Int. J. Nonlinear Anal. Appl. In Press, 1–19
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2025.36498.5373

Comparative Analysis of Risk Hedging Models in the Petrochemical Industry: LSTM vs Linear Regression

Morteza Robatmili[a,*], Mohammad Ali Habibi Vostakolayi[b]

[a] *M. A., Department of Accounting, Kharazmi University, Tehran, Iran*
[b] *Department of Accounting, S.R.C., Islamic Azad University, Tehran, Iran*

(Communicated by Majid Eshaghi Gordji)

Abstract

This article aims to use the neural network approach to model risk coverage for petrochemical companies and compare its results with the traditional linear regression model. In this regard, the type and characteristics of the contracts used for risk hedging in this market, the list of included goods and fees were identified in the Iran Commodity Exchange, which shows that the history of using this tool in the petrochemical market includes cases of seasonal forward contracts for products such as polystyrene, Heavy polyethylene, PVC, urea and urea granules are returned. In the following, using a short-term long-term memory model, risk hedging is done using this contract in a selected market (polystyrene), the results of which show the clearly better performance of the neural network model compared to the linear regression model, as well as the increasing trend of the optimal coefficient. Risk coverage is in the period of six years from the spring of 2017 to the winter of 2023 in both models. The results also showed that in all final versions, the short-term long-memory neural network models performed better than the linear regression model. Models with a training ratio of 30% have relatively better performance than models with a training ratio of 40%. The polystyrene market study shows that the optimal seasonal hedging rate has been increasing mainly during the six-year period from spring 2017 to winter 2023. It also seems that the neural network model with a training ratio of 30% has the ability to adapt faster in response to market changes than other models, which could be due to its better training.

Keywords: (Risk hedging, risk management, petrochemical market, long short term memory model, commodity derivatives)
2022 MSC: O20, O10

1 Introduction

The petrochemical market has an important position in the country's economy, accounting for 30% of the country's capital market and 40% of Iran's non-oil exports. On the other hand, the changes in the pricing system in Blandmet usually face the petrochemical market with an increase in price risk. Identifying the common models, choosing the appropriate model, optimizing and interpreting the outputs of these models according to the accurate knowledge of the influencing factors and the macroeconomics will help the future decision makers of this market to use the capacities of commodity derivatives for risk hedging and economic exploitation and cause It provided the stable growth of the

*Corresponding author
Email addresses: mortezarobatmili@gmail.com (Morteza Robatmili), alihabibi.e9595@gmail.com (Mohammad Ali Habibi Vostakolayi)

Received: january 2025 *Accepted:* February 2025

petrochemical market. Coinciding with the fifth wave of the industrial revolution, the adoption of artificial intelligence for pricing and hedging using derivative instruments is increasingly growing [23], and recent research has specifically focused on the use of neural networks and reinforcement learning to achieve They focus on an optimal risk hedging strategy [5]. Hedging policy has positive economic benefits and will lead to market expansion in the long run [25]. In this regard, the article aims to identify the current situation of risk hedging in the Iranian petrochemical market, implement and develop the use of the neural network approach in market risk hedging for Iranian petrochemical companies, in order to provide insights and recommendations for market parties. especially petrochemical companies) and future researchers to expand this sector even more.

The process that initially includes the review of the risk hedging tools available in the Iranian petrochemical market and then includes the selection and optimization of the neural network model according to the needs and conditions of the market, as well as the evaluation of the performance of this model compared to the traditional linear regression model.

2 Research background

In the following, we will examine and compare previous studies in the field of market risk coverage. In recent years, the number of researches that have used neural networks in the discussion of derivatives pricing and risk management is increasing, and nowadays neural networks are widely used to solve problems related to options pricing, simulating the behavior of derivatives, and risk hedging [16].

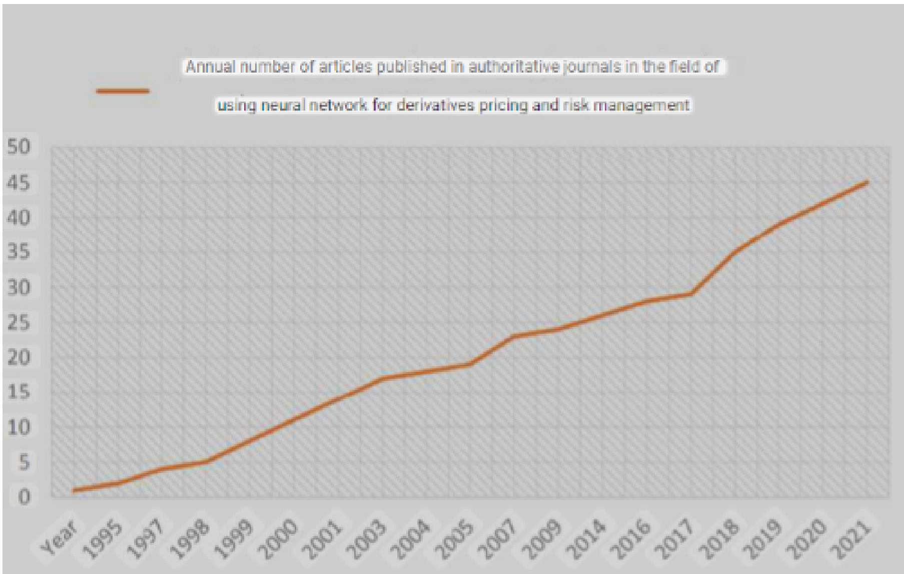


Figure 1: the increase in the use of neural networks in financial discussions Kumar [16]

2.1 Foreign studies

Pagnottoni [20] stated that the neural network model has a significant improvement in price prediction. Carverhill colleagues [6] concluded that neural network can provide more accurate hedging parameters than traditional options pricing methods. Bennell & Sutcliffe [4] showed that for out-of-the-money, the neural network performed better. In the case of in-the-money with long maturity, neural and linear network perform the same. Barunıkova et al.'s [3] results showed that in European call and put option pricing, feed forward neural network has defeated the linear model in both buying and selling cases. Shabri & Ruhