International Journal of Energy Economics and Policy

ISSN: 2146-4553

available at http: www.econjournals.com

International Journal of Energy Economics and Policy, 2025, 15(6), 531-542.

Forecasting Carbon Emission Allowance Prices With Deep Learning Models

Pinar Unal[1]*, Seyma Nur Guner[2]

[1]Department of International Finance, Faculty of Economics, Administrative and Social Sciences, Bahcesehir University, Istanbul, Turkiye, [2]Department of Economics, Faculty of Economics, Administrative and Social Sciences, Karabuk University, Karabuk, Turkiye. *Email: pinar.unal@bau.edu.tr

Received: 30 May 2025 **Accepted:** 25 September 2025 **DOI:** https://doi.org/10.32479/ijeep.21072

ABSTRACT

As the effects of the climate crisis deepen, the measures taken against it are increasing. These precautionary mechanisms significantly affect companies. One of these mechanisms is carbon trading systems, which limit companies' carbon emissions. These systems create additional costs for companies, and predicting these potential costs is crucial today. For this reason, the importance of correctly estimating carbon permit prices has increased. To estimate carbon permit prices, the correct variables must first be determined. Many factors affect the carbon markets due to their complex nature. Using the correct estimation method is another important issue. Traditional econometric models cannot clearly explain the complex structure of the carbon market, resulting in low accuracy. In this direction, artificial intelligence-based estimation models have been developed to estimate carbon permit prices. This study estimated the prices of European Union Emissions Allowances (EUA) using energy and power market prices and the economic outlook. The estimation model utilized deep learning algorithms, which are feedforward neural networks and recurrent neural networks (Elman). As a result of the study, both models were successful, and strong accuracy was obtained. While the FFNN model provided more stable and low-variance results, the Elman model provided advantages in short-term predictions.

Keywords: Deep Learning, Artificial Neural Networks, Carbon Trading Systems, Carbon Price
JEL Classifications: Q1, Q4, Q5, C6

1. INTRODUCTION

Currently, environmental challenges cause irreversible harm to ecosystems and substantially impact economic and social equilibrium. Given that human and economic activity aggravate environmental issues, it is essential to establish preventive methods that address this base. In this context, market-oriented strategies are progressively employed to mitigate greenhouse gas emissions. Carbon trading systems are market-based regulatory mechanisms that incentivize firms to adhere to specified carbon emission limitations and assist them in minimizing emission reduction expenses.

The largest carbon trade system in the world is The European Union Emission Trading System (EU ETS). The main purpose of the EU ETS is to reduce carbon emissions and promote clean energy. This ability creates a financial market by allowing companies to assign and distribute certain carbon emission amounts (Huang and He, 2020). European Union Emission Allowances (EUA), which allow this system to be represented by financial instruments that allow carbon emissions, are bought and sold. Companies must comply with the limits assigned by EUA's (Koch et al., 2014). If a company exceeds the emission amount given to it, the EUA must be purchased to continue its activities. Similarly, if it has not filled the permit limit given to it, it has the right to sell the remaining limit. EUA prices are of great importance in terms of both commercial activities and carbon control mechanisms. Therefore, carbon prices are affected by many factors such as the energy market, economic outlook, and legal improvements (Unal, 2025).

The EU ETS is intended to reduce CO_2 emissions, expedite industrial restructuring and increase the cost-effectiveness of emission reductions (Huang and He, 2020). The effects of carbon trading schemes on the economy and the environment are quite diverse. High carbon emitting industries are moving to more environmentally friendly manufacturing techniques when EUA costs vary (Creti et al., 2011). But for industrial companies and investors, the continual changes in EUA prices cause major uncertainty that highlights the need of risk management techniques. EUA price changes are directly related to energy market prices including brent oil, natural gas, coal, and electricity. The present economic condition is also one of the key elements influencing the price dynamics in carbon markets.

Policymakers, industrial companies, and investors should forecast future changes in carbon costs. Accurate forecasts enable businesses to create their emission trading plans, fund low-carbon technology, and control their expenses. Uncertain carbon pricing confuses long-term investment choices and hinder market players' ability to control risk (Benz and Truck, 2008). Long-term investment plans must be created using strong forecasting models, which will help to lower carbon market volatility.

Estimating carbon prices helps to create the carbon pricing system and offers important direction for production, operation, and investment choices (Narassimhan et al., 2018). Many ways so exist to forecast carbon pricing. Carbon price forecasting is done using a wide methodological approach including machine learning and econometric-statistical analysis. Though conventional econometric techniques are frequently employed to detect important patterns in the carbon market, but they have certain drawbacks since they cannot completely simulate the complexity of the market. Recent standout artificial intelligence (AI), machine learning (ML) and deep learning (DL) models have shown an ability to identify nonlinear correlations and extract significant insights from extensive data sets.

Econometric and statistical time series studies form the basis of the most often used conventional method for predicting carbon pricing. Among them are vector auto regression (VAR), autoregressive integrated moving average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH). Though they forecast data by looking at the volatility in prior data, these conventional methods lack long-term projections as they do not completely represent the volatility of carbon pricing and market dynamics.

In the context of carbon price forecasting, artificial intelligence-based models have acquired increased interest recently. Because they can describe nonlinear relationships in complex data, machine learning and deep learning models outperform conventional techniques. Artificial neural networks, decision trees, random forests, XGBoost, and lightGBM are some of the methods used in this field. These models can replicate nonlinear relationships and market dynamics in large data sets. Therefore, compared to conventional techniques, artificial intelligence-based models are more precise, adaptable, and capable of deriving meaning from massive data sets.

Economic circumstances and energy markets are non-linear and intricately linked to carbon pricing. Artificial intelligence algorithms that better manage complicated structures and vast data sets therefore catch these interactions more consistently.

This paper suggests using AI based deep learning models to forecast carbon prices for the European Union ETS. Among the variables employed for forecasting are the energy market, economic situation, and quantity of emission. Thus, the model includes all internal and external factors influencing the carbon market. EUA prices were estimated using Artificial Neural Networks (ANN) as one of the deep learning models. Two distinct kinds of ANNs—feedforward and recurrent—were used to compare results.

2. LITERATURE REVIEW

Accurate prediction of carbon prices plays a critical role in the development of sustainable economic policies, ensuring the effective functioning of carbon markets, and the strategic decision-making processes of companies. Within the framework of carbon trading systems, companies are trying to balance their emissions by buying and selling carbon certificates and managing their financial risks based on price fluctuations. As a result, research on carbon price forecasting is crucial for businesses to decide on their market strategies, for regulatory agencies to create more efficient policy tools, and for investors to make the best investment choices. Numerous forecasting techniques, based on both conventional models and AI-based methodologies, have been developed in the literature in this regard.

