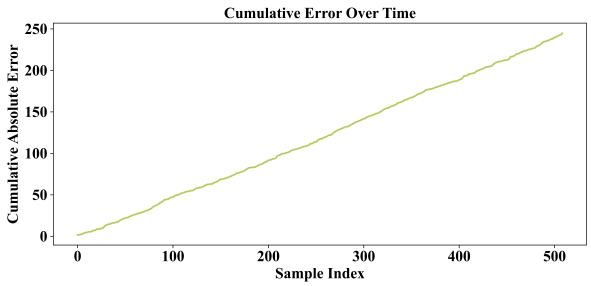

Fig. 9. Cumulative Error Over Time.

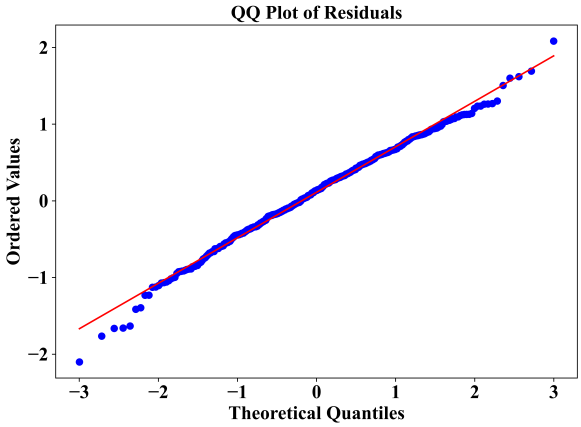

Fig. 10. QQ Plot of Residuals.

line at zero. Random dispersion of residuals indicates unbiased predictions and no systematic errors, confirming the model's suitability.

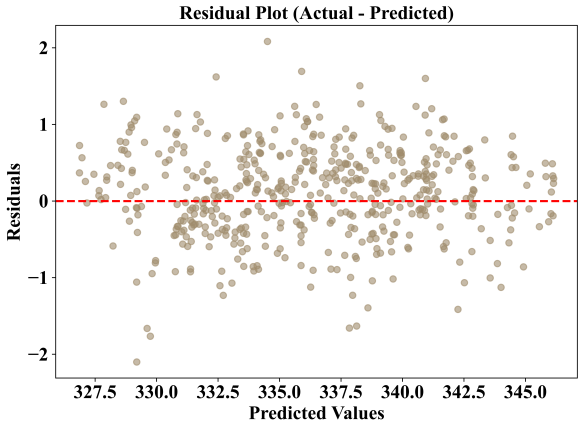

Fig. 11. Residual Plot (Actual - Predicted).

This Fig. 12 represented the comparison between the actual data (pink) and the expected data (teal) in the sample index. The graph shows that the expected and actual data points align closely, with smaller vertical lines indicating minimal differences, suggesting the model's accuracy and performance, despite some minor inaccuracies.

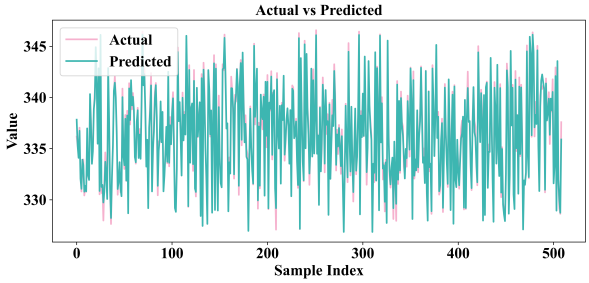

Fig. 12. Actual vs Predicted.

This Fig. 13 provides an analysis of how the actual values align with the predictive ones for validation data, where actual values are given on the x-axis, while the y-axis corresponds to the values that can be predicted. The red dashed line represents an ideal prediction scenario, while the green dots show the close alignment of actual and predicted data, indicating minimal variation in predicting accuracy.

