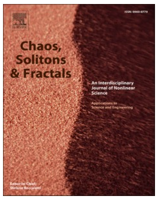
Chaos, Solitons and Fractals 188 (2024) 115533

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos



An optimized and interpretable carbon price prediction: Explainable deep learning model

Gehad Ismail Sayed [a,1], Eman I. Abd El-Latif [b], Ashraf Darwish [c,1], Vaclav Snasel [d], Aboul Ella Hassanien [e,f,*,1]

[a] *School of Computer Science, Canadian International College (CIC), Cairo, Egypt*
[b] *Faculty of Science, Benha University, Benha, Egypt*
[c] *Faculty of Science, Helwan University, Cairo, Egypt*
[d] *Faculty of Electrical Engineering and Computer Science VSB–Technical University of Ostrava, Ostrava, Czech Republic*
[e] *College of Business Administration, Kuwait University, Shidadiya Office B2-104, P.O. Box 5486, Safat, Kuwait 13055, Kuwait*
[f] *Faculty of Computers and Artificial Intelligence, Cairo University, Cairo, Egypt*

ARTICLE INFO

Keywords:
Carbon price
Carbon emission
Light spectrum optimizer
Explainable deep-learning
Long short term memory
Hyper-parameters optimization

ABSTRACT

Accurate prediction of carbon prices is crucial for a stable market, enabling informed decision-making and strategic planning. Over the years, several models for predicting carbon prices have been proposed to enhance accuracy. However, previous research has primarily focused on improving accuracy, often neglecting the importance of making the findings understandable and meaningful. This paper aims to bridge that gap by not only improving prediction accuracy but also ensuring that the results are transparent and comprehensible, thus contributing to more effective and informed decision-making in the carbon market. An optimized Long Short-Term Memory (LSTM) network enhanced with the modified light spectrum optimizer (MLSO) is proposed to improve carbon price prediction accuracy. Additionally, the paper incorporates Explainable AI (XAI) techniques to interpret the results, bridging the gap between accuracy and interpretability. The proposed model is evaluated on carbon price historical transaction data acquired from Investing.com and tested on eight other benchmark datasets with different characteristics. The proposed model achieved 0.66 root mean square error (RMSE), 0.99 R^2, 0.37 mean absolute error (MAE), 0.15 mean absolute percentage error (MAPE), and 0.44 mean square error (MSE). The results showed that low price, high price, and open price features have the highest significance in driving the model's predictions in comparison to other features like date, volume, and price change features. Additionally, the results indicate that the year, day, and month do not significantly influence the carbon price. The proposed model outperforms state-of-the-art models and other well-known machine learning algorithms according to the experimental results. Moreover, the results indicate that the predictive capability of the proposed model serves as a valuable tool for investors and carbon traders to understand the factors influencing price changes, optimize their strategies, and minimize risk.

1. Introduction

Carbon pricing is a market-based strategy targeted at reducing greenhouse gas (GHG) emissions by placing a price on carbon emissions. It involves making emitters pay for the greenhouse gases released into the environment, such as carbon dioxide (CO2) [1,2]. A carbon tax is a charge directly related to the amount of carbon in fossil fuels. Making it more expensive to release carbon provides a financial incentive for businesses and individuals to limit their carbon emissions. The cost of generating carbon increases, promoting the adoption of greener practices and technology [3,4]. The annual State and Trends of Carbon Pricing report, which the World Bank has been producing for 10 years, has been following carbon markets for around 20 years. Only 7 % of worldwide emissions were subject to a carbon price when the first study was released ten years ago [5]. Governments, businesses, and long-term investors all face serious consequences as a result of rising carbon costs.

* Corresponding author at: College of Business Adminstration (CBA) Kuwait University, Kuwait.
E-mail addresses: gehad_sayed@cic-cairo.com (G.I. Sayed), eman.mohamed@fsc.bu.edu.eg (E.I. Abd El-Latif), ashraf.darwish.eg@ieee.org (A. Darwish), vaclav.snasel@vsb.cz (V. Snasel), abo.aly@ku.edu.kw, aboitcairo@cu.edu.eg (A.E. Hassanien).

[1] Scientific Research School of Egypt (SRSEG), https://www.egyptscience-srge.com/.

https://doi.org/10.1016/j.chaos.2024.115533
Received 25 April 2024; Received in revised form 10 September 2024; Accepted 11 September 2024
Available online 20 September 2024

One tool governments employ to reduce carbon emissions is carbon pricing, which can also generate revenue [6,7]. Companies can use carbon pricing to study how required carbon pricing affects their operations and to find opportunities for revenue generation and potential climate hazards. Carbon pricing is a tool that long-term investors use to analyze their investment strategies. Therefore, independent of viewpoint, a reliable and consistent carbon price forecasting system must be established.

Variations in the price of carbon greatly influence how agriculture, energy, industries, and stock investments grow. Accurate prediction of the regional carbon price can aid in reducing carbon dioxide emissions and serve as a crucial foundation for modifying carbon pricing policies, which will regulate the carbon trading markets and help investors minimize investment risk [8–10]. It is expected that carbon prices will rise globally between 2026 and 2030 across all emissions trading schemes, compared to 2022 to 2026. According to a study of members of the International Emissions Trading Association, the average The European Union Emissions Trading Scheme (EUETS) carbon price is anticipated to be 84.4 euros per ton of CO2 during the years 2022 to 2025, but it is anticipated to increase to approximately 100 euros per ton of CO2 during the years 2026 to 2030 [11]. The majority of research techniques for carbon price prediction focus on creating models using data from previous studies. The high complexity, non-stationarity, nonlinearity, and of carbon pricing make it challenging to make precise forecasts. Deep learning has recently been applied in many papers because of its superior prediction accuracy and ability to overcome the drawbacks of traditional algorithms. Numerous forecasting domains, including wind power forecasting [13,14] and power load forecasting [15]; have effectively used the deep learning approach. Additionally, deep learning's long short-term memory networks (LSTM) are especially efficient at fitting time-series data. As a result, using LSTM to estimate carbon prices can improve the results of research.

In this paper, a model for accurate carbon price prediction was proposed. The proposed model mainly depends on LSTM, where the LSTM layer is adopted and a modified version of light spectrum optimizer, shortly MLSO. The proposed custom LSTM architecture utilizes the proposed MLSO algorithm to find the optimal hyper-parameters to enhance the capacity to learn data sequence characteristics. Three primary phases make up the proposed model: data preprocessing phase, hyper-parameters optimization using MLSO phase, and evaluation and result interpretation phase. Missing values are addressed and data normalization is applied to the original historical transaction data during the data preprocessing phase. Next, train, valid, and test subsets are constructed from the processed data. The subsequent step, which involves optimizing hyperparameters using MLSO, is fed by the processed train and valid subset data. The proposed MLSO is employed in this phase to find the optimal hyper-parameter values of LSTM. These hyper-parameters can significantly affect the performance of a regression model; thus, they can significantly affect the results. Then, the optimal hyper-parameters selected from the previous phase are used to train the proposed custom LSTM architecture. Finally, explainable artificial intelligence (XAI) and a number of measurements are used to assess the effectiveness of the proposed model throughout the evaluation and results interpretation phase. As far as we know, no existing model incorporates the interpretation technique described in this paper with hyper-parameter optimization of LSTM. The following is a list of the paper's main contributions:

1. A new model combining the optimized LSTM network with MLSO is proposed for precise carbon price forecasting, leveraging the strengths of both techniques for superior performance.
2. The proposed MLSO is employed to determine the optimal batch size, learning rate, and solver optimization algorithm for the LSTM, enhancing its accuracy and efficiency.
3. The Explainable Artificial Intelligence (XAI) technique is utilized to improve the transparency and interpretability of the model's predictions, ensuring that stakeholders can understand and act on the results.
4. The proposed model is evaluated on nine diverse benchmark datasets, including one from Investing.com, six from Kaggle, and two recent datasets (BeijingETS and HubeiETS), demonstrating its robustness, and effectiveness across multiple scenarios, ensuring its applicability in various contexts.
5. A comparative analysis of different swarm-based hyper-parameters optimization algorithms for LSTM is performed, highlighting the superiority and advantages of the proposed MLSO algorithm over other algorithms.

The remainder of the paper is structured as follows: The existing methods of predicting carbon prices are reviewed in Section 2. The long short-term memory AND light spectrum optimizer are briefly addressed in Section 3. Section 4 provides a thorough description and analysis of the adopted dataset. The proposed model for carbon price prediction is presented in details in Section 5. In Section 6, the outcomes and experimental findings are discussed. In Section 7, the main findings and the future directions is presented.

2. Literature review

Researchers have conducted several studies on carbon price prediction throughout the years and have put forward numerous predicted techniques. These techniques may generally be divided into two groups: econometric and machine learning methods. Econometric methods combine sales data with the knowledge that affects carbon prices. Then, develop a mathematical formula to forecast carbon price. Econometric techniques, which include autoregressive integrated moving average (ARIMA) [17], and generalized autoregressive conditional heteroskedasticity (GARCH) [16] are the linear forecasting tools that are often used in carbon prediction and deliver acceptable performance. Bangzhu et al. [18] used ARIMA with least squares support vector machine (LSSVM) to predict carbon prices. The best LSSVM parameters have been identified using particle swarm optimization (PSO). Ajay et al. [19] used the ARIMA-GARCH model to predict the carbon's volatility. However, in the face of volatile carbon series, forecast accuracy is impossible to achieve due to a lack of capacity to manage nonlinearity and nonstationary.