García-Martos et al. and Benz and Truck are studies that use traditional forecasting models. García-Martos et al. used the ARIMA, VARIMA, and Multivariate GARCH models to forecast electricity prices, CO_2 emission rights, and fossil fuels (oil, natural gas, and coal) in combination (García-Martos et al., 2012). For CO_2 and electricity prices, the univariate models ARIMA and VARIMA performed better in forecasting, whereas the multivariate GARCH model performed better for fossil fuel prices. Benz and Truck used the GARCH and Markov-Switching Model (MSM) to look at the short-term changes in CO_2 emission allowance prices in the EU ETS (Benz and Truck, 2008). Markov -switching models showed how the market moved between periods of high and low volatility and price changes. GARCH models, on the other hand, showed how carbon prices changed and were very good at predicting short-term volatility. But they could only get short-term modeling results because they didn't have enough data.

To compare traditional models with AI-based models The Adaptive Multiscale Ensemble Learning Paradigm model was developed by Zhu et al. to estimate the prices of EUA, oil, natural gas, gasoline, and electricity (Zhu et al., 2016). In comparison to more traditional methods of forecasting, the proposed model has been discovered to have far higher levels of accuracy.

Qin et al. also developed a hybrid model to compare with traditional forecasting models to predict carbon futures prices on the European Climate Exchange (ECX) and China carbon prices

(Qin et al., 2022). They concluded that the hybrid model has a stronger accuracy rate than traditional models.

Kumar et al. compared ARIMA-based classical time series models with machine learning algorithms for forecasting the future prices for carbon emissions (Kumar et al., 2024). As a result, the research has shown that performance of machine learning models is over ARIMA model. Regarding the machine learning models, the decision tree model has better performance; and the random forest model has worse performance.

Other hybrid model introduced by Zhang and Wu for predicting European Climate Exchange (ECX) futures carbon prices, utilizing Least Squares Support Vector Machine (LSSVM), ARMA-type time series models, and data decomposition via Empirical Mode Decomposition (EMD) (Zhang and Wu, 2021). It was determined that traditional models exhibit inadequate accuracy compared to the hybrid model.

Sanin et al. analyzes the short-term price dynamics and volatility of carbon permit markets in the EU ETS (Sanin et al., 2015). In this context, ARMAX-GARCH and the Bernoulli mixture model (BMN) were employed to compute volatility. The findings indicate that the conventional GARCH model inadequately detect volatility. The BMN model clarified the volatility more well.

Ji et al. proposed a fusion model that is composed of ARIMA-CNN-LSTM models for predicting carbon futures price (Ji et al., 2019). The findings indicate the ARIMA-CNN-LSTM model was able to generate more accurate predictions by accounting for both linear and nonlinear dependencies in the carbon market.

In their analysis, Koop and Tole applied the Dynamic Model Averaging technique to forecast volatility in the carbon markets of the EU ETS (Koop and Tole, 2012). The appropriate performance of the model was verified by the agreement between the simulated market characteristics and the evolution of the market itself observed during the studied period.

Huang, He proposed a combinatorial optimization forecasting model to enhance the price prediction accuracy in Chinese carbon trading market (Huang and He, 2020). In the research, MOEMD-CKA- ELM (Mean value Optimization Empirical Mode decomposition - Cooperative Kidney Algorithm - Extreme Learning Machine) model was adopted, which realized the high accuracy to some extent.

Li et al. created a Multivariate LSTM model for Guangdong (GDEA) and Hubei (HBEA) carbon price prediction and compared with other methods (Li et al., 2022). In their models, they incorporated historical prices for coal, natural gas and oil as well as historical prices for carbon. The LSTM model attained the lowest error rates among the other forecasting methods in their study.

Zhou et al. developed a carbon pricing forecasting model based on CEEMDAN and a long short-term memory model (Zhou et al., 2022). Their models were based on daily closing prices of the Guangzhou trading system. They demonstrated that this model produced better accuracy than single LSTM and other benchmark models. By simulating the non-linear and non-stationary law, the error rate was reduced by using the combination of CEEMDAN and LSTM.

Byun and Cho compare three methods to forecast the volatility of carbon futures: GARCH-type models, implied volatility (IV) on option prices, and the k-nearest neighbor (Byun and Cho, 2013). To forecast the volatility of the ECX, the closing price daily returns of the ECX and the closing price daily returns of brent oil, natural gas, coal and electricity futures markets were used. It is concluded that GARCH-type models outperform the IV on option prices and the k-NN model.

Wang et al. proposed a hybrid model with the use of CEEMDAN (complete ensemble empirical mode decomposition with adaptive Noise), SE (sample entropy), LSTM (long short-term memory), and RF (random forest) model (Wang et al., 2020). By the model proposed, they forecast carbon prices based on the historical data of the China's Tianjin and Guangdong carbon markets. They have found that their proposed model has better prediction abilities than any single model. However, they did research on historical data, and they proposed that the model can be enriched by macroeconomic and financial variables.

Many models have been used in the literature to estimate carbon permit prices. The common result in all of them is that AI-based models provide better estimates than traditional models. The points that need to be improved are including more factors affecting the carbon market to the models instead of using only historical data on carbon prices. Moreover, to capture long-term effects and patterns, it is necessary to work with long interval periods.

3. METHODOLOGY AND DATA

Artificial neural networks (ANN), one of the deep learning methods, have been used to predict carbon prices in this study. The interest in artificial neural networks stems from the belief that they can help us better understand the brain, human cognition and perception, and the hope that these brain-like systems can achieve greater success than ever before in tasks such as pattern classification, decision making, prediction and adaptive control (Fine, 1999).

Inspired by biological neural networks, artificial neural networks are highly parallel computer systems formed by simple units with numerous interconnections. One category of network regards the nodes as "artificial neurons." An artificial neuron is a computational model derived from biological neurons. Given that the primary role of artificial neural networks is information processing, they are predominantly employed in related areas. A diverse array of artificial neural networks is employed to simulate actual neural networks and to investigate behavior and control in both living organisms and machines. Conversely, artificial neural networks are employed in engineering applications like pattern recognition, prediction, and data compression. They fundamentally include inputs (such as synapses) adjusted by weights. The weights attributed to each direction denote the flow of information. The weights are subsequently computed by a mathematical algorithm that ascertains the neuron's activity. Another execution computes the output of the artificial neuron, occasionally relying on a specific threshold. The neurons in this network only aggregate their inputs. The output of the input neurons, which possess a singular input, will be the received input multiplied by a weight.

It is possible to separate the architecture of the neural network into two distinct categories, depending on the connection configuration. These are feed forward networks, where the graphs do not contain cycles. The other is recurrent neural networks, where cycles occur due to feedback connections. Feed forward and Elman neural networks were chosen as estimation methods to make comparisons with two different types, both feed forward and recurrent.