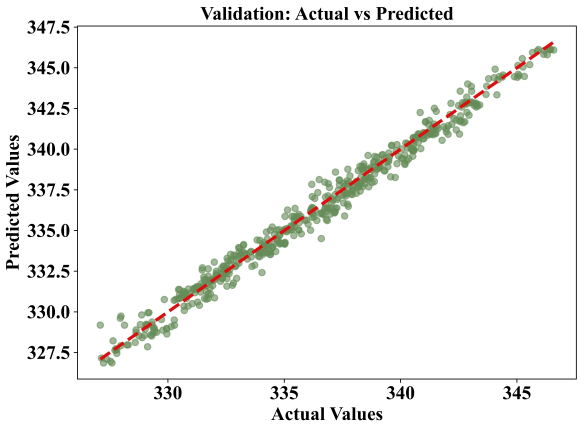

Fig. 13. Validation. Actual vs Predicted.

This graph enables us to compare the actual values with the predicted values for the training dataset. The actual values are plotted on the x-coordinate, and the predicted values are plotted on the y-coordinate. The red dashed line in this graph shows the points where the predicted values would be equal to the actual values. The data points in this Fig. 14 shown using green colors. These points being close to the red dashed line indicate that the predictions made by the model are quite accurate. The data points in the graph show a strong positive linear correlation between actual

and predicted values, indicating the model effectively understands patterns with minimal deviation from the ideal line.

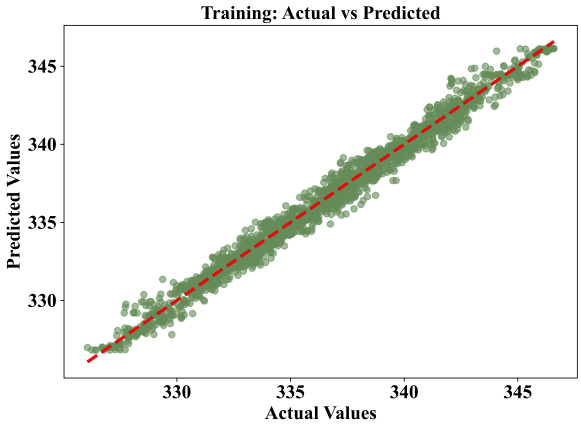

Fig. 14. Training. Actual vs Predicted.

4.4. HDFS Directory View of Model Evaluation Artifacts

This Fig. 15 shows a Hadoop File System (HDFS) directory view, displaying a list of files stored under the /project results/results/Architectural_Artifacts directory.

The files comprise several metrics for model performances like accuracy, confusion matrix metrics, precision metrics, and ROC, which are denoted by the file names class1 performance.png, class1 confusion metrics.png, and class1 roc.png respectively. The files are listed along with their size, permissions, and other characteristics like ownership. The image signifies the organization of the model's performance analysis in Hadoop for easier access of evaluations.

4.5. Ablation table

This Table 3 offers a comparative study of different model designs on the basis of error measures (RMSE, MAE), goodness of fit (R^2), and runtime. Although the LSTM-only model is able to identify the time series dependencies quite effectively, it does not include any realtime refinements on the feature set, while on the other hand, the purely XGBoost model is able to address non-linear relations quite effectively but is not designed to address time series.

Nevertheless, it is evident that a model designed using both LSTM and XGBoost techniques outperforms all other options on the basis of low error values and high goodness of fit. Even those with a reduced set of features, without any time series, and real-time information also exhibit a degraded performance.

4.6. Performance Comparison

As the following Table 4 and Fig. 16 show, there are comparisons made through four metrics: MAE, MSE, RMSE, and R^2, between the predictive capacity of three carbon emission forecasting approaches. The error values and R^2 metrics of the CNN-LSTM-BERT model are the largest, meaning its forecasting capacity is relatively poor.

The AdaBoost helps to decrease the error terms and enhance the goodness of fit. The proposed model of LSTM + Boost has the least MAE, MSE, and RMSE along with the highest R^2 value to present better accuracy and robustness. The graphical representation plots the efficacy of the proposed hybrid model to depict a substantial minimization of prediction error and a maximization of explanatory power.

The cross-validation performance results are presented in Table 5. The 5-fold cross-validation demonstrates consistent model performance across all folds, with minimal variation in RMSE, MAE, MSE, and R^2 values. The low standard deviation indicates stability and robustness of the proposed model across different data splits. These results confirm the reliability and generalization capability of the model.

The confidence interval results for model performance metrics are presented in Table 6. The 95% confidence intervals for RMSE, MAE, MSE, and R^2 are relatively narrow, indicating high precision and stability of the model predictions. This demonstrates that the reported performance values are reliable and not significantly affected by data variability.

The statistical significance test results are presented in Table 7. A paired t-test is conducted to compare the proposed LSTM + XGBoost model with CNN-LSTM-BERT and AdaBoost models. The obtained p-values (<0.0001) indicate that the performance differences are statistically significant.

4.7. Discussion

The proposed framework for the prediction and monitoring of carbon emission shows high effectiveness in terms of incorporating big data technologies with a hybrid LSTM + XGBoost approach. The results obtained show that the prediction process with temporal information incorporated with feature learning has a highly positive effect on improving the accuracy of predictions. The LSTM technique has effectively identified the temporal patterns and seasonal variations in the consumption of electricity, which is a highly essential process for the prediction of real-world carbon emission patterns, as indicated by low values for errors, with MAE = 0.0234, MSE = 0.00086, and RMSE = 0.0294, and a high R-squared value of 0.963 , showing that a large amount of variance in the carbon emission is

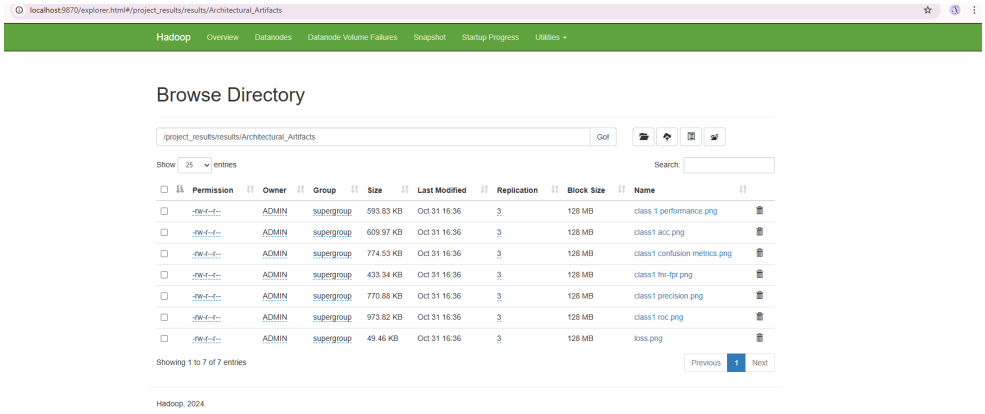

Fig. 15. Hadoop File System Browser Directory View.

Table 3. Comparative Performance Analysis of LSTM, XGBoost, and Hybrid Forecasting Models.

Model Configuration	RMSE	MAE	R^2	Execution Time	Key Observations
1. LSTM Only (Baseline)	0.0297	0.0230	0.960	120s	Captures temporal dependencies, but lacks refinement with external factors.
2. XGBoost Only	0.0453	0.0314	0.930	150s	Works well for non-linear relationships, but does not capture temporal trends.
3. LSTM + XGBoost (Proposed Framework)	0.0294	0.0234	0.963	180s	Best performance with accurate temporal and feature-based adjustments.
4. LSTM + XGBoost with Limited Features	0.0301	0.0242	0.957	175s	Missing key external factors, resulting in slightly reduced accuracy.
5. LSTM + XGBoost with Missing Temporal Features	0.0325	0.0261	0.950	160s	Temporal features significantly improve forecasting, making them crucial.
6. LSTM + XGBoost with No Real-Time Updates	0.0338	0.0270	0.945	190s	Real-time updates provide more accurate and up-todate predictions.

Table 4. Performance Comparison of Carbon Emission Prediction Systems.

Method	MAE	MSE	RMSE	R^2
CNN, BERT [29]	0.0186	0.00062	0.0212	0.910
AdaBoost [30]	0.0251	0.00008	0.0228	0.952
LSTM + Boost	0.0234	0.00086	0.0294	0.963

Table 5. 5-Fold Cross-Validation Results.