On the other hand, machine learning algorithms are employed to forecast the price of carbon to overcome the drawbacks of econometric models. In [20], a binary-class prediction issue is first created from the carbon price prediction problem. Then, the SVM parameters are optimized using the genetic algorithm (GA). Fast multi-output relevance vector regression (FMRVR) and variational mode decomposition (VMD) are used by Xiong et al. [21]. VMD is applied to extract the features from the input data and for forecasting module, FMRVR, is then constructed using the features. Katarzyna et al. [22] merged the Principal Component Analysis (PCA) and different classifiers such as decision trees, and random forests for predicting carbon prices. In [23], first, three traditional indicators are used to determine the chaotic features of the carbon price series. A multi-layer perceptron neural network prediction model has been constructed to identify the carbon price's significant nonlinearity. The model validation is demonstrated using the K-fold cross-validation approach. A total of four metrics are utilized to determine the MLP model's performance. ZHU et al. introduced an algorithm for estimating the price of carbon [24]. The selection of input-layer units is made simple by using the group method of data handling (GMDH). Then, particle swarm optimization (PSO) is applied to determine the ideal parameters. Finally, the forecast of the testing samples' carbon price is made using the trained least squares-SVM. Authors in [25] predict the carbon trading price using the suggested hybrid model of SSA + LSTM based on the carbon trading price. After that, the suggested algorithm is compared with the classical econometrics methods. In terms of carbon forecasts, machine learning techniques are producing encouraging results. However, the implementation of machine learning techniques is more challenging and requires complex parameter settings, which can quickly lead to overfitting and poor convergence. Hybrid approaches are suggested as a method to overcome the limits by utilizing various features of several models. For instance, Zhu et al. [26] improved accuracy greater than SVM individually by combining empirical mode decomposition (EMD) with SVM to forecast carbon data. Later, EMD was added to GARCH, improving the model's accuracy [27]. In [28], the authors suggested a hybrid forecasting algorithm for predicting carbon prices. First, the best input variables for modeling are chosen using an extremely randomized tree (ET). The screened input variables are then decomposed into sub-modes using multivariate variational mode decomposition (MVMD). Then, long short-term memory (LSTM) is applied to each sub-mode to extract features. The ensemble of all sub-mode forecasts reconstructs the final forecast. In [29], the authors proposed a hybrid approach based on utilizing the Adaptive Neuro-Fuzzy Inference System (ANFIS) and the Electric Fish Optimization (EFO). In this paper, a modified version of EFO is proposed to find the optimal hyper-parameters settings for ANFIS. Although the suggested model performed well in predicting carbon prices over five datasets, the authors mainly concentrated on improving accuracy without explaining the results. C. Yang et al. [30] utilized XGBoost, stochastic gradient descent, linear regression, support vector regression, random forest, and among other machine learning algorithms to predict the carbon price. To examine and evaluate the factors impacting carbon pricing, they also utilized the SHAPley Additive exPlanations (SHAP) method, taking into account the COVID-19 vaccination program's influence. According to the results, daily vaccination rates are a better indicator of carbon pricing than market volatility. However, the authors neglected to take into consideration the machine learning algorithms' hyper-parameters optimization, which could have had a big influence on the results. In [31], the lower upper bound estimation model (LUBE) was developed as a method for generating asymmetric prediction intervals without making assumptions. The authors utilized causal inference in feature selection to enhance generalization and developed multi-objective evolutionary algorithms to attain more effective outcomes. The results showed the advantages of using ensemble methods in conjunction with LUBE. The authors of this paper did not discuss the results' interpretation, which could provide useful information on how the model's predictions are formed and what factors influence them. Table 1 compares the econometric, machine learning, and hybrid techniques for carbon price prediction.

Previous studies on the prediction of carbon prices have mostly concentrated on improving forecasting model accuracy. Interpretability and model hyper-parameters optimization are two key aspects that are often ignored in previous studies, although these efforts have improved predictive performance. Many existing models lack interpretability, making it difficult for stakeholders such as policymakers and investors to clearly understand the underlying causes driving predictions. Furthermore, hyper-parameters optimization of an algorithm has generally been based on manual tuning or standard optimization approaches, which it is time-consuming and may not produce the best effective configurations across different datasets. This paper fills these research gaps by combining Explainable AI (XAI) techniques with an optimized LSTM model, where MLSO is employed to fine-tune the hyper-parameters of LSTM. Through the use of XAI, stakeholders can obtain insights into factors impacting carbon prices and be certain that the predictions produced by the model are both accurate and understandable. The proposed model's detailed description will be covered next.

3. Material and methods

This section covers the original light spectrum optimizer algorithm's (LSO) primary source of inspiration as well as its mathematical model. It also offers a brief overview of long short-term memory (LSTM) architecture.

3.1. Light spectrum optimizer (LSO)

Rainbow spectrum rays are the result of colorful light being scattered. The LSO algorithm is motivated by this behavior [32–35]. LSO is based on the following presumptions in particular:

- The ranges of light ray scattering are between 40° and 42°, and the refractive index of light ray scattering varies between $t^{violet}= 1.344$ and $t^{red}= 1.331$.
- A potential solution is shown by each colorful beam.
- Refraction and reflection (internal and external) are determined randomly.
- Compared to the fitness of the best so far solution, the fitness value of the present solution determines the first and second scattering phases of a colorful rainbow curve. Imagine that they have similar fitness values. In that case, the algorithm will use the first scattering phase to search surrounding regions because the current solution can be so close to the near-optimal one. If not, the second step will keep the proposed model from stopping in the best-so-far solution's areas because those areas might contain local minima.

The following describes the mathematical explanations of the LSO algorithm through three stages called initialization stage, the rainbow

Table 1
Comparison of carbon price prediction algorithms.

Parameter	Economic methods	Machine learning methods
Data characteristics	Linear data	Nonlinear and non-stationary data
Accurate forecasting	Highly accurate and reliable in linear data	Produce more accurate forecasts
Time and resources	Faster computational speed	Training might take anything from a few hours to several days, depending on the complexity of the data that is supplied.
Advantages	Simple to understand and easy to execute.	• Deal with nonlinear data • Achieving promising results in carbon predictions
Drawbacks	• Because nonlinearity and non-stationary are difficult to control, prediction accuracy cannot be obtained. • Longer-term forecasts perform worse • Require more computational resources and time	Require a lot of data to be able to learn and make accurate forecasts.
Models	• ARIMA [16] • GARCH [15]	• GA-SVM [19] • PCA-DT [21] • PSO-LSVM [23] • SSA + LSTM [27] • ANFIS+EFO [28]

spectrum's direction, and the generation of the new colorful ray. **The initial population** of the white lights is reset at random to begin the LSO search process, which is calculated by [32].

$$\overrightarrow{y^0} = lB + VU_1(uB - lB) \tag{1}$$

where $\overrightarrow{y^0}$ is the initialization of the light spectrum's position, VU_1 is a vector made up of arbitrary values between 0 and 1., and lB is the lower boundary of the search space and uB is the upper boundary of the search space.

3.1.1. Rainbow spectrums direction

The normal vector of inner refraction $\overrightarrow{y_{nI}}$, inner reflection $\overrightarrow{y_{nO}}$, and outer refraction $\overrightarrow{y_{nR}}$, are determined after initialization as follows [33]:

$$\overrightarrow{y_{nI}} = \frac{\overrightarrow{y^{\gamma}{}_k}}{norm\ \left(\overrightarrow{x^{\gamma}{}_k}\right)} \tag{2}$$

$$\overrightarrow{y_{nO}} = \frac{\overrightarrow{y^{P}{}_k}}{norm\ \left(\overrightarrow{x_k}\right)} \tag{3}$$

$$\overrightarrow{y_{nR}} = \frac{\overrightarrow{y^{*}}}{norm\ \left(\overrightarrow{y^{*}}\right)} \tag{4}$$

where $\overrightarrow{y^{\gamma}{}_k}$ is a random selected position from the present population at $k-th$ iteration, $\overrightarrow{y^{P}{}_k}$ is the current position, $\overrightarrow{\boldsymbol{y}^{*}}$ is the best position obtained so far, and $norm(.)$ is a vector's normalization value, and it is computed using the following equation:

$$norm\left(\overrightarrow{y}\right) = \sqrt{\sum_{j=0}^{d} y_j^2} \tag{5}$$

where d represents the number of dimensions, $\overrightarrow{y}$ is the $norm$ function's input vector and y_j is the j^{th} dimension in the input vector. The following equations show the mathematical definition of the incident light ray $\overrightarrow{y_{L0}}$.

$$\overrightarrow{y_{L0}} = \frac{x_{mean}}{norm(x_{mean})} \tag{6}$$

where

$$x_{mean} = \frac{\sum_{i=1}^{N} \overrightarrow{y_i}}{N} \tag{7}$$

x_{mean} is the average of the current population of positions $\overrightarrow{y_i}$ and N is the population size.

The vectors for the outer refracted $(\overrightarrow{y_{L3}})$, inner reflected $(\overrightarrow{y_{L2}})$, and inner refracted rays $(\overrightarrow{y_{LI}})$ of reflected light are determined using the following equations [34]:

$$\overrightarrow{y_{LI}} = \frac{1}{k^{\gamma}}\left[\overrightarrow{y_{L0}} - \overrightarrow{y_{nI}}\left(\overrightarrow{y_{nI}}*\overrightarrow{y_{L0}}\right)\right] - \overrightarrow{y_{nI}}\left|1 - \frac{1}{(k^{\gamma})^2} + \frac{1}{(k^{\gamma})^2}\left(\overrightarrow{y_{nI}}*\overrightarrow{y_{L0}}\right)^2\right|^{\frac{1}{2}} \tag{8}$$

$$\overrightarrow{y_{L2}} = \overrightarrow{y_{LI}} - 2\overrightarrow{y_{nO}}\left(\overrightarrow{y_{LI}}*\overrightarrow{y_{nO}}\right) \tag{9}$$

$$\overrightarrow{y_{L3}} = k^{\gamma}\left[\overrightarrow{y_{L2}} - \overrightarrow{y_{nR}}\left(\overrightarrow{y_{nR}}*\overrightarrow{y_{L2}}\right)\right] + \overrightarrow{y_{nR}}\left|1 - (k^{\gamma})^2 + (k^{\gamma})^2\left(\overrightarrow{y_{nR}}*\overrightarrow{y_{L2}}\right)^2\right|^{\frac{1}{2}} \tag{10}$$

where k^{γ} represents the refractive index and it can be calculated as the following:

$$k^{\gamma} = k^{red} + VU_1\left(k^{violet} - k^{red}\right) \tag{11}$$

3.1.2. Generation the new colorful ray

Following the determination of the rays' directions, we determine the candidate solutions based on the value of p, a probability that is created at random and ranges from 0 to 1. Specifically, the new candidate solution will be determined if the P's value is less than a number chosen at random in [0,1].

$$\overrightarrow{y_{t+1}} = \overrightarrow{y_t} + \epsilon VU_1^n AC\left(\overrightarrow{y_{LI}} - \overrightarrow{y_{L3}}\right) \times \left(\overrightarrow{y_{rI}} - \overrightarrow{y_{r2}}\right) \tag{12}$$

Otherwise, the new potential solution will be computed as

$$\overrightarrow{y_{t+1}} = \overrightarrow{y_t} + \epsilon VU_2^n AC\left(\overrightarrow{y_{L2}} - \overrightarrow{y_{L3}}\right) \times \left(\overrightarrow{y_{r3}} - \overrightarrow{y_{r4}}\right) \tag{13}$$

where $\overrightarrow{y_{t+1}}$ is the new created candidate position, $\overrightarrow{y_t}$ is the current candidate position at $t\text{-}th$ iteration. $r1$, $r2$, $r3$, and $r4$ are four solutions that were chosen randomly from the current population. VU_1^n and VU_2^n are vectors of uniform random numbers. ϵ is a scaling factor and can be calculated using Eq. (14) while AC is an adaptive control factor and can be computed using Eq. (15).

$$\epsilon = w \times VU_3^n \tag{14}$$

$$AC = w \times q^{-1} \times P^{-1}(w, 1) \tag{15}$$

$$w = r5 \times \left(1 - \frac{t}{T_{Max}}\right) \tag{16}$$

where w is the adaptive parameter, P^{-1} represents the inverse incomplete gamma function for a given value w, where 1 value (the second parameter of the inverse incomplete gamma function) corresponds to the upper limit of integration, t is the present iteration, T_{Max} represents the most iterations possible and q and $r5$ are random number generated between [0,1].