3.1. Feedforward Neural Networks

Feedforward neural networks (FFNN) represent the fundamental category of artificial neural networks. FFNN has a multi-layered structure and transmits information unidirectionally from the input layer to the output layer. Since there is no feedback mechanism, they have a loopless structure. They are widely used in pattern classification because they do not require any information about the probability distribution and prior probabilities of different classes (Wang et al., 2015). The learning process consists of the stages of forward pass of the input data and backpropagation of the error.

The foundations of Feedforward Neural Networks are based on the study titled "A Logical Calculus of the Ideas Immanent in Nervous Activity" (McCulloch and Pitts, 1943). In the study, biological nervous systems were analyzed mathematically, and a computational model was developed inspired by this analysis. This developed model forms the basis of feed forward neural networks.

Subsequently, Frank Rosenblatt presented a more pragmatic iteration of feedforward neural networks through the creation of the Perceptron model in 1958 (Rosenblatt, 1958). The genuine popularization of multilayer feedforward networks and their efficient training transpired in 1986 when Rumelhart, Hinton, and Williams unveiled the backpropagation method (Rumelhart et al., 1986).

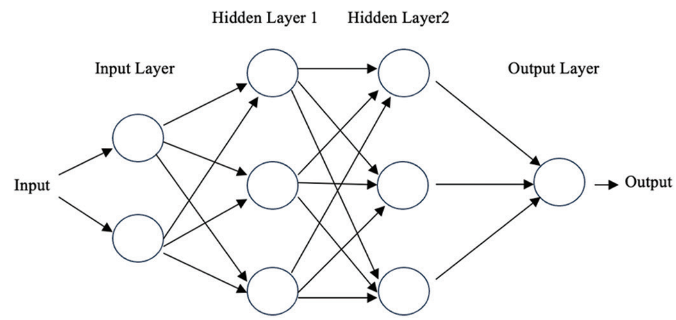
FFNN has an architecture where neurons in each layer receive information from neurons in the previous layer, but there is no feedback or looping.

FFNN consists of three basic layers as seen in Figure 1. This architecture is the input layer, which represents the data used in the model, the hidden layer, where the weighted sum of the inputs is calculated and transformed through activation functions, and the output layer, where the final predictions are located. The FFNN mathematical function is generally as following Equation 1:

$$X \rightarrow (W_1, b_1) \rightarrow f_1(Z_1) \rightarrow (W_2, b_2) \rightarrow f_2(Z_2) \rightarrow \cdots \rightarrow$$
$$(W_L, b_L) \rightarrow f_L(Z_L) \rightarrow \hat{Y} \tag{1}$$

Here; $X \rightarrow$ input matrix, $W_L, b_L \rightarrow$ weight matrix and bias vector in L^{th} layer, $Z_L \rightarrow$ weighted total, $F_L \rightarrow$ activation function, $\hat{Y} \rightarrow$ model's prediction.

Figure 1: Feed forward neural networks layers



In the model, the inputs are passed through the network layers to produce output. First, each neuron adds weights to the inputs it receives and adds bias. With this process, the weighted sum is obtained as in following Equation 2:

$$Z_L = W_L A_{L-1} + b_L \tag{2}$$

Here; $Z_L \rightarrow$ weighted sum at layer L^{th} (neuron input), $W_L \rightarrow$ weight matrix in layer L^{th} ($m_L \times m_{L-1}$), $A_L \rightarrow$ activation output from the previous layer, $b_L \rightarrow$ bias vector ($m_L \times 1$)

In this process, each neuron applies a nonlinear activation function. This process continues from the input layer to the output layer.

$$A_L \rightarrow f(Z_L) \tag{3}$$

A nonlinear activation function represented as f(.) is employed to derive the activation output (as seen in Equation 3). Sigmoid, ReLU (Rectified Linear Unit), and Tanh (Hyperbolic Tangent) are activation functions that are often applied.

FFNN learning relies on reducing the discrepancy between the model's prediction and the actual value. Mean Squared Error (MSE) is typically employed as an error calculation function for regression problems, while Cross-Entropy Loss functions are utilized for classification problems.

Upon computing the errors, the model endeavors to minimize them through a backpropagation technique. The gradient descent method is employed to adjust the weights in this process.

Firstly, gradient for weight and bias is calculated as seen in Equation 4 and 5.

$$\frac{\partial L}{\partial W_L} = \delta_L A_{L-1}^T \tag{4}$$

$$\frac{\partial L}{\partial b_L} = \delta_L \tag{5}$$

The general method for calculating the error signal (δ_L) in the gradient calculation is as in Equation 6. It is adapted according to the activation function used.

$$\delta_L = \frac{\partial L}{\partial A_L}.f'(Z_L) \tag{6}$$

According to the calculated error signal, the weights and bias are updated using the learning rate (η) as seen in Equation 7 and 8. Thus, FFNN minimizes the error function. The feedforward process is repeated with the updated weights and the model outputs are calculated. When this process reaches the specified epoch (iteration) number or the error falls below a specified threshold value, the model training is completed and is ready for use.

$$W_L = W_L - \eta.\frac{\partial L}{\partial W_L} \tag{7}$$

$$b_L = b_L - \eta.\frac{\partial L}{\partial b_L} \tag{8}$$

3.2. The Recurrent Neural Network (Elman)

The Elman Neural Network introduced by Jeffrey L. Elman in 1990, is a recurrent artificial neural network model (Elman, 1990). In contrast to conventional feedforward networks, it may make judgments by retaining information from prior time steps and incorporating it into the learning process through a feedback mechanism in its hidden layer. This characteristic enables Elman neural networks to expert in time series prediction and modeling.

Elman neural networks have 4 basic layers in seen in Figure 2. These are the input layer, which contains the input vectors that feed the network, the hidden layer, which consists of neurons processed with activation functions, the context layer, which receives and stores the output of the previous hidden layer and provides feedback, and the output layer, where the output of the model is produced (Ren et al., 2018).

For the Elman neural network design, the input vector X_t is first created. Then, the $Z_{j,t-1}$ values in the context layer are pulled from the context layer. Finally, to calculate the hidden layer activation $Z_{j,t,}$ the values taken from the input layer and the context layer are processed with the following Equation 9: (Claveria et al., 2014).

$$Z_{j,t} = g\left(\sum_{i=1}^{p} \varphi_{ij} X_{t-i} + \varphi_{0j} + \delta_{ij} Z_{j,t-1}\right) \tag{9}$$

Here;

$\{x_{t-i} = (1, x_{t-1}, x_{t-2}, \ldots, x_{t-p}), i = 1,\ldots,p\}$

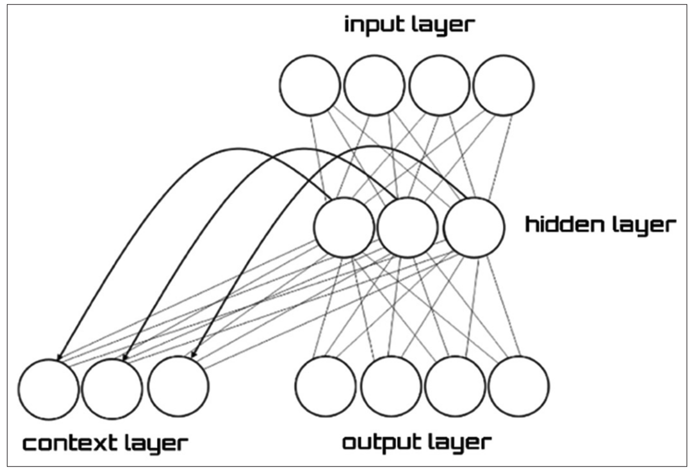
$\{\varphi_{ij}, i = 1,\ldots,p, j = 1,\ldots,q\}$

$\{\delta_{ij}\ i = 1,\ldots,p, j = 1,\ldots,q\}$

The Elman neural network algorithm receives data from the input layer, processes it in the hidden layer, and stores it in the context layer for next utilization. Thereby, both current and past effects are processed together to create the Elman neural network prediction architecture.

First, an input vector is created using past inputs in the input layer. This inputs vector (X_{t-i}) allows the network architecture to learn by seeing previous data. Then, the data coming from the input layer is multiplied by the weight matrix (φ_{ij}) and transmitted to the hidden layer. The bias term (φ_{0j}) is added for each neuron to the weighted input data coming to the hidden layer. The use of the bias term is important. Because it makes the learning process more flexible and facilitates the modeling of nonlinear relationships. The neurons in the hidden layer include the activation coming from the context layer in the processing process in addition to the data coming from the input layer. The hidden layer activation of the previous time step in the context layer ($Z_{j,-1}$) is included in the new hidden layer activation by multiplying it with the feedback weight coefficient (δ_{ij}). This entire process is performed for the activation of each neuron in the hidden layer. Then, the total activation is processed with the activation function g(·). With the activation function, the inputs are transformed into a nonlinear form, which allows the network to learn complex relationships. Sigmoid, ReLu (Rectified Linear Unit), Tanh (Hyperbolic Tangent) functions are commonly used as the activation function in the same way as FFNN.

Figure 2: Elman neural networks layers



The layer of the network that creates predictions is the output layer. The activations created in the hidden layer are transmitted to the output layer (Y_t). The output layer multiplies the data coming from the hidden layer with the output weights (β_j) and produces the output by passing it through the output activation function as seen in Equation 10.

$$Y_t = \beta_0 + \left(\sum_{j=1}^{q} \beta_j Z_{j,t}\right) \tag{10}$$

Here;

$\{\beta_j, j = 1,\ldots,q\}$

The learning of Elman networks is also based on minimizing the error between the estimated value and the actual value, in the same way as FFNN. Mean Squared Error is usually used to calculate the error function. This error function (E) measures how much error the model makes during the learning process.

After measuring the errors, Elman networks are trained with the Elman Backpropagation through time (BPTT) algorithm to better train the model and minimize the errors. With this algorithm, the error gradients of the context layer, hidden layer and output layer are calculated, and the weights are updated with these error gradients using the Gradient Descent method. The speed of weight updates is determined by including the learning rate η in the update process as seen in Equation 11, 12 and 13.

$$\varphi_{ij} \leftarrow \varphi_{ij} - \eta \sum_{t-1}^{T} \frac{\partial E}{\partial \varphi ij} \text{ : Input-Hidden Layer Weights Update} \tag{11}$$

$$\beta_{ij} \leftarrow \beta_{ij} - \eta \sum_{t-1}^{T} \frac{\partial E}{\partial \beta ij} \text{ : Hidden Layer-Output Layer Weights Update} \tag{12}$$

$$\delta_{ij} \leftarrow \delta_{ij} - \eta \sum_{t-1}^{T} \frac{\partial E}{\partial \delta ij} \text{ : Context Layer (Memory) Weights Update} \tag{13}$$

In this way, the model tries to make more accurate predictions by minimizing the error function in each iteration. The feed forward process with updated weights is repeated to calculate the model's outputs, and this process continues until the specified epoch (iteration) number is reached or until the error falls below a specified threshold value. In the final stage, when the training is complete, the model is ready to make predictions with the patterns it has learned and the relationships over time.

3.3. Data

In most studies on estimating carbon prices, only historical data on carbon prices are considered. However, the carbon market is dynamic and is affected by many factors. To fill this gap in the literature, in addition to historical prices, the energy market, economic situation and carbon emission amount factors are also included in the model to estimate carbon prices. It has also been used over a long period of time to capture long-term effects and patterns.

The aim of the study is to forecast the European Union Emission Allowances prices. The phases of the EU ETS System were taken into consideration for the period of the data used in the study. The system started in 2005 and the first phase between 2005 and 2007 was the pilot period. Since carbon prices were determined according to estimates in the pilot period and there was no trading market condition, this phase was not included in the study. The beginning of phase 2, when the EU ETS market system started to work actively, was taken into consideration in 2008. The data period used in the study is 2008-2023 and a daily data frequency was applied (4131 observations). Thus, the entire period during which the system worked actively was included in the study to contribute to the literature. Other variables used to estimate carbon prices(EUA) are energy market variables such as coal, brent oil, natural gas, electricity, heating oil, MSCI and STOXX indices reflecting the economic situation, and CO_2 emission amount as an environmental condition. Table 1 shows the descriptive variables of the variables. All data were obtained from the Bloomberg Database.

3.4. Research Design

The design of the study is shown in Figure 3. After the time interval was determined by considering the important issues specific to the subject of the study and the data was collected, the data was first normalized. Then, a preliminary analysis was made to see the relationship between the data used in the study and the carbon market. After testing the suitability of the data, a deep learning prediction algorithm was designed. Finally, the performance tests of the established models were performed, comparisons were made, and the best model was evaluated.

4. ANALYSIS

4.1. Processing Data

The variables to be used in the model are at different scales. If included in the model in this way, the model may produce incorrect results. For this reason, the data was first analyzed with the min-max method as shown in Equation 14 below.

$$x^{'} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{14}$$

4.2. Preliminary Analysis

First, it was tested whether the variables selected to determine carbon prices really influence carbon prices. In this context, the Granger causality test was first applied with the E-Views 12 program. The Granger causality test is used to determine whether a time series helps predict another time series. The Granger causality test tests whether the past values of a variable can explain the current or future values of another variable (Granger, 1969).

The results of the Granger causality test are in Table 2 above. First, the appropriate lag for the determined input variables was calculated as "2" and the Granger causality test was applied. The causality hypothesis test result with the carbon price for brent oil, natural gas, heating oil, electricity, MSCI index, STOXX index and CO_2 emission amount was calculated as <0.05. In other words, carbon prices are affected by these variables, and the predictive power of these variables is high in estimating carbon prices. The causality test result of coal prices was >0.05. According to this result, it is seen that there is no causality relationship between coal prices and carbon prices.

A correlation relationship test was also examined to test the relationship between input variables and carbon price with the Python 3.7 programming language.

The Table 3 shows the correlation relationship between carbon price and input variables. The results show that there is a strong relationship between carbon prices of electricity, coal, MSCI and STOXX variables, a moderate relationship between heating oil and carbon price, and a weak relationship between natural gas, brent oil and CO_2 emission amount and carbon prices.

Figure 4 shows the heat map of the relationship between input variables and carbon price. Dark shades of red represent strong positive correlation, while shades of blue represent negative correlation.

It has been examined in detail whether the input variables selected to estimate carbon prices really influence carbon prices.

Figure 3: Design of the study

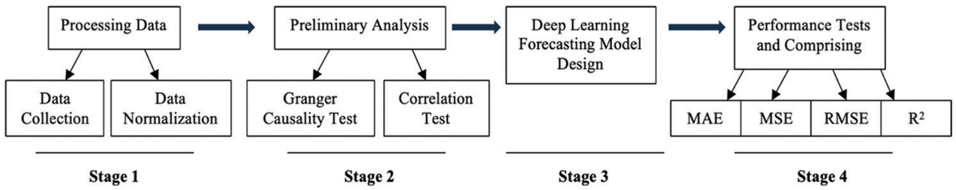


Table 1: Descriptive variables of the variables

Variable	Mean	Sum	Max	Min	SD	Variance	Skewness	Kurtosis
Carbonprice	22.577	93265.99	100.340	2.75	25.422	646.27	1.708	4.544
Coal	100.591	415542.47	439.000	38.45	60.106	3612.734	2.757	11.603
Brent	78.512	324332.1	146.080	19.33	25.532	651.867	0.146	1.971
Naturalgas	3.834	15838.831	13.577	1.482	1.876	3.521	2.19	8.774
Electricity	382.115	1578516.6	595.720	208.32	86.791	7532.695	0.426	2.098
Heatingoil	229.614	948535.59	513.540	61.04	74.013	5477.929	0.311	2.408
Co2amount	6304059.5	2.604e+10	9995485.66	7981.819	2987101.1	8.923e+12	−0.895	2.474
Msci	115.856	478601.04	163.980	55.2	23.587	556.328	−0.149	2.256
Stoxx	341.491	1410701.3	494.350	157.97	73.952	5468.85	−0.119	2.202

SD: Standard deviation

Table 2: The results of the granger causality test

Null hypothesis	Obs	F-statistic	Probability
Brent does not Granger Cause carbonprice	4129	11.2432	1.E-05
Carbonprice does not Granger Cause brent	4129	0.35397	0.7019
Naturalgas does not Granger Cause carbonprice	4129	3.08753	0.0407
Carbonprice does not Granger Cause naturalgas	4129	0.14499	0.8650
Heatingoil does not Granger Cause carbonprice	4129	10.005	5.E-05
Carbonprice does not Granger Cause heatingoil	4129	1.76570	0.1712
Coal does not Granger Cause carbonprice	4129	2.83319	0.0589
Carbonprice does not Granger Cause coal	4129	1.14328	0.3189
Electricity does not Granger Cause carbonprice	4129	21.9631	3.E-10
Carbonprice does not Granger Cause electricity	4129	1.30413	0.2715
Msci does not Granger Cause carbonprice	4129	47.0112	6.E-21
Carbonprice does not Granger Cause msci	4129	1.32509	0.2659
Stoxx does not Granger Cause carbonprice	4129	48.5034	2.E-21
Carbonprice does not Granger Cause stoxx	4129	1.18308	0.3064
Co2amount does not Granger Cause carbonprice	4129	4.03265	0.0178
Carbonprice does not Granger Cause co2amount	4129	1.73331	0.1765

When all tests are evaluated as a whole, coal, brent oil, natural gas, electricity, heating oil MSCI and STOXX indices, CO_2 emission amount variables are among the factors affecting the carbon market. For this reason, they will play an important role in estimating carbon prices.

Table 3: Correlation relationship results

Variable	Correlation with carbon price
Carbon price	1.000000
Coal	0.640852
Brent	0.062447
Naturalgas	0.094593
Electricity	0.681457
Heating oil	0.302246
Msci	0.648717
Stoxx	0.653874
CO_2	0.129941

4.3. Forecast Model Building

In the study, coal, brent oil, natural gas, electricity, heating oil prices, MSCI and STOXX indices and CO_2 emission data between 2008-2023 were used to estimate the carbon price. In the first stage of the model design, the data was divided into training, validation and test. Although there is no general method to be used when separating the data, the separation was made using the separation mostly applied in the literature as 70% training, 15% validation and 15% test data. To account for the changing units, ranges and sizes in the model, normalized data between 0-1 was implemented.

4.3.1. Feedforward neural networks design

The FFNN model was developed using the MATLAB Neural Network Toolbox and trained with the Levenberg-Marquardt (TRAINLM) algorithm. As seen in Figure 5, there are 8 input neurons representing the variables in the input layer of the model. The hidden layer consists of 10 neurons, and the output layer consists of 1 neuron.

TANSIG (Hyperbolic Tangent Sigmoid) was used to obtain the activation output in the hidden layer, as seen in Equation 15.

$$f(z) = \frac{2}{1+e^{-2z}} - 1 \tag{15}$$

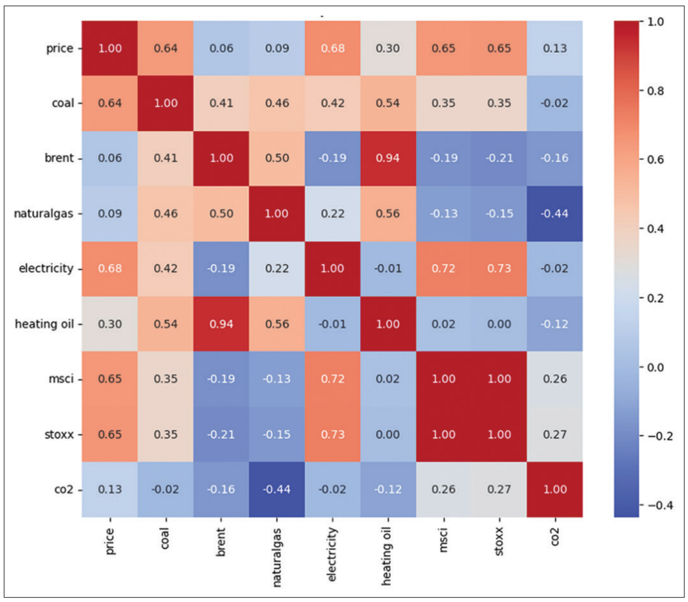
During the training process, the algorithm tries to minimize the error between the estimated value and the actual value. As seen in Equation 16 Levenberg-Marquardt algorithm is used as the training function in the model and Gradient Descent with Momentum is used as the adaptive learning function for updating the weights.

$$\Delta W = - (J^T J + \lambda I)^{-1} JT_e \tag{16}$$

Here;
ΔW: Optimized error function, $J^T J$: Approximation of the Hesse matrix, λI: Regularization term, $J^T e$: Gradient value

The error function was optimized with the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963). The gradient of the error function was calculated with the Gradient Descent with Momentum method, and the weights and biases of the network were updated. Finally, the data from the hidden layer was processed using PURELIN (linear activation function) in the output layer and the carbon price was estimated.

Figure 4: Heat map of the correlation



4.3.2. The recurrent neural network (Elman) design

Elman Neural Network was developed using MATLAB Neural Network Toolbox. As seen in Figure 6, the network consists of an input layer consisting of 8 neurons, a hidden layer containing 10 neurons, a context layer providing feedback connection from the hidden layer to the input layer, and an output layer containing 1 neuron.

Elman network has a structure like classical feedforward neural networks. However, the context layer enables it to model temporal dependencies by considering previous time steps. TANSIG was used as the activation function in the hidden layer of the model.

The training of the network was performed with the Levenberg-Marquardt algorithm to optimize the error function, and Gradient Descent with Momentum was applied as the adaptive learning function for weight updates.

4.4. Performance Tests and Evaluation of Results

Several evaluation metrics are used to compare the performance of prediction models. In this study, mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), R-square (R^2) metrics are used to test and compare the performance of Feed forward Neural Network (FFNN) and Elman Neural Network models. The metrics are calculated as in Equations 17, 18, 19 and 20:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{17}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{18}$$

Figure 5: Feed forward neural networks design

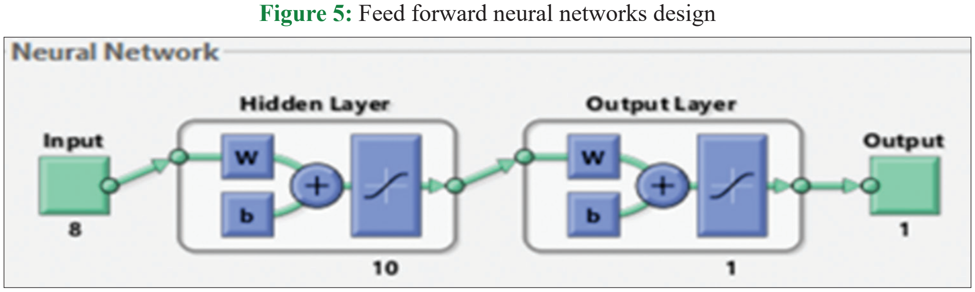
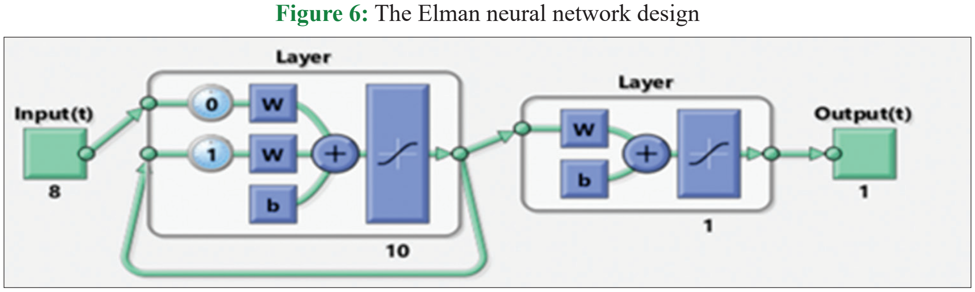


Figure 6: The Elman neural network design



$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \quad (19)$$

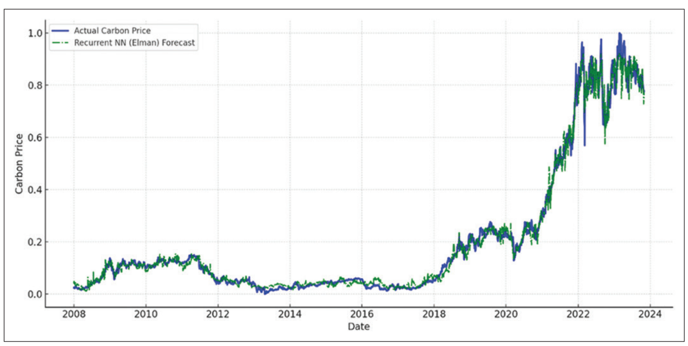
$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \quad (20)$$

Table 4: Performance results

Forecasting model/ performance test	MAE	MSE	RMSE	R^2
FFNN	0.016248	0.000524	0.022896	0,9961
Elman	0.015831	0.000521	0.022833	0.9923

Figure 7: Time series of actual carbon price and FFNN forecast

Figure 8: Time series of actual carbon price and elman RNN forecast



Error levels were examined during the training phase. The iteration that provided the best validation performance was determined and training was terminated at this stage. The purpose of interrupting at this point is to prevent overfitting and to preserve the generalization ability of the model.

The performance results of the Feedforward Neural Network and Elman Neural Network models are shown in the Table 4.

When the results are examined, it is seen that the Elman network produces slightly lower errors than the FFNN model in terms of error metrics. However, the FFNN model reached a higher R^2 value by explaining the relationships between the variables better. In general, the performances of the two models are quite close to each other, and the choice can be determined according to the purpose of use. If low error rate is a priority, the Elman network can be preferred; however, if it is desired that the variables have a high explanatory power for the dependent variable, FFNN may be a more suitable choice.

The carbon price forecast performances of FFNN and Elman Network models were also analyzed with the help of time series analysis as seen in Figure 7 and 8. All figures were created using Python 3.7. While the FFNN model produced more stable forecasts in periods with low volatility (2008-2016), it showed some large deviations in periods with high volatility (after 2020). Elman network is more sensitive to sudden price changes because it takes historical data into account. However, this situation caused the model to show excessive volatility and high fluctuations occurred in the forecasts in some time periods. As a result, FFNN follows the general trend better, while Elman network reacts faster to short-term price movements. Model preference may vary depending on the purpose of use.