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean ± Std
RMSE	0.0301	0.0289	0.0294	0.0298	0.0288	0.0294 ± 0.0005
MAE	0.0241	0.0229	0.0234	0.0238	0.0228	0.0234 ± 0.0005
MSE	0.00091	0.00083	0.00086	0.00089	0.00083	0.00086 ± 0.000030
R^2	0.961	0.965	0.963	0.962	0.964	0.963 ± 0.0014

being explained by the technique. The results obtained are highly efficient in terms of demonstrating high values for the goodness of fit for the proposed technique in comparison to others, including the CNN-LSTM-BERT and AdaBoost approaches, with low values for errors in prediction.

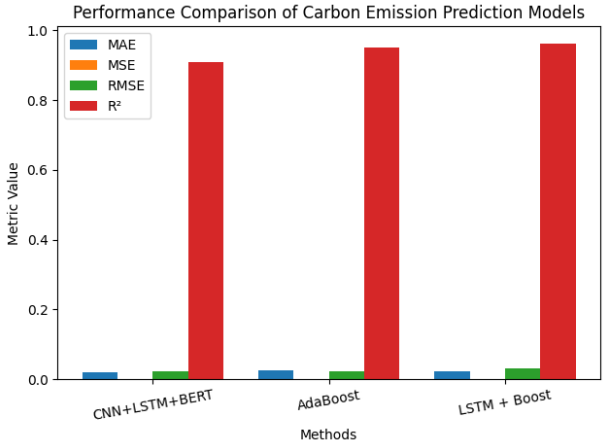

Fig. 16. Performance Comparison of Carbon Emission Prediction Models.

Table 6. 95% Confidence Intervals for Performance Metrics.

Metric	Reported Value	95% CI Lower	95% CI Upper
RMSE	0.0294	0.0267	0.0321
MAE	0.0234	0.0211	0.0258
MSE	0.00086	0.00070	0.00102
R^2	0.963	0.955	0.971

Table 7. Statistical Significance Test.

Comparison	Test Used	t-statistic	p-value	Significant?
LSTM+XGBoost vs CNN-LSTM-BERT	Paired t-test	25.81	< 0.0001	Yes
LSTM+XGBoost vs AdaBoost	Paired t-test	-8.69	< 0.0001	Yes

Besides, big data platforms such as Apache Kafka and Apache Spark enable real-time data ingestion, processing, and prediction. This is a fundamental improvement compared to traditional static approaches, which do not benefit from these features. Because the framework continuously updates the forecast from the real-time electricity consumption and other conditions, it promises adaptability to dynamic industrial conditions. Different variants of the model with a lack of temporal features or real-time updates showed a noticeable degradation in performance, indicating that temporal learning and live data streams play an important role. On the whole, experimental results and comparative analysis confirm the proposed framework as robust, scalable, and well suited for an enterprise-level carbon-emission monitoring task that can support timely decision-making on emission reduction and sustainability planning.

5. Conclusion and future works

Carbon emission monitoring and prediction are critical for supporting sustainable industrial development. A big data-driven framework combining Long Short-Term Memory (LSTM) networks and XGBoost is used to model temporal energy consumption patterns and nonlinear relationships among industrial variables. The integration of real-time electricity consumption data with contextual factors such as industry type, energy mix, and economic indicators enables more accurate modelling of carbon emission behaviour. The hybrid LSTM-XGBoost approach achieves strong predictive performance with MAE $= 0.0234$, MSE $= 0.00086$, RMSE $= 0.0294$, and $R^2 = 0.963$. The results highlight the effectiveness of combining temporal learning with feature-based optimisation for carbon emission forecasting. The use of distributed big data platforms including Apache Kafka and Apache Spark enables continuous data ingestion and scalable processing, supporting real-time monitoring and prediction in industrial environments.

Future Works:

- Add new components by utilising other data sources such as satellite data for emissions and IoT sensors for better resolution in space and time.

- Investigate more complex deep learning architectures, for instance, the Transformer or attention mechanisms, to further improve the modelling of long-term dependencies and prediction capability.
- Develop an interactive decision support tool incorporating results of predictions, optimisation algorithms, and recommendations related to real-time carbon management and carbon trading.

Declaration

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflict of Interest

The author declares no conflict of interest.

Funding Statement

This study is funded by the National Philosophy and Social Science Foundation of China (Grant No. 22CTJ010) and China Postdoctoral Science Foundation (Grant No. 2024M750205).