3.2. Long short-term memory (LSTM)

Recurrent neural network (RNN) networks, such as the Long Short-Term Memory (LSTM) network, can be trained by Bayesian optimization [35]. To solve the vanishing gradient issue, LSTM incorporates a cell state vector. There are three distinct kinds of gates in the LSTM model. Forget Gate f_t is the name of the first gate and it establishes the amount of data that can be extracted from the prior state. It uses the sigmoid function to create a value between 0 and 1 by taking the current input (x_t) from the current state and the prior state h_{t-1}.

$$f_t = \sigma\left(W^f(\ x_t \bullet h_{t-1}) + b_f\ \right) \tag{17}$$

where b_f represents bias and W^f represents weight.

The second gate, also referred to as the input or update gate (C_t), establishes whether the data needs to be updated. For the purpose of determining the cell state C_t, the state g_t computed the value between -1 and 1 and it is merged with i_t and then multiplied f_t and the previous state C_{t-1}.

$$C_t = f_t \bullet C_{t-1} + i_t \bullet g_t \tag{18}$$

where,

$$i_t = \sigma\left(W^i(\ x_t \bullet h_{t-1}) + b_i\ \right) \tag{19}$$

$$g_t = tanh\left(W^c(\ x_t \bullet h_{t-1}) + b_c\ \right) \tag{20}$$

The last gate selects the model's output by using the sigmoid layer to determine which cell state segments are taken. Then, apply the tanh function on the cell state, changing the values between -1 and 1.

$$o_t = \sigma(w_o \bullet [h_{t-1}, x_t] + b_o) \quad (21)$$

$$h_t = o_t \bullet tanh (C_t) \quad (22)$$

4. Dataset characteristics

The used carbon price historical transaction data was acquired from Investing.com (https://www.investing.com/equities/keycorp-new-historical-data). The acquired data was collected from 25/04/2005 to 25/04/2023. It consists of seven features. These features are transaction date, low, high, open, change, volume, and price. The Low and High features show the price range in which the carbon is traded during a specific trading session. They include details on the price swings that occurred over that time interval as well as the price volatility. Investors and traders can use this information to evaluate the possible risks and benefits of trading carbon. The first traded price of the day, known as the "open," is important for understanding the current state of the market at that moment. It can aid in determining if the market has a bullish (positive) or bearish (negative) tilt at the outset. The direction and magnitude of price fluctuations can be determined by looking at the percentage change from the previous trading session. It aids in determining how far the price has moved from the closing of the previous day and can be a crucial signal for short-term traders. The level of market participation and liquidity is represented by the volume characteristic. Lower volumes may indicate poor interest or potential price manipulation, whereas higher volumes frequently reflect robust market demand. Volume plays a critical role in validating price fluctuations. Fig. 1 shows the carbon price value over a time interval starting from 25/04/2005 to 25/04/2023. The graph prominently showcases a steep and consistent upward trajectory in carbon prices starting from 2018, commencing from the starting point five years ago and persisting through the present day. The ascending curve is indicative of a robust and unyielding upward movement in pricing, underlining the vigor and intensity of this market transformation.

Fig. 2 demonstrates an in-depth investigation of the average, standard deviation, minimum, and maximum prices over time intervals starting from 2005 to 2023, highlighting notable trends and fluctuations in the data. As can be seen, a steady decline in the value of the commodity from 2005 to 2012 can be seen as a downtrend. But from 2012 to 2017, there is a significant stable pricing pattern. The market dynamics at this time and various economic factors can be blamed for this decline. Around 2018, an interesting trend reversal took place. Following a protracted price decline from 2012 to 2017, the commodity's value has started to increase. The maximum and average price lines both show a consistent ascent, pointing to a renewed upward trajectory.

Another experiment is being conducted to show how the month and the day, not just the year, can carry valuable information about the price of carbon. Fig. 3 shows the total sum of carbon prices per month and day. Through this, seasonality and trends can be detected. Fig. 3 (A) shows the month's influence on the carbon price. As can be observed, almost June has a high influence on the price of carbon. Finally, Fig. 3 (B) shows the influence of the day on the carbon price. As can be observed, the working days from Monday to Friday greatly influence the carbon price, despite the weekend.

5. The proposed carbon price predication model

Three major phases make up the proposed carbon price prediction model. As seen in Fig. 4, these phases include data preprocessing, hyper-parameters optimization, evaluation, and result interpretation. It is mainly based on utilizing custom LSTM architecture. The proposed custom LSTM is comprised of four layers, namely the regression output layer, dense layer, LSTM layer, and sequence input layer. Table 2 shows the layers' structure of the proposed architecture with parameter setting values per layer. For sequence data, such as time series or text sequences, the first layer, called the input sequence layer, is used. A set of feature vectors are fed into this layer. This layer's input size is six in this paper, which corresponds to the number of features. The proposed architecture's second layer is called long short-term memory (LSTM), which is a type of recurrent neural network (RNN) layer. It processes data in sequential order and preserves hidden states over time steps. The series of feature vectors from the preceding layer is used as input for this layer, while the hidden state of the LSTM at the most recent time step is used as output. An activation function is applied to the output of the previous LSTM layer by the third layer, known as the dense layer, after a linear transformation. When performing regression tasks, the last layer is employed in order to predict a continuous value. It calculates the difference in output between the predicted and actual outputs. However, the LSTM layer has shown great performance for regression tasks [15,25,28], and tuning the hyper-parameters values can significantly affect the performance of the overall proposed carbon price prediction model. Thus, this paper proposes a modified version of the light spectrum optimizer (MLSO) to find the optimal hyper-parameters values of the proposed custom LSTM architecture. The proposed custom LSTM architecture is then trained using the ideal hyper-parameters that were chosen in the preceding phase. Finally, explainable artificial intelligence (XAI) and a number of measurements are used to assess the effectiveness of the proposed model throughout the evaluation and results interpretation phase. The following sections will provide a thorough explanation of every phase of the proposed model.

5.1. Data preprocessing phase

During the data preprocessing phase, the intrinsic quality of raw historical transaction data is enhanced for accurate carbon price predictions. First, the mean value imputation approach is applied to tackle the missing values problem. This approach replaces missing data points with the mean of the relevant feature. This reduces the chance of adding bias due to missing values while preserving the overall statistical coherence of the dataset. Data normalization is then used to standardize the distribution and diversity of features. This normalization technique makes sure that each variable contributes adequately to the regression model's learning process, regardless of the variations in their numerical magnitudes. Next, the train subset, test subset, and valid subset are

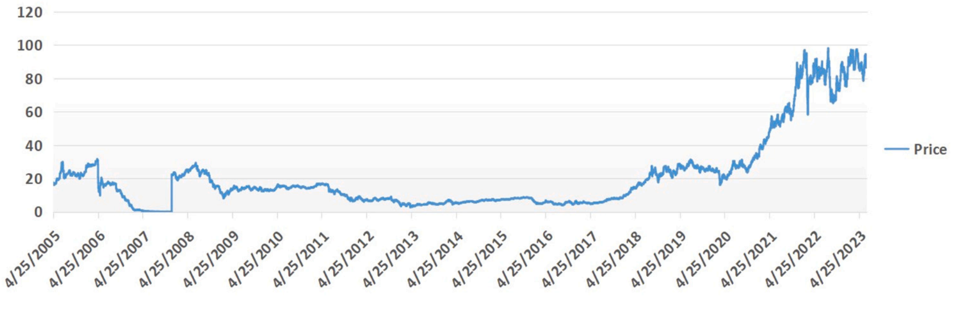


Fig. 1. Carbon price values from 2005 to 2023.

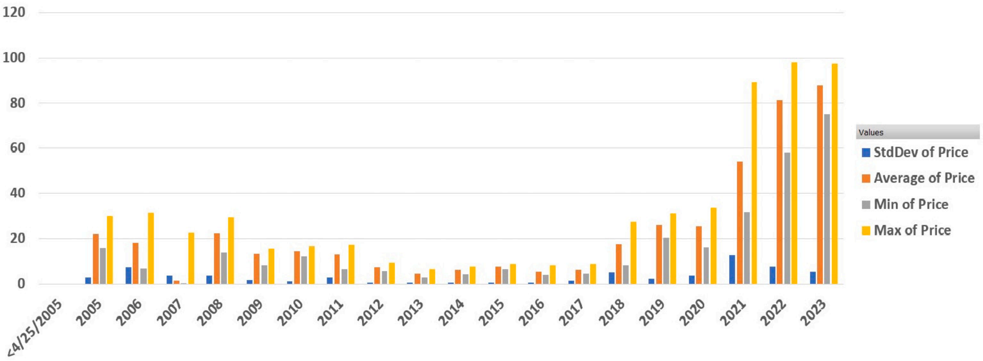


Fig. 2. Carbon price fluctuations pattern over time (2005–2023).

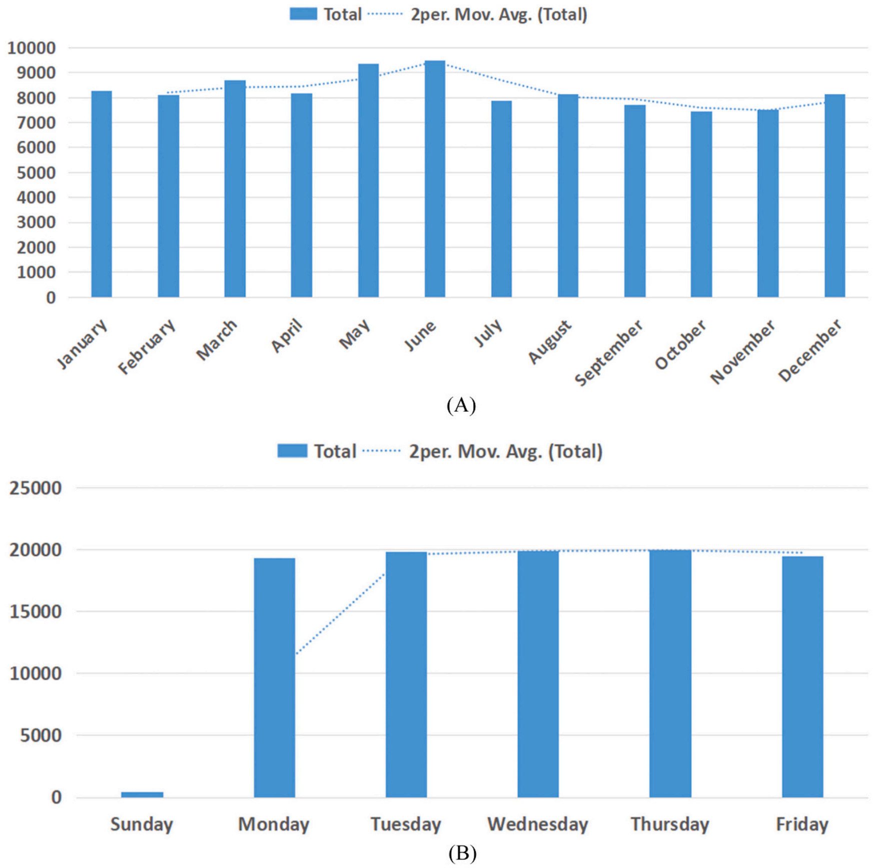


Fig. 3. The influence of the month, and day for the carbon price. (B) Sum of price vs. the month, and (C) Sum of price vs. the day.

created by randomly dividing the processed data. Seventy percent of the total data samples are in the training subset, and 15 % are in the test and valid subsets, respectively. The following phase involves determining the ideal batch size, learning rate, and solver optimization techniques for the proposed custom/optimized LSTM architecture using the train and valid subsets as input. Lastly, the proposed carbon prediction model's overall performance is assessed using the test subset.