Figure 9 shows the error distributions of the FFNN and Elman models. According to the analysis results, the error distribution of the Elman model is wider than the FFNN model. This shows that there are deviations in Elman estimates in some periods. Since the error distribution of the FFNN model is narrower and moves to the center, it is seen that it provides more stable results.

Figure 9: Error distributions of The FFNN and Elman

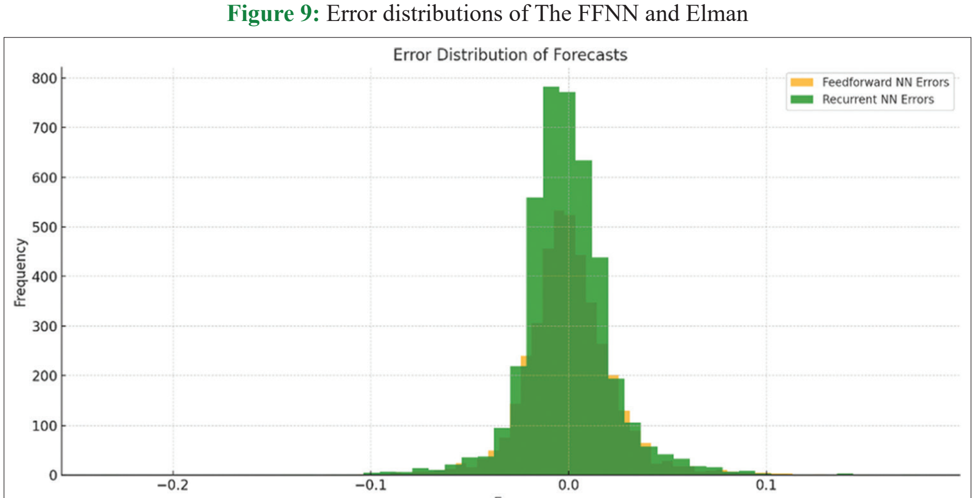


Figure 10: Scatter plot of FFNN

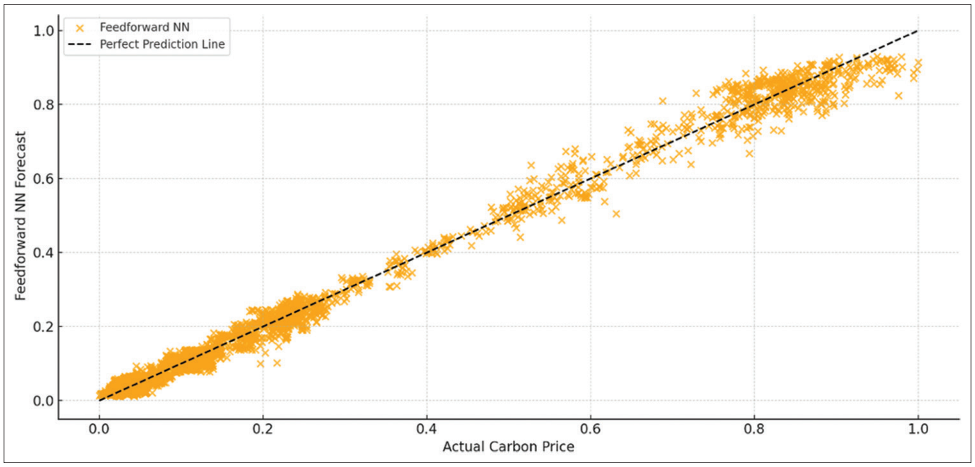
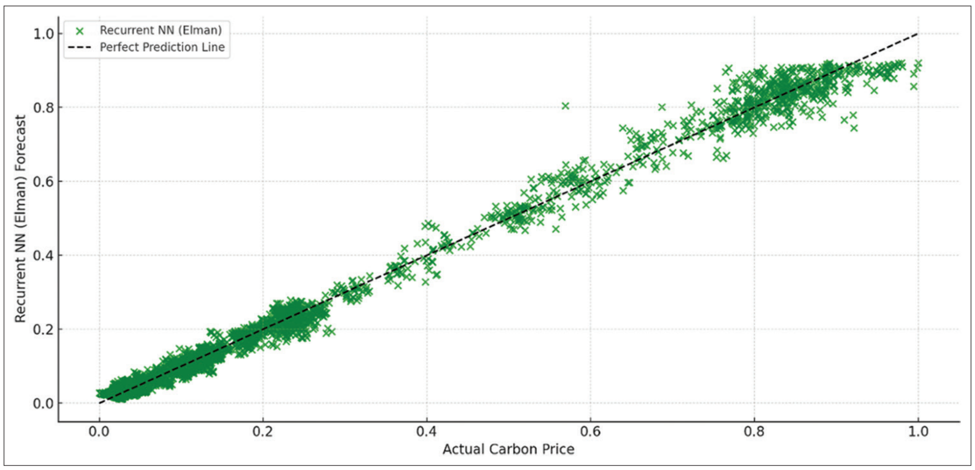


Figure 11: Scatter plot of Elman



Finally, the forecast performances of the models were examined with scatter plots as seen in Figure 10 and 11. The scatter plot shows the distribution of deviations of the predicted values from the actual values. Although both models showed successful forecast performance, the FFNN model forecasts were closer to the perfect forecast line than the Elman forecasts. Especially in periods when prices were high, the Elman forecast spreads were greater. The scatter plot analysis shows that both models showed successful results, but the FFNN model provided more stable results, and the Elman forecasts had more deviations in volatile periods.

5. CONCLUSION

Accurate estimation of carbon prices is highly important for economic, environmental and political decision-making processes. One of the most effective ways to limit greenhouse gas emissions in the battle against global warming and climate change is through carbon pricing, a market-oriented policy tool. Hence, forecasting the future price of carbon is valuable for long-term planning for the policy makers and the industry.

The calculation of carbon prices not only mitigates the uncertainties on the markets but has also a high relevance concerning sustainability, energy, industry and global climate targets. As such, the use of AI-based estimation models on the basis of property and the reliance on data-driven mechanisms are constructive to the more mature management of carbon markets.

In this study, carbon prices (EUA) in European Union were estimated using deep learning methods such as Feedforward Neural Network (FFNN) and Recurrent Neural Network (Elman RNN). Error metrics such as MAE, MSE, RMSE and R^2 were calculated to evaluate the performance of the models and the estimation results were analyzed with different visuals.

When the results are examined, the FFNN model generally produced more stable and systematic estimates, and the error distribution was more central. The Elman RNN model performed slightly better than FFNN in terms of average error values (MAE, MSE, RMSE) because it has the advantage of taking historical data into account. However, the Elman model showed some large estimate deviations, especially in carbon prices with high volatility. The FFNN model achieved a higher R^2 value in validation performance and produced results closer to the perfect estimate line.