Author Contribution

The author contributed to the conception, design, analysis, and writing of the manuscript.

Ethical Approval

Ethical approval was not required for this study.

References

[1] G. Demirdöğen, Z. Işık, and Y. Arayici, (2020) *"Lean Management Framework for Healthcare Facilities Integrating BIM, BEPS and Big Data Analytics"* **Sustainability *12***(17): 7061. DOI: 10.3390/su12177061.

[2] A. H. A. AL-Jumaili, Y. I. A. Mashhadany, R. Sulaiman, and Z. A. A. Alyasseri, (2021) *"A Conceptual and Systematics for Intelligent Power Management System-Based Cloud Computing: Prospects, and Challenges"* **Applied Sciences *11***(21): 9820. DOI: 10.3390/app11219820.

[3] S. P. R. Asaithambi, R. Venkatraman, and S. Venkatraman, (2020) *"MOBDA: Microservice-Oriented Big Data Architecture for Smart City Transport Systems"* **Big Data and Cognitive Computing *4***(3): 17. DOI: 10.3390/bdcc4030017.

[4] P. Fraga-Lamas, S. I. Lopes, and T. M. Fernández-Caramés, (2021) *"Green IoT and Edge AI as Key Technological Enablers for a Sustainable Digital Transition towards a Smart Circular Economy: An Industry 5.0 Use Case"* **Sensors *21***(17): 5745. DOI: 10.3390/s21175745.

[5] X. Tao, L. Cheng, R. Zhang, W. K. Chan, H. Chao, and J. Qin, (2024) *"Towards Green Innovation in Smart Cities: Leveraging Traffic Flow Prediction with Machine Learning Algorithms for Sustainable Transportation Systems"* **Sustainability *16***(1): 251. DOI: 10.3390/su16010251.

[6] R. Singh, S. V. Akram, A. Gehlot, D. Buddhi, N. Priyadarshi, and B. Twala, (2022) *"Energy System 4.0: Digitalization of the Energy Sector with Inclination towards Sustainability"* **Sensors *22***(17): 6619. DOI: 10.3390/s22176619.

[7] X. Dai, Y. Chen, C. Zhang, Y. He, and J. Li, (2023) *"Technological Revolution in the Field: Green Development of Chinese Agriculture Driven by Digital Information Technology (DIT)"* **Agriculture *13***(1): 199. DOI: 10.3390/agriculture13010199.

[8] E. Badidi, Z. Mahrez, and E. Sabir, (2020) *"Fog Computing for Smart Cities' Big Data Management and Analytics: A Review"* **Future Internet *12***(11): 190. DOI: 10.3390/fi12110190.

[9] N. L. H. Hien and A.-L. Kor, (2022) *"Analysis and Prediction Model of Fuel Consumption and Carbon Dioxide Emissions of Light-Duty Vehicles"* **Applied Sciences *12***(2): 803. DOI: 10.3390/app12020803.

[10] J. Gusc, P. Bosma, S. Jarka, and A. Biernat-Jarka, (2022) *"The Big Data, Artificial Intelligence, and Blockchain in True Cost Accounting for Energy Transition in Europe"* **Energies *15***(3): 1089. DOI: 10.3390/en15031089.

[11] H. Huang, X. Wu, and X. Cheng, (2021) *"The Prediction of Carbon Emission Information in Yangtze River Economic Zone by Deep Learning"* **Land *10***(12): 1380. DOI: 10.3390/land10121380.

[12] P. V. Thayyib and et al., (2023) *"State-of-the-Art of Artificial Intelligence and Big Data Analytics Reviews in Five Different Domains: A Bibliometric Summary"* **Sustainability *15***(5): 4026. DOI: 10.3390/su15054026.

[13] H. Zheng, T. Zhang, C. Fang, J. Zeng, and X. Yang, (2021) *"Design and Implementation of Poultry Farming Information Management System Based on Cloud Database"* **Animals *11***(3): 900. DOI: 10.3390/ani11030900.

[14] K. N. Shivaprakash and et al., (2022) *"Potential for Artificial Intelligence (AI) and Machine Learning (ML) Applications in Biodiversity Conservation, Managing Forests, and Related Services in India"* **Sustainability *14***(12): 7154. DOI: 10.3390/su14127154.