5.2. Hyper parameters optimization of LSTM using MLSO phase

During this phase, a modified version of LSO algorithm is proposed in order to optimize the custom LSTM architecture's hyper-parameters. As seen in Table 2, the proposed architecture has a significant layer that highly influences the performance of the model. This layer is LSTM. This paper considered finding the optimal batch size value, the learning rate value, and the best solver optimization algorithm. The efficiency and performance of the proposed custom LSTM architecture are greatly influenced by these parameters. A model can converge more quickly and achieve superior accuracy and generalization on unobserved data if hyper-parameters are carefully selected. On the other hand, poorly selected hyper-parameters may cause underfitting or overfitting, slow convergence, or poor performance in general. Swarm optimization can greatly speed up the process of finding the optimal hyper-parameters when used in hyper-parameters tuning. Navigating through the countless possible combinations of hyper-parameters automates the search, decreases the need for manual trial and error, and allows the model to reach its full potential [36]. Next, the proposed MLSO-based hyper-parameters optimization of LSTM is described in detail.

The proposed MLSO-based hyper-parameters optimization of LSTM first starts initializing the population using Eq. (1), the maximum

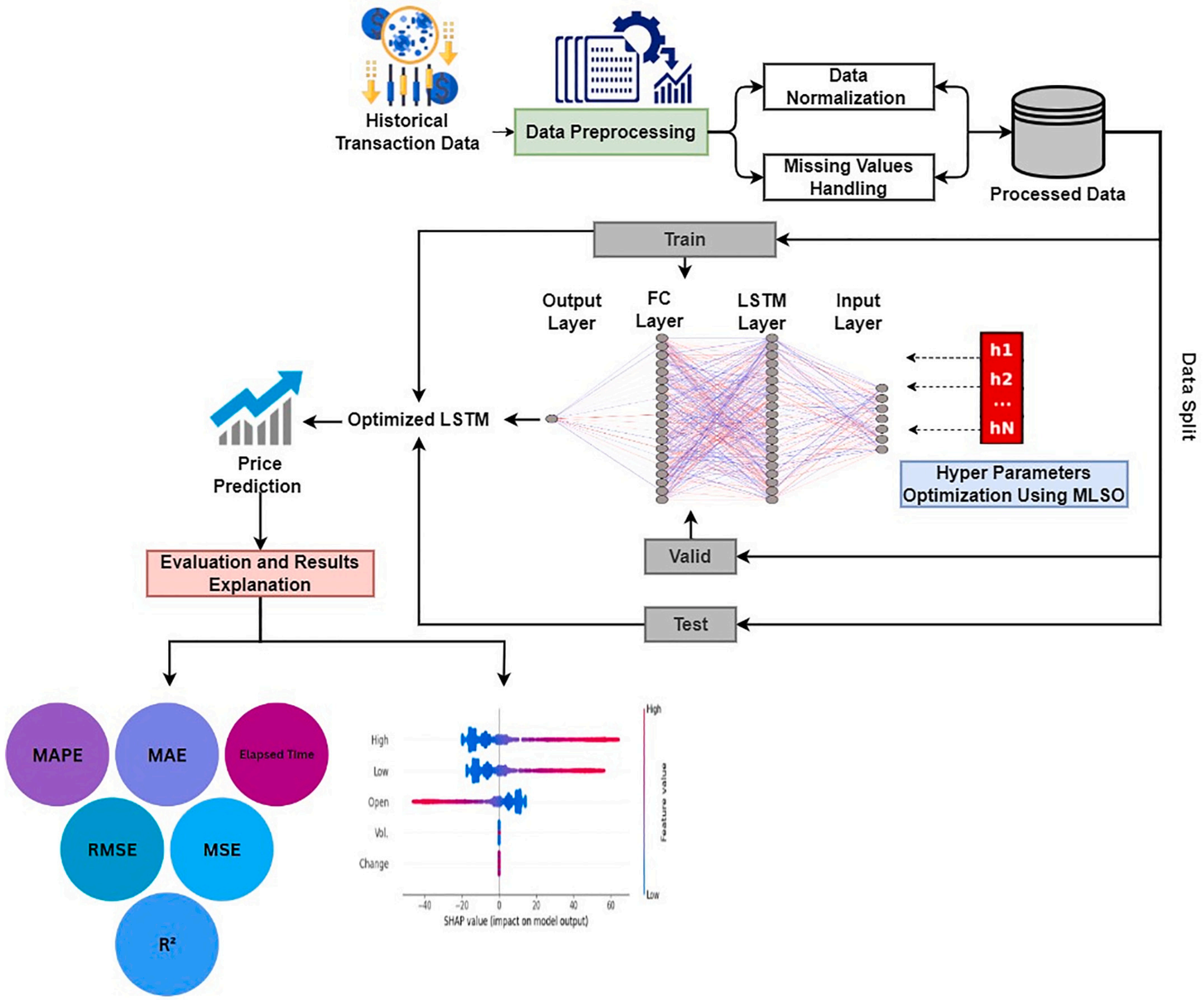


Fig. 4. The proposed carbon price prediction model.

Table 2
The proposed optimized LSTM architecture's layer structure.

Layer	Parameters
Sequence input layer	Input size = 6
LSTM	Units = 64, Activation='RelU', Output Mode = 'last'
Dense layer	Units = 1, Activation='linear'
The regression output layer	Loss='MSE'

number of iterations, the probability of first and second scattering stages, the controlling parameter to switch between the initial and secondary scattering, the probability of hybridization between two various search boundary methods, the exploitation probability in the first scattering stage, the search boundary, and the dimension length. The proposed MLSO-based hyper-parameter optimization of the LSTM algorithm is proposed to optimize the batch size, learning rate, and the finding of the optimal solver optimization algorithm. The dimension length is set to three. The lower and upper limits for the batch size are set to [1256], while the lower and upper limit for the learning rate is set to [0.000001,0.01]. Finally, the proposed algorithm is used to determine the optimal solver optimization algorithm between Adaptive Moment Estimation (Adam), Stochastic Gradient Descent with momentum (SGD), Root Mean Square Propagation (RMSprop), and Adaptive Moment Estimation (Adam). To accelerate the random generation of a solver optimization algorithm, each option is set to a numerical value; Adam is set to 1, SGD is set to 2, and RMSprop is set to 3. Each individual

Table 3
The assigned value of the proposed MLSO based hyper parameter optimization of LSTM.

Parameter	Assigned value
The probability of first and second scattering stages	0.05
The controlling parameter to exchange between the first and second scattering	0.6
The probability of Hybridization between two various search boundary methods	0.4
The exploitation probability in the first scattering stage	0.05
Population size	70
Maximum number of iterations	100
Lower search boundary	[1,0.000001,1]
Upper search boundary	[256,0.01,3]
Dimension length	3

has a different value of batch size, learning, and solver option. Table 3 shows the assigned parameter setting of the proposed MLSO-based hyper-parameters optimization of the custom LSTM architecture.

Each individual is assessed using a predetermined fitness function during the optimization process. To assess how far each light spectrum (individual) is good, a fitness function is employed. The fitness function in this paper is mean square error. Eq. (23) provides the mathematical definition of the fitness function that is being used, where $\overline{X}_j$ represents the predicted value for the *j-th* valid data point, X_j represents the actual value for the *j-th* valid data point, and N is the total number of samples in the valid subset data.

$$F(\overrightarrow{y_{t+1}}) = \frac{1}{N}\sum_{j=1}^{M}\left(\overline{X}_j - X_j\right)^2 \tag{23}$$

$$F^{*} = Minimize(F(\overrightarrow{y_{t+1}})) \tag{24}$$

The best individual is the one that has the minimum fitness value F^{*}. Then, through the curse of iteration, each individual updates its position using Eq. (12) and Eq. (13). This procedure is carried out again until the maximum number of iterations is reached. In our case, when the algorithm reaches the *100th* iteration. Finally, the optimal individual with it corresponding fitness value is reported. After finding the optimal batch size, learning, and solver optimization algorithm using the proposed MLSO, the proposed custom LSTM architecture is trained using the training subset. Lastly, the test subset is used to evaluate the proposed architecture's performance.

5.3. Evaluation and results explanation phase

In the process of model evolution, the performance of the proposed model is assessed using different evaluation metrics. These evaluation metrics play a crucial role when contrasting various regression models rather than relying just on one. Utilizing several measures can give us a complete idea of how each model is functioning because different metrics offer various insights into the performance of a model. In this paper, five different measurements are utilized. These measurements are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R^2), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The mean squared error (MSE) is a measurement of the discrepancy between the predicted and actual values. Greater errors are given more weight. An improved model's performance is shown by a reduced MSE. As opposed to this, the RMSE is the MSE's square root. The target variable's units are used to express the average error magnitude. An improvement in model performance is indicated by a reduced RMSE, as with MSE. Due to its easier interpretation in light of the original data, RMSE is frequently favored over MSE. R^2 is a statistical expression that indicates how much of the variance in the dependent variable in a regression model is explained by the independent variables. It ranges from 0 to 1, with 1 denoting that the model completely accounts for the data's variability. A higher R^2 value is preferable because it indicates that the model fits the data more accurately. The MAE calculates the average absolute differences between the predicted and actual values. It is less sensitive to outliers and gives all errors the same weight. A lower MAE is preferable, like other metrics. The average percentage difference between the predicted and actual values, relative to the actual values, is then calculated by the MAPE. When dealing with data of various scales, it is frequently used. Better model accuracy is indicated by a lower MAPE. Eqs. (25), (26), (27), (28), and (29) provide a mathematical explanation for each assessment measure that has been adopted, where P_j represents the *j-th* data point's predicted value, A_j shows the *j-th* data point's actual value, *M* is the total number of samples, and $\overline{A}_j$ is the mean of the actual value.

$$RMSE = \sqrt{\sum_{j=1}^{M}\frac{\left(P_j - A_j\right)^2}{M}} \tag{25}$$

$$R^2 = 1 - \frac{\sum_{j=1}^{M}\left(P_j - A_j\right)^2}{\sum_{j=1}^{M}\left(A_j - \overline{A}_j\right)^2} \tag{26}$$

$$MAE = \frac{1}{M}\sum_{j=1}^{M}\left|P_j - A_j\right| \tag{27}$$

$$MAPE = \frac{1}{M}\sum_{j=1}^{M}\frac{\left|P_j - A_j\right|}{A_j} \tag{28}$$

$$MSE = \frac{1}{M}\sum_{j=1}^{M}\left(P_j - A_j\right)^2 \tag{29}$$

Finally, eXplainable AI (XAI) is used for the result explanation. In the areas of model evolution and interpretation, XAI has been regarded as an essential element. It has been difficult to interpret AI models, especially those with high levels of complexity like deep neural networks. Here, XAI acts as a lighthouse, opening the black box and revealing model behavior. XAI techniques are utilized to better understand the reasoning behind AI decisions by generating explanations for model predictions. A comprehensive understanding of how input data is transformed into output predictions is provided by XAI through visualizations, feature importance attribution, and interactive interfaces [37]. In this paper, SHAP (Shapley Additive exPlanations), one of the XAI techniques, is adopted to explain the result of the proposed MLSO-LSTM. SHAP values provide a precise and comprehensible understanding of how input features affect model outcomes by quantifying the significance of each feature in the prediction process.

6. Experimental results and discussion

Several experiments were carried out to assess the overall effectiveness of the proposed carbon price prediction model in this section. The first experiment evaluates the effectiveness of the proposed model for predicting carbon prices both before and after hyper-parameter optimization using MLSO. The main objective of this experiment is to assess whether MLSO improves model accuracy and robustness. This experiment also compares the MLSO-based hyper-parameters optimization algorithm to five different swarm-based hyper-parameters optimization algorithms for LSTM networks to evaluate how effective they are in comparison. MSE, RMSE, R^2, MAE, MAPE, and elapsed time are the used evaluation metrics for providing a comprehensive review. These metrics provide insight into the model's prediction accuracy and error distribution. In addition, the experiment presents the optimal batch size, learning rate, and solver optimizer hyper-parameters of LSTM for each competing algorithm, providing thorough details on how each optimization technique affects model performance. The second experiment aims to interpret how the proposed model forecasts carbon prices for carbon price historical transaction dataset acquired from Investing.com using the SHAP method. This involves generating SHAP visualizations to help understand the model's results. The SHAP summary plot shows the relative relevance of features across all predictions, whereas the SHAP values plot shows how each feature contributes to individual predictions. Furthermore, the SHAP force plot is utilized to show how individual feature values influence predictions for specific instances. The third experiment aims to compare the performance of the proposed model, which employs optimized LSTM, to other well-known machine learning algorithms. The algorithms are Random Forest and Gradient Boosting. This comparison compares the performance of the optimized LSTM model to conventional machine learning algorithms, providing insights into its effectiveness and potential advantages in projecting carbon pricing. The fourth experiment aims to examine the performance and robustness of the proposed carbon price prediction model using eight benchmark datasets. These include six Kaggle datasets and two recently published carbon pricing datasets (BeijingETS and HubeiETS). The experiment examines each dataset's distinct properties and assesses the model using a variety of performance metrics, including RMSE, MSE, MAE, MAPE, and R-squared. Furthermore, the model's performance is compared to that of two other machine learning algorithms to provide a full assessment of its efficacy in various settings. The fifth and final experiment aims to assess the performance of the proposed model by comparing it to state-of-the-art models. This comparison is based on key metrics including RMSE, MSE, and R-squared. Finding out how well the proposed model performs in comparison to existing models will shed light on its efficacy and its benefits. It should be mentioned that the proposed model is implemented on an Intel Core I7 computer running at 2.6 GHz with 16 GB of RAM using the MATLAB 2020 tool. To process the data and interpret the results, the Python programming language is used in conjunction with the Sklearn, Keras, and Pandas libraries [38]. To ensure fair comparisons, all of the comparative models were implemented on the same platform and with the same programming language. In all the conducted results, the best results are underlined.

The purpose of Table 4 is to compare the effectiveness of the proposed MLSO-based hyper-parameters optimization of the optimized LSTM architecture before and after MLSO is employed. Additionally, it compares the performance of the proposed MLSO-based hyper-parameters optimization of LSTM with other recent and well-known swarms' optimization-based hyper-parameters optimization of LSTM. These algorithms are Modified Salp Swarm Algorithm (MSSA) based hyper-parameters optimization of LSTM [36], Modified Whale Optimization Algorithm (MWOA) based hyper-parameters optimization of LSTM [39], Modified African Vulture Optimization Algorithm (MAVOA) based hyper-parameters optimization of LSTM [40], Modified Grey Wolf Optimizer (MGWO) based hyper-parameters optimization of LSTM [41], and Modified Differential Evolution (MDE) based hyper-parameters optimization of LSTM [42]. It is worth noting that this experiment used the default values of parameters linked to each hyper-parameter optimization algorithm. The remaining parameters, including population size, maximum iteration count, dimension length, search boundaries, and fitness value, were kept constant across all these algorithms to ensure a fair comparison. As can be observed, tuning the batch size, learning rate, and the solver parameters can significantly boost the performance of the proposed carbon price model. As all the adopted swarm optimization algorithms significantly produce lower RMSE, MAE, MAPE, and MSE with higher R^2 compared with the original LSTM with its default parameters settings. Another significant finding was that the proposed MLSO-based hyper-parameters optimization for LSTM produced better results than other hyper-parameters optimization algorithms. This shows that the MPSO algorithm is more effective at improving the performance of the LSTM model, resulting in higher prediction accuracy and overall model efficiency.

Table 5 shows the ideal hyper-parameters for the proposed custom LSTM architecture, aligned with the fitness functions that are measured using mean square error. This table compares the proposed MLSO-based hyper-parameters optimization performance with MSSA, MWOA, MAVOA, MEO, MGWO, and MDE-based hyper-parameters optimization algorithms for accurately predicting the carbon price. In this experiment, to make a fair comparison, the same preprocessing to the adopted dataset is used with the same population size, dimension length, fitness function, and maximum number of iterations. As can be seen, the results highlight the strength of the proposed MLSO-based hyper-parameter optimization algorithm, which stands out as a highly competitive choice when compared to other swarm-based hyper-parameter optimization algorithms. This result is relevant to the result obtained in Table 4. It should be mentioned that the optimal batch size, the learning rate, and the solver algorithm selected from MLSO will be further used in the upcoming experiments.

Table 5
The optimal hyper-parameters setting with its corresponding fitness value of the proposed carbon prediction model using MLSO, and other hyper-parameters setting optimization-based swarm intelligence algorithms.

	Batch size	Learning rate	Solver	Fitness value
MLSO	8	0.0052	SGD	0.8264
MSSA	1	0.01	Adam	0.5145
MWOA	1	0.0088	SGD	1.0584
MAVOA	8	0.0053	Adam	0.8882
MEO	1	0.0072	Adam	0.8546
MGWO	1	0.0056	Adam	1.0470
MDE	64	0.0079	SGD	3.4455

For further evaluation of the performance, another measurement is adopted. This measure is the elapsed time. Fig. 5 provides a thorough comparison of the proposed MLSO-based hyper-parameters optimization algorithm with other swarm-based hyper-parameters optimization algorithms, with an emphasis on the amount of time needed for the optimization process. There were significant insights into the effectiveness and computing performance of each algorithm by monitoring the time it took for it to converge toward optimal solutions. These results

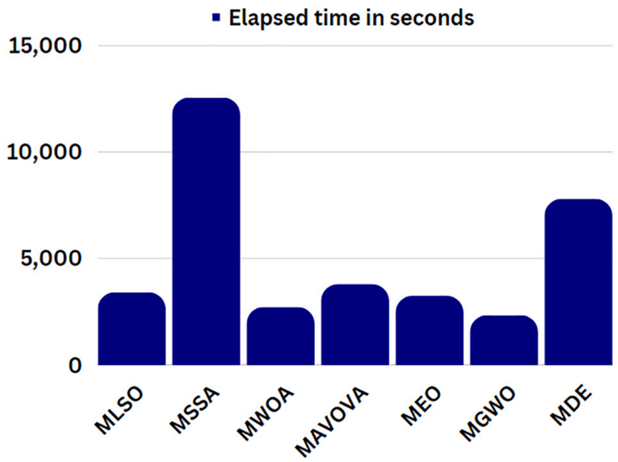


Fig. 5. The proposed carbon prediction model using MLSO and other swarm optimization algorithms in terms of elapsed time in second.

Table 4
The proposed carbon prediction model using LSTM without hyper-parameters optimization and with using MLSO and other swarm optimization algorithms in terms of RMSE, MAE, MSE, MAPE, and R^2.

	LSTM	MLSO + LSTM	MSSA + LSTM	MWOA + LSTM	MAVOA + LSTM	MGWO + LSTM	MDE + LSTM
RMSE	19.3317	0.6644031	1.1549753	1.0858660	0.7522277	1.0846496	0.8719593
R^2	0.3619	0.9991269	0.9972751	0.9977263	0.9989029	0.9976506	0.998492
MAE	14.7869	0.3780583	0.3944538	0.4535912	0.3303106	0.4897030	0.5189371
MAPE	1.4149	0.1520864	0.1142464	0.0872333	0.1461522	0.2574679	0.4974064
MSE	373.7139	0.4414315	1.3339680	1.1791051	0.5658466	1.1764648	0.7603131

clarify the trade-offs among various optimization algorithms, giving a clearer comprehension of their capabilities and time-related advantages in the context of hyper-parameters optimization. As shown in the figure, the proposed MLSO-based hyper-parameters optimization algorithm is highly effective in predicting carbon prices. It excels over other algorithms in both speed and accuracy by quickly determining the ideal hyper-parameter values. This efficiency not only improves the model's predictive performance but also greatly reduces the time necessary for hyper-parameter tuning, resulting in more timely and accurate carbon price predictions.

In Fig. 6, the effect of the proposed MLSO-based hyper-parameters optimization algorithm on precise carbon price prediction is vividly depicted. The values that are closely matched and near to the actual values demonstrate the MLSO algorithm's extraordinary ability to converge toward outcomes that closely match actual values. This high level of accuracy extends straight into the field of carbon price forecasting, where precise hyper-parameters are crucial for improving the forecasting model. The carbon price prediction model exhibits improved accuracy by utilizing the MLSO-based optimization, providing important insights for well-informed decision-making in carbon markets and sustainability planning.

In the experiment in Fig. 7, the data is split into training and testing sets, with 80 % of the data being used for training and the remaining 20 % being used for testing. Two random rows are chosen as test data for each distinct year in the dataset. The remaining rows for every year are utilized as training data. This guarantees that the model gains knowledge from multiple instances of the data from each year. The proposed model is trained using training data and assessed using test data. To assess how effectively the model predicts the target values on test data, performance measures like RMSE, MAE, and MAPE are computed. As can be observed, when MAE and RMSE are compared from 2009 to 2017, a constant and little difference in predicting accuracy is seen. The proposed model performs consistently during this period, demonstrating a trustworthy assessment of the target variable. The tight agreement between the MAE and RMSE values indicates that the model's predictions and actual observations are highly correlated, and the model's mistakes are modest and reliable. But starting in 2017, there has been a noticeable change in the model's performance. The increased trajectory of both MAE and RMSE values aligns with the rise in carbon pricing, as seen in Figs. 1 and 2. This divergence denotes a shift in the model's capacity to make correct predictions, which may be related to the evolving dynamics of the dataset. From 2017 to 2023, the growing discrepancy between MAE and RMSE indicates that prediction mistakes have gotten more variable, with RMSE indicating greater variances. This pattern may be impacted by external factors such as economic fluctuations, evolving government rules, and the impact of the COVID-19 pandemic on the carbon price market [30]. The pandemic has created tremendous volatility and uncertainty, which is expected to worsen forecast mistakes. As carbon prices have risen, the model's predictive ability has decreased, emphasizing the significance of updating the model to account for market changes and future uncertainties, such as those caused by global events like COVID-19.

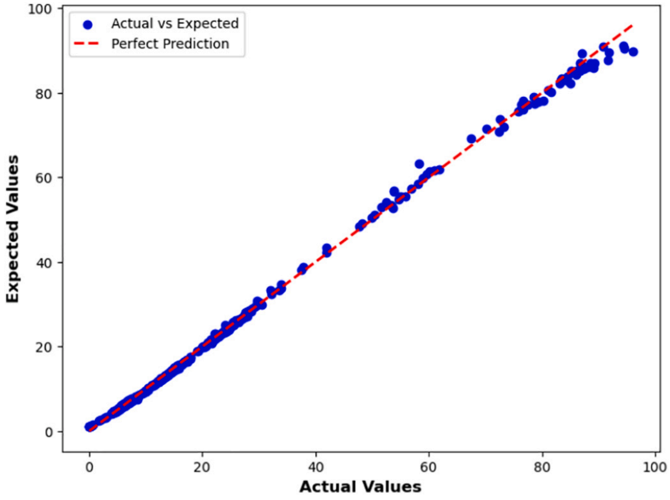


Fig. 6. Predicted values from the proposed model vs. the actual values.

SHAP analysis is utilized to interpret how the proposed model is used to accurately predict the carbon price that day given the input features. For the SHAP plot, the feature's impact on the overall model performance is sorted, where the feature with a high impact on the overall model performance appears first. The plotted color point in the SHAP plot indicates the test sample. The colored point's gradation from blue (low values) to red (high values) indicates the actual value of the test sample data. The visual depiction of feature relevance in connection to price dynamics is powerful in the SHAP plot summary. In Fig. 8, the date feature is split into year, month, and day to investigate whether the year or month, or day affects carbon prices. As seen from this figure, low and high price features—which show the carbon price range during a trading session—are represented by red dots that go toward negative SHAP values, unique patterns may be seen in the plot. Similarly to that, the red-colored dots that reflect the opening prices of the trading day also line up with negative SHAP values. These strong patterns highlight how important these particular characteristics are for affecting price changes. The SHAP plot highlights the increased significance of low price, high price, and open price features in driving the model's predictions in comparison to other features like date, volume, and price change features. This highlights their dominant role in determining the overall price behavior and offers crucial insights for thorough pricing analysis and decision-making processes. Additionally, it can be observed that the carbon price is unaffected by variations in the year, day, or month.

Another privilege of utilizing SHAP analysis for the model's interpretation is that it can be applied to specific test instances. This allows us to determine which features have a significant impact on the proposed model's performance for a single test case. Fig. 9 shows the SHAP plot for the first instance in the test data. In this figure, feature 0 indicates the year feature, feature 1 indicates the month feature, feature 2 indicates the day feature, feature 3 indicates the open price feature, feature 4 indicates the high price feature, feature 5 indicates the low price feature, feature 6 indicates the volume feature, and finally feature 7 indicates the change feature. As observed, the highest influenced features are low, high, and open features. Another finding, neither the price change feature nor the date has an influence on the performance of the proposed carbon price prediction model. This finding is consistent with the obtained result in Fig. 8.

Fig. 10 shows the SHAP force plot for an individual test that demonstrated how each feature contributed to the final prediction, making it clear which features drove the model's decisions at different time steps. As can be observed low price, high price, and open price features have the highest impact on the prediction of the carbon price. The features called “open price,” “high price,” and “low price” appear to have the greatest influence on predicting carbon pricing. The feature known as “low price” refers to the lowest values attained during trading, representing the minimum trading value of carbon prices during a day. The greatest trade value, which reflects the highest levels attained, is represented by the “high price.” The “open price” denotes the value of trade at the beginning of the day, establishing the current state of the market and serving as a benchmark for daily price changes. Together, these features capture critical market dynamics, providing a full perspective of price variations and trading behaviors. Their significant influence on the prediction model emphasizes how crucial they are for precisely identifying and predicting the patterns and fluctuations in the carbon market. With this knowledge, more accurate predictions and well-informed decisions based on the model's results are possible.

The proposed optimized LSTM architecture is evaluated against several well-known machine learning models. These models are random

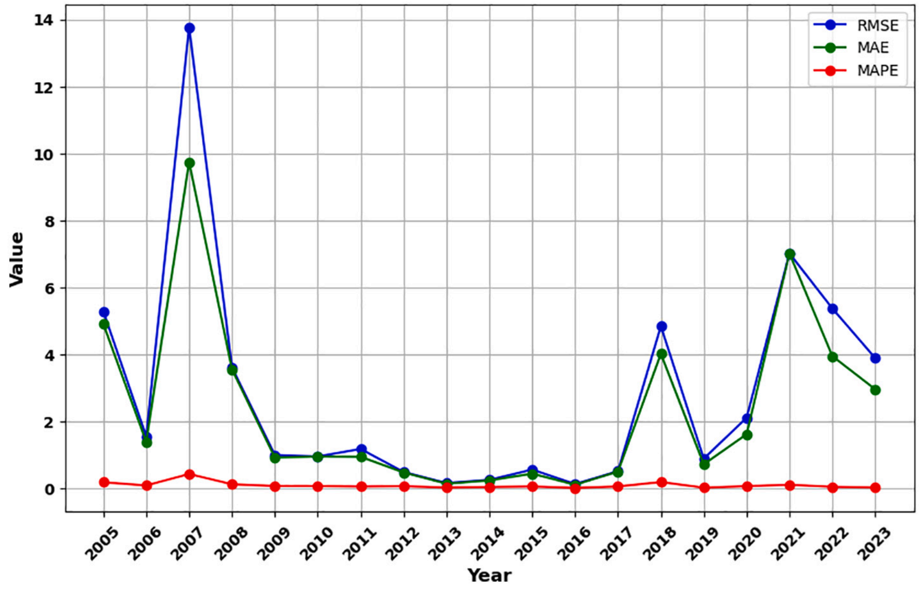


Fig. 7. RMSE, MAE, and MAPE over years.

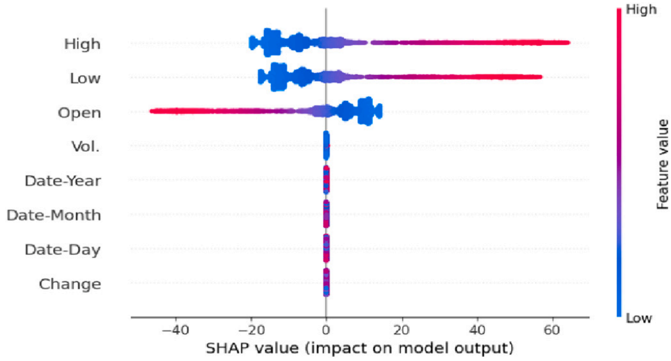


Fig. 8. SHAP summary plot.

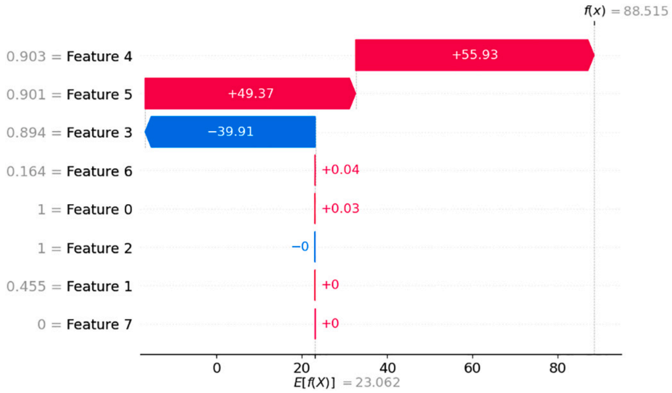


Fig. 9. SHAP values plot for the first instance in the test data.

forest and gradient-boosting regression models. When compared to the outcomes achieved from the Random Forest and Gradient Boosting regression models, using the proposed model for accurate carbon price prediction produces outstanding results as shown in Table 6. MSE, RMSE, MAE, R^2, and MAPE are important performance indicators used in this experiment. Notably, the MSE, MAE, MAPE, and RMSE values for the deep neural networks (DNN) show a significant decrease with a significant increase in terms of R^2, demonstrating the MLSO-LSTM's improved predicting skills. This improved accuracy is a testament to the DNN's ability to capture complex nonlinear correlations in the data, allowing it to more effectively identify and simulate complicated carbon pricing patterns. In contrast, the Random Forest and Gradient Boosting models produce decent outcomes, although their MSE and RMSE values continue to be comparatively higher. This demonstrates the DNN's capacity to provide a more accurate and exact prediction of carbon pricing, highlighting its potential to enhance the study of carbon market forecasting and help people make better decisions in the field of sustainable economics.

To further evaluate the efficiency of the proposed model, two additional benchmark datasets are utilized for validation. The first dataset, referred to as “BeijingETS,” spans from June 4th, 2015, to April 11th, 2024 with the total number of records equal to 1545 [43]. The other dataset denoted as “HubeiETS”, spans from April 6, 2017, to April 12, 2024, with the total number of records equal to 1641 [44]. Fig. 11 shows

Table 6
The performance of the proposed carbon prediction model using MLSO and custom LSTM vs using random forest and gradient boosting regression models.

	MLSO + LSTM	Random Forest	Gradient Boosting
RMSE	0.664391134	7.147564475	7.105414707
R^2	0.999126944	0.494156331	0.494725953
MAE	0.378058307	2.294760587	2.322754518
MAPE	0.152086465	37.9354649	43.73074147
MSE	0.441431558	51.08767793	50.48691817

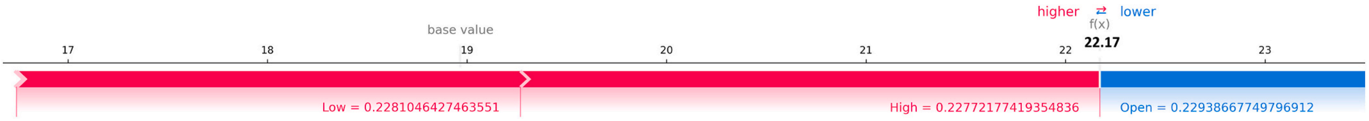


Fig. 10. SHAP force plot for the first instance in the test data.

the sum of carbon prices over the years for both BeijingETS and HubeiETS datasets. As can be observed, there is a continuous increase in carbon prices until the end of 2023. However, since 2024 is not yet complete, it cannot be ascertained whether the fluctuation pattern will continue to increase or decrease. Nevertheless, the trend over the years indicates a consistent upward trajectory.

Table 7 compares the performance of the proposed carbon prediction model using MLSO and custom LSTM in terms of RMSE, MAE, R^2, MAPE, and MSE for BeijingETS and HubeiETS. As can be observed, the performance of the proposed carbon prediction model appears highly promising, demonstrating its effectiveness in forecasting carbon prices. These results align consistently with findings from previous experiments, further validating the reliability and accuracy of the model's predictions.

An additional experiment was carried out to provide a more thorough assessment of the carbon price prediction model's performance. Six different benchmark datasets from the UCI machine learning repository were employed in this experiment; each dataset had distinct characteristics. The goal was to assess the model's robustness and adaptability across various types of data. Table 8 offers a comprehensive summary of every dataset, comprising the name, the number of features, and the overall count of records. This comprehensive evaluation aids in better understanding the model's efficacy and potential applications.

Table 9 compares the performance of the proposed carbon price prediction model based on utilizing LSTM and ALSO algorithms with random forest and gradient boosting algorithms. The results of this experiment were averaged over ten different runs using the k-fold cross-validation method. The data was divided randomly into ten separate folds, each with a different training and testing set. This approach ensures that the model is evaluated thoroughly and unbiasedly. The proposed model, which incorporates LSTM and MLSO, outperformed the random forest and gradient boosting methods. Metrics used for evaluation included mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and R-squared (R^2). The findings revealed that the proposed model consistently attained optimal values across these metrics, highlighting its effectiveness and competitiveness.

In the last experiment conducted in Table 10, the performance of the proposed model is compared with the state-of-the-art model that has already been published for carbon price prediction such as principle component analysis (PCA) with decision tree (DT), ensemble tree, Gaussian process regression model (GPR), singular spectrum analysis (SSA), extremely randomized trees (ET), multivariate variational mode decomposition (MVMD), bidirectional long short-term memory (BiLSTM) integrated to completely ensemble empirical mode decomposition with adaptive noise (CEEMDAN) with improved particle swarm optimization (IPSO) and variational mode decomposition (VMD), Adaptive Neuro-Fuzzy Inference System (ANFIS) with improved Electric Fish Optimization (IEFO), Wasser-stein GAN with Gradient Penalty (WGANGP) and LSTM. It should be mentioned that most of the models used a private dataset, however, this paper considers using publicly available datasets. The reported results in Table 10 compare MSE, RMSE, and R^2 with the used model for all published models for carbon price prediction. This table shows that the proposed carbon price prediction model outperformed the other models. As it obtained the minimum MSE and RMSE with the highest R^2. Additionally, it can be observed that most of the models that used LSTM obtained better results than those that used classical machine learning algorithms. From all the conducted results, it can be demonstrated that the proposed model's ability to correctly forecast carbon pricing has profound consequences for sustainability and decision-making. By utilizing cutting-edge computational methods such as LSO and LSTM, the model can remarkably capture the subtle nuances of carbon price fluctuations. This improved accuracy not only gives stakeholders the foresight to predict price changes more accurately but also makes it easier for stakeholders in different industries to make educated decisions. Such well-informed choices are essential for maximizing resource allocation, developing sustainable corporate plans, and creating regulations that respect the environment.

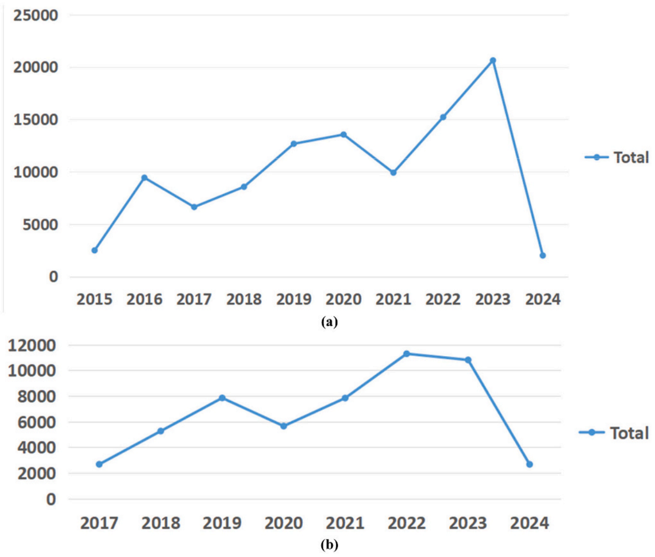


Fig. 11. Carbon prices over years; (a) BeijingETS, and (b) HubeiETS.

Table 7
The performance of the proposed carbon prediction model using MLSO and custom LSTM for BeijingETS and HubeiETS.

	BeijingETS	HubeiETS
RMSE	0.8275	0.6308
R^2	0.9993	0.9970
MAE	0.1849	0.4756
MAPE	0.4817	1.5591
MSE	0.6848	0.3979

Table 8
The benchmark datasets's description.

	Dataset name	No. of records	No. of features
D-1	House Price	507	19
D-2	California Housing Prices	20,641	10
D-3	Medical Cost Personal	1338	7
D-4	Diamonds Price	50,001	10
D-5	Tesla Stock Price	1692	7
D-6	Gold Price Prediction	1718	81

The time complexity of the proposed carbon price prediction model based on MLSO and LSTM is illustrated as follows: The time complexity for processing sequence input is linear with the sequence length (T), while the memory space scales proportionally to store the input vectors across the sequence ($T \times D$). The LSTM layer, being the most computationally intensive, has a time complexity dominated by the matrix multiplications between weight matrices and input/hidden state vectors, resulting in O($T \times N^2$), where N represents the number of hidden units. The memory complexity for the LSTM includes the cell state and hidden state (O($T \times N$)) and the weights (O(N^2)). For a dense layer, the time complexity is O($M\times N$), where M is the number of neurons in the dense layer. The memory complexity is O($M \times N + N$) to store the weights and biases. Lastly, the final regression output layer involves a linear transformation, resulting in a time complexity of O(N) with a memory complexity of O(N) to store the weights and biases. The time complexity of LSO according to [31] is O($K \times D \times I$), where K is the number of search agents, D is the number of dimensions (hyper-parameters), and I is the number of iterations. The memory space

Table 9
The performance of the proposed carbon price prediction model for six benchmark datasets collected from Kaggle repository.

		D-1	D-2	D-3	D-4	D-5	D-6
MLSO + LSTM	**RMSE**	3.9326	0.1584	0.0945	0.0906	0.00612	0.0220
	R^2	0.8155	0.5565	0.7568	0.8243	0.9996	0.9919
	MAE	2.7031	0.1170	0.0715	0.0673	0.0042	0.0173
	MAPE	0.1390	49.1202	712.4181	143.3360	2.7341	7.2779
	MSE	15.4701	0.0251	0.0089	0.0082	0.00003	0.0004
Random Forest	**RMSE**	4.1728	0.15542	0.10407	0.13763	0.0443	0.0742
	R^2	0.4657	0.4328	0.7050	0.6257	0.9429	0.6916
	MAE	3.0341	0.1154	0.0779	0.1117	0.02545	0.0436
	MAPE	16.0448	25.3584	1673.6312	466.9028	105.4297	22.8252
	MSE	21.7313	0.02592	0.0108	0.0189	0.0019	0.0055
Gradient Boosting	**RMSE**	3.7520	0.1412	0.1146	0.1164	0.0067	0.0368
	R^2	0.5387	0.5562	0.6474	0.7319	0.9760	0.9240
	MAE	2.8151	0.1043	0.0874	0.0899	0.0041	0.0207
	MAPE	14.6451	115.5684	1616.6701	455.5731	24.5184	21.3762
	MSE	16.9375	0.02074	0.0131	0.01356	0.0002	0.0013

Table 10
The proposed carbon price prediction model vs. The state-of-the-art models.

Ref.	Model	Dataset	RMSE	R^2	MSE
[21]	**PCA + DT**	Private	2.2242	0.94	4.9469
(2022)	**PCA+ ensemble Tree**	Private	1.7114	0.97	2.9288
	PCA + SVM	Private	1.9363	0.96	3.7491
	PCA + GPR	Private	1.3496	0.98	1.8214
[24]	**Radom Forest**	Private	7.4106	–	54.9169
(2022)	**LSTM**	Private	4.1257	–	17.0214
	SSA + LSTM	Private	3.9114	–	15.2904
[27]	**ANN**	Private	1.023	0.898	1.04652
(2023)	**RNN**	Private	1.052	0.892	1.10670
	LSTM	Private	0.901	0.921	0.81180
	ET + MVMD + LSTM	Private	0.376	0.986	0.14137
[44] (2024)	**WGANGP**	Public (Daily trading price of EU allowance)	3.040	–	9.2416
[28]	**ANFIS + IEFO**	Public (Carbon price historical transaction data)	2.7204	0.8635	7.40057
(2024)		Public (BeijingETS)	6.4698	0.8790	41.8583
		Public (HubeiETS)	1.0943	0.9370	1.19749
[45]	**CEEMDAN + VMD + IPSO + BiLSTM**	Public (BeijingETS)	3.6541	0.9612	13.3524
(2024)		Public (HubeiETS)	0.9719	0.9836	0.98584
The Proposed Model	**MLSO + LSTM**	Public (Carbon price historical transaction data)	0.66440	0.99912	0.44143
		Public (BeijingETS)	0.8275	0.9993	0.6848
		Public (HubeiETS)	0.6308	0.9970	0.3979

complexity for LSO is O($K \times D$). So, the overall time complexity of the proposed carbon price prediction model based on MLSO and LSTM is O($K \times D \times I \times T \times N^2$) and the space complexity is O($N + K \times D + T_{DS} + V_{DS}$), where T_{DS} is the size of the training data and V_{DS} is the size of the validation data.

7. Conclusion and future work

This paper proposes a new carbon price prediction model based on an optimized LSTM architecture and XAI. The proposed model is comprised of three main phases: data preprocessing, hyper-parameter optimization of LSTM using MLSO, and result interpretation using XAI. During the data preprocessing stage, missing values are handled and the data are normalized. In the second phase, the ideal batch size, learning rate, and solver optimization algorithm of the proposed optimized LSTM architecture are determined. Finally, the overall performance of the proposed model was evaluated using multiple metrics in conjunction with the SHAP XAI technique. The proposed model is evaluated on carbon price historical transaction data and tested on eight other benchmark datasets. The proposed model is quite promising, according to the experimental findings. It obtained 0.99 of R^2, 0.66 of RMSE, 0.44 of MSE, 0.15 of MAPE, and 0.37 of MAE for carbon price historical transaction data. Additionally, the results indicated that fine-tuning the batch size, learning rate, and solver hyper-parameters of the LSTM model can significantly enhance both its performance and the overall accuracy of carbon price predictions. The proposed MLSO-based hyper-parameters optimization performed better than MSSA, MWOA, MAVOA, MEO, MGWO, and MDE-based hyper-parameters optimization of LSTM algorithms, according to the results. Moreover, the SHAP analysis results demonstrated that low-price, high-price, and open-price features have the highest importance in driving the model's predictions and the overall price behavior in comparison to other features. The proposed model, utilizing the optimized LSTM, outperformed other well-known machine learning models as well as the latest state-of-the-art model. Future research will consider various factors influencing carbon prices, such as policies, economic development, and energy costs, to better understand their relationships, reduce model errors, and improve predictive accuracy. As a future direction, the proposed MLSO can be further applied to different deep-learning architectures. Furthermore, the proposed model can be extended to more complicated datasets.