In general, both models were successful for carbon price estimates, but it was concluded that the model selection should be made according to the purpose of use. The FFNN model produces more stable and low-variance estimates because it focuses directly on the input-output relationship. Therefore, it was determined that it is a suitable method for long-term trend analyses in carbon prices. The Elman model provided an advantage in short-term estimates because it could learn time series dependencies thanks to feedback connections. However, since it showed excessive volatility due to its dependence on historical information, estimate errors increased in certain periods.

In line with the findings obtained, various suggestions can be made for model selection and improvement studies in carbon price estimates. First, when the estimated variable is examined specifically, additional variables affecting carbon prices (e.g. global economic indicators, energy demand, political regulations, weather conditions) can be included in the model to increase the accuracy of the estimate. Accuracy can be increased by testing different deep learning estimate models or using hybrid models, and both stable and time-dependent estimates can be obtained. Accuracy analysis can be expanded by making comparisons by testing with machine learning-based models.

Accurate estimation of carbon prices contributes to the creation of more effective and predictable climate policies. Reducing uncertainties in carbon markets accelerates the transition to a low-carbon economy by supporting the more efficient operation of regulations such as emissions trading systems and carbon taxes.

Reliable estimate models for policy makers facilitate the planning of sectoral transformation strategies and help develop incentive mechanisms that minimize the impact of high carbon prices on industry. In addition, analyzing carbon price differences at an international level can create more balanced carbon markets and reduce the risk of carbon leakage.

As a result, carbon price forecasts increase the effectiveness of climate policies, support investor confidence and contribute to low-carbon economic growth. Therefore, integrating forecast models into policy-making processes by policymakers will increase the transparency of carbon markets and promote sustainable development.

REFERENCES

Benz, E., Trück, S. (2008), Modeling the price dynamics of CO_2 emission allowances. Energy Economics, 31(1), 4-15.

Byun, S.J., Cho, H. (2013), Forecasting carbon futures volatility using GARCH models with energy volatilities. Energy Economics, 40, 207-221.

Claveria, O., Monte, E., Torra, S. (2014), Tourism demand forecasting with neural network models: Different ways of treating information. International Journal of Tourism Research, 17(5), 492-500.

Creti, A., Jouvet, P., Mignon, V. (2011), Carbon price drivers: Phase I versus Phase II equilibrium? Energy Economics, 34(1), 327-334.

Elman, J.L. (1990), Finding structure in time. Cognitive Science, 14(2), 179-211.

Fine, T.L. (1999), Feedforward Neural Network Methodology. Springer Science and Business Media.

García-Martos, C., Rodríguez, J., Sánchez, M.J. (2012), Modelling and forecasting fossil fuels, CO_2 and electricity prices and their volatilities. Applied Energy, 101, 363-375.

Granger, C.W.J. (1969), Investigating causal relations by econometric models and cross-spectral methods. Econometrica, 37(3), 424.

Huang, Y., He, Z. (2020), Carbon price forecasting with optimization prediction method based on unstructured combination. The Science of the Total Environment, 725, 138350.

Ji, L., Zou, Y., He, K., Zhu, B. (2019), Carbon futures price forecasting based with ARIMA-CNN-LSTM model. Procedia Computer Science, 162, 33-38.

Koch, N., Fuss, S., Grosjean, G., Edenhofer, O. (2014), Causes of the EU ETS price drop: Recession, CDM, renewable policies or a bit of everything?-New evidence. Energy Policy, 73, 676-685.

Koop, G., Tole, L. (2012), Forecasting the European carbon market. Journal of the Royal Statistical Society Series a (Statistics in Society), 176(3), 723-741.

Kumar, N., Kayal, P., Maiti, M. (2024), A study on the carbon emission Futures price prediction. Journal of Cleaner Production, 2024, 144309.

Levenberg, K. (1944), A method for the solution of certain non-linear problems in least squares. Quarterly of Applied Mathematics, 2(2), 164-168.

Li, H., Huang, X., Zhou, D., Cao, A., Su, M., Wang, Y., Guo, L. (2022), Forecasting carbon price in China: A multimodel comparison. International Journal of Environmental Research and Public Health, 19(10), 6217.

Marquardt, D.W. (1963), An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics, 11(2), 431-441.

McCulloch, W.S., Pitts, W. (1943), A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), 115-133.

Narassimhan, E., Gallagher, K.S., Koester, S., Alejo, J.R. (2018), Carbon pricing in practice: A review of existing emissions trading systems. Climate Policy, 18(8), 967-991.

Qin, Q., Huang, Z., Zhou, Z., Chen, Y., Zhao, W. (2022), Hodrick-prescott filter-based hybrid ARIMA-SLFNs model with residual decomposition scheme for carbon price forecasting. Applied Soft Computing, 119, 108560.

Ren, G., Cao, Y., Wen, S., Huang, T., Zeng, Z. (2018), A modified Elman neural network with a new learning rate scheme. Neurocomputing, 286, 11-18.

Rosenblatt, F. (1958), The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386-408.

Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986), Learning representations by back-propagating errors. Nature, 323(6088), 533-536.

Sanin, M.E., Violante, F., Mansanet-Bataller, M. (2015), Understanding volatility dynamics in the EU-ETS market. Energy Policy, 82, 321-331.

Unal, P. (2025), The interrelation between the carbon trading systems and energy markets and economic outlook: A comparative analysis using VECM and ARDL. Economic Studies Journal, 3, 145-169.

Wang, J., Sun, X., Cheng, Q., Cui, Q. (2020), An innovative random forest-based nonlinear ensemble paradigm of improved feature extraction and deep learning for carbon price forecasting. The Science of the Total Environment, 762, 143099.

Wang, S., Zhang, Y., Dong, Z., Du, S., Ji, G., Yan, J., Yang, J., Wang, Q., Feng, C., Phillips, P. (2015), Feed-forward neural network optimized by hybridization of PSO and ABC for abnormal brain detection. International Journal of Imaging Systems and Technology, 25(2), 153-164.

Zhang, W., Wu, Z. (2021), Optimal hybrid framework for carbon price forecasting using time series analysis and least squares support vector machine. Journal of Forecasting, 41(3), 615-632.

Zhou, F., Huang, Z., Zhang, C. (2022), Carbon price forecasting based on CEEMDAN and LSTM. Applied Energy, 311, 118601.

Zhu, B., Shi, X., Chevallier, J., Wang, P., Wei, Y. (2016), An adaptive multiscale ensemble learning paradigm for nonstationary and nonlinear energy price time series forecasting. Journal of Forecasting, 35(7), 633-651.