[15] D. Zhou and et al., (2022) *"Intelligent Manufacturing Technology in the Steel Industry of China: A Review"* **Sensors *22***(21): 8194. DOI: 10.3390/s22218194.

[16] C. Zhou, X. Lin, R. Wang, and B. Song, (2023) *"Real-Time Carbon Emissions Monitoring of High-Energy-Consumption Enterprises in Guangxi Based on Electricity Big Data"* **Energies *16***(13): 5124. DOI: 10.3390/en16135124.

[17] S. Almanasra, (2024) *"Applications of integrating artificial intelligence and big data: A comprehensive analysis"* **Journal of Intelligent Systems *33***(1): DOI: 10.1515/jisys-2024-0237.

[18] T. Peng, X. Yang, Z. Xu, and Y. Liang, (2020) *"Constructing an Environmental Friendly Low-Carbon-Emission Intelligent Transportation System Based on Big Data and Machine Learning Methods"* **Sustainability *12***(19): 8118. DOI: 10.3390/su12198118.

[19] F. Xu and et al., (2025) *"Geospatial Big Data-Driven Fine-Scale Carbon Emission Modeling"* **Remote Sensing *17***(18): 3185. DOI: 10.3390/rs17183185.

[20] J. Rubio-Loyola and W. R. S. Paul-Fils, (2022) *"Applied Machine Learning in Industry 4.0: Case-Study Research in Predictive Models for Black Carbon Emissions"* **Sensors 22**(10): 3947. DOI: 10.3390/s22103947.

[21] J. Udoh, J. Lu, and Q. Xu, (2024) *"Application of Machine Learning to Predict CO2 Emissions in Light-Duty Vehicles"* **Sensors *24***(24): 8219. DOI: 10.3390/s24248219.

[22] C. Zhou, Y. Tang, D. Zhu, and Z. Cui, (2024) *"Tracking the Carbon Emissions Using Electricity Big Data: A Case Study of the Metal Smelting Industry"* **Energies *17***(3): 652. DOI: 10.3390/en17030652.

[23] C. D. Fay, B. Corcoran, and D. Diamond, (2024) *"Green IoT Event Detection for Carbon-Emission Monitoring in Sensor Networks"* **Sensors *24***(1): 162. DOI: 10.3390/s24010162.

[24] K. U. Khan, G. Ali, N. Murtaza, Y. Pan, and V. Kysucky, (2025) *"Toward Net-Zero Emissions: The Role of Smart City Technologies in Reducing Carbon Emissions in China"* **Urban Science *9***(9): 374. DOI: 10.3390/urbansci9090374.

[25] L. Wang and et al., (2024) *"The Design and Implementation of an Intelligent Carbon Data Management Platform for Digital Twin Industrial Parks"* **Energies *17***(23): 5972. DOI: 10.3390/en17235972.

[26] E. Bicamumakuba, M. N. Reza, H. Jin, Samsuzzaman, K.-H. Lee, and S.-O. Chung, (2025) *"Multi-Sensor Monitoring, Intelligent Control, and Data Processing for Smart Greenhouse Environment Management"* **Sensors *25***(19): 6134. DOI: 10.3390/s25196134.

[27] A. A. Alhussan and M. Metwally, (2025) *"Enhanced CO2 Emissions Prediction Using Temporal Fusion Transformer Optimized by Football Optimization Algorithm"* **Mathematics *13***(10): 1627. DOI: 10.3390/math13101627.

[28] NOAA Global Monitoring Laboratory. *Data Finder - NOAA Global Monitoring Laboratory*.

[29] K. Liu, H. Ren, S. Lu, X. Shang, Z. Liu, and B. Yu, (2026) *"Analysis and Prediction Evaluation of Provincial Carbon Emissions Under Multi-Model Fusion"* **Sustainability *18***(5): 2545. DOI: 10.3390/su18052545.

[30] R. Ji, (2024) *"Research on Factors Influencing Global Carbon Emissions and Forecasting Models"* **Sustainability *16***(23): 10782. DOI: 10.3390/su162310782.