Funding

This research received no external funding.

CRediT authorship contribution statement

Gehad Ismail Sayed: Writing – original draft, Visualization, Software, Data curation. **Eman I. Abd El-Latif:** Writing – review & editing, Writing – original draft, Validation, Data curation. **Ashraf Darwish:** Writing – review & editing, Validation, Formal analysis. **Vaclav Snasel:** Writing – review & editing, Validation, Resources. **Aboul Ella Hassanien:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors of this paper declare that there is no conflict of interest regarding its publication.

Data availability

The data presented in this paper is a public dataset obtained from https://www.investing.com/equities/keycorp-new-historical-data.

References

[1] Pradhan BB, Shrestha RM, Hoa NT, Matsuoka Y. Carbon prices and greenhouse gases abatement from agriculture, forestry and land use in Nepal. Glob Environ Chang 2017;43:26–36. https://doi.org/10.1016/j.gloenvcha.2017.01.005.

[2] Kamyab H, SaberiKamarposhti M, Hashim H, Yusuf M. Carbon dynamics in agricultural greenhouse gas emissions and removals: a comprehensive review. Carbon Lett 2024;34(1):265–89. https://doi.org/10.1007/s42823-023-00647-4.

[3] Tan J, Niu J, Niu Y, Wang H. Carbon-oriented pricing scheme in digitalized power transmission networks utilizing carbon-driven demand response program under risk investigation. J Clean Prod 2024;142210. https://doi.org/10.1016/j.jclepro.2024.142210.

[4] Hao Y, Tian C, Wu C. Modelling of carbon price in two real carbon trading markets. J Clean Prod 2020;244:118556. https://doi.org/10.1016/j.jclepro.2019.118556.

[5] The world Bank. online, available at, https://www.worldbank.org/en/news/press-release/2023/05/23/record-high-revenues-from-global-carbon-pricing-near-100-billion. Access 10/8/2023.

[6] Cheng HC, Shu MH, Huang JC. Economic strategies for efficient use of natural resources: the impact of carbon taxation and fiscal policy. Res Policy 2024;92:104927. https://doi.org/10.1016/j.resourpol.2024.104927.

[7] Green JF. Does carbon pricing reduce emissions? A review of ex-post analyses. Environ Res Lett 2021;16(4):043004. https://doi.org/10.1088/1748-9326/abdae9.

[8] Zhang WW, Zhao B, Jiang YQ, Nie YY, Sharp B, Gu Y, et al. Co-benefits of regionally-differentiated carbon pricing policies across China. Clim Pol 2024;24(1):57–70. https://doi.org/10.1080/14693062.2022.2119198.

[9] Mao S, Zeng XJ. SimVGNets: similarity-based visibility graph networks for carbon Price forecasting. Expert Syst Appl 2023;120647. https://doi.org/10.1016/j.eswa.2023.120647.

[10] Wang J, Wang Y, Li H, Yang H, Li Z. Ensemble forecasting system based on decomposition-selection-optimization for point and interval carbon price prediction. Appl Math Model 2023;113:262–86. https://doi.org/10.1016/j.apm.2022.09.004.

[11] Average carbon price projections worldwide 2022–2030, by trading system. online. Available: https://www.statista.com/statistics/1334906/average-carbon-price-projections-worldwide-by-region/.

[13] Wang J, Niu X, Zhang L, Liu Z, Huang X. A wind speed forecasting system for the construction of a smart grid with two-stage data processing based on improved ELM and deep learning strategies. Expert Syst Appl 2024;241:122487. https://doi.org/10.1016/j.eswa.2023.122487.

[14] Tarek Z, Shams MY, Elshewey AM, El-kenawy ESM, Ibrahim A, Abdelhamid AA, et al. Wind power prediction based on machine learning and deep learning models. Computers, Materials & Continua 2023;75(1). https://doi.org/10.32604/cmc.2023.032533.

[15] Li K, Huang W, Hu G, Li J. Ultra-short term power load forecasting based on CEEMDAN-SE and LSTM neural network. Energ Buildings 2023;279:112666. https://doi.org/10.1016/j.enbuild.2022.112666.

[16] Paolella MS, Taschini L. An econometric analysis of emission allowance prices. J Bank Financ 2008;32(10):2022–32. https://doi.org/10.1016/j.jbankfin.2007.09.024.

[17] Zhu B, Chevallier J, Zhu B, Chevallier J. Carbon price forecasting with a hybrid Arima and least squares support vector machines methodology. Pricing and Forecasting Carbon Markets: Models and Empirical Analyses 2017;87-107. https://doi.org/10.1007/978-3-319-57618-3_6.

[18] Nadirgil O. Carbon price prediction using multiple hybrid machine learning models optimized by genetic algorithm. J Environ Manag 2023;342:118061. https://doi.org/10.1016/j.jenvman.2023.118061.

[19] Dhamija AK, Yadav SS, Jain PK. Forecasting volatility of carbon under EU ETS: a multi-phase study. Environ Econ Policy Stud 2017;19:299–335. https://doi.org/10.1007/s10018-016-0155-4.

[20] Zhu B, Zhang J, Wan C, Chevallier J, Wang P. An evolutionary cost-sensitive support vector machine for carbon price trend forecasting. J Forecast 2023;42(4):741–55. https://doi.org/10.1002/for.2916.

[21] Xiong S, Wang C, Fang Z, Ma D. Multi-step-ahead carbon price forecasting based on variational mode decomposition and fast multi-output relevance vector regression optimized by the multi-objective whale optimization algorithm. Energies 2019;12(1):147. https://doi.org/10.3390/en12010147.

[22] Rudnik K, Hnydiuk-Stefan A, Kucińska-Landwójtowicz A, Mach Ł. Forecasting day-ahead carbon price by modelling its determinants using the PCA-based approach. Energies 2022;15(21):8057. https://doi.org/10.3390/en15218057.

[23] Fan X, Li S, Tian L. Chaotic characteristic identification for carbon price and an multi-layer perceptron network prediction model. Expert Syst Appl 2015;42(8):3945–52. https://doi.org/10.1016/j.eswa.2014.12.047.

[24] Zhu BZ, Wei YM. Carbon price prediction based on integration of GMDH, particle swarm optimization and least squares support vector machines. Systems Engineering-Theory & Practice 2011;31(12):2264–71. https://doi.org/10.12011/1000-6788(2011)12-2264.

[25] Wang F, Jiang J, Shu J. Carbon trading price forecasting: based on improved deep learning method. Procedia Computer Science 2022;214:845–50. https://doi.org/10.1016/j.procs.2022.11.250.

[26] Zhu B, Han D, Wang P, Wu Z, Zhang T, Wei YM. Forecasting carbon price using empirical mode decomposition and evolutionary least squares support vector regression. Appl Energy 2017;191:521–30. https://doi.org/10.1016/j.apenergy.2017.01.076.

[27] Lin B, Zhang C. Forecasting carbon price in the European carbon market: the role of structural changes. Process Saf Environ Prot 2022;166:341–54. https://doi.org/10.1016/j.psep.2022.08.011.

[28] Zhang K, Yang X, Wang T, Thé J, Tan Z, Yu H. Multi-step carbon price forecasting using a hybrid model based on multivariate decomposition strategy and deep learning algorithms. J Clean Prod 2023;405:136959. https://doi.org/10.1016/j.jclepro.2023.136959.

[29] Abd Elfattah M, Ewees AA, Sayed GI, Darwish A, Hassanien AE. Carbon price time series forecasting utilizing an optimized ANFIS model. Evol Intel 2024;1-21. https://doi.org/10.1007/s12065-024-00955-2.

[30] Yang C, Zhang H, Weng F. Effects of COVID-19 vaccination programs on EU carbon price forecasts: evidence from explainable machine learning. Int Rev Financ Anal 2024;91:102953. https://doi.org/10.1016/j.irfa.2023.102953.

[31] Wang J, He M, Jiang W. Causal carbon price interval prediction using lower upper bound estimation combined with asymmetric multi-objective evolutionary algorithm and long short-term memory. Expert Syst Appl 2024;236:121286. https://doi.org/10.1016/j.eswa.2023.121286.

[32] Thulasi T, Sivamohan K. LSO-CSL: light spectrum optimizer-based convolutional stacked long short term memory for attack detection in IoT-based healthcare applications. Expert Syst Appl 2023;232:120772. https://doi.org/10.1016/j.eswa.2023.120772.

[33] Abdel-Basset M, Mohamed R, Sallam KM, Chakrabortty RK. Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. Mathematics 2022;10(19):3466. https://doi.org/10.3390/math10193466.

[34] Abdel-Basset M, Mohamed R, Abouhawwash M, Alshamrani AM, Mohamed AW, Sallam K. Binary light spectrum optimizer for knapsack problems: an improved model. Alex Eng J 2023;67:609–32. https://doi.org/10.1016/j.aej.2022.12.025.

[35] Lindemann B, Müller T, Vietz H, Jazdi N, Weyrich M. A survey on long short-term memory networks for time series prediction. Procedia CIRP 2021;99:650–5. https://doi.org/10.1016/j.procir.2021.03.088.

[36] Alsubai S, Dutta AK, Alkhayyat AH, Jaber MM, Abbas AH, Kumar A. Hybrid deep learning with improved Salp swarm optimization based multi-class grape disease classification model. Comput Electr Eng 2023;108:108733. https://doi.org/10.1016/j.compeleceng.2023.108733.

[37] Hassija V, Chamola V, Mahapatra A, Singal A, Goel D, Huang K, et al. Interpreting black-box models: a review on explainable artificial intelligence. Cogn Comput 2024;16(1):45–74. https://doi.org/10.1007/s12559-023-10179-8.

[38] Chollet F Keras. Deep learning for humans. online, available at, https://github.com/fchollet/keras-resources. Access 22/8/2023.

[39] Brodzicki A, Piekarski M, Jaworek-Korjakowska J. The whale optimization algorithm approach for deep neural networks. Sensors 2021;21(23):8003. https://doi.org/10.3390/s21238003.

[40] Uppamma P, Bhattacharya S. Diabetic retinopathy detection: a blockchain and African vulture optimization algorithm-based deep learning framework. Electronics 2023;12(3):742. https://doi.org/10.3390/electronics12030742.

[41] Le TL, Huynh TT, Hong SK. A modified grey wolf optimizer for optimum parameters of multilayer type-2 asymmetric fuzzy controller. IEEE Access 2020;8:121611–29. https://doi.org/10.1109/ACCESS.2020.3006469.

[42] Baioletti M, Di Bari G, Milani A, Poggioni V. Differential evolution for neural networks optimization. Mathematics 2020;8(1):69. https://doi.org/10.3390/math8010069.

[43] Beijing's carbon emission rights public trading situation. online, available at, https://www.bjets.com.cn/.

[44] Hubei carbon emission exchange. online, available at, https://www.hbets.cn/list_51.html. Access, 14/4/2024.

[45] Chen Y. Carbon Price prediction for the European carbon market using generative adversarial networks. Mod Econ 2024;15(03):219–32. https://doi.org/10.4236/me.2024.153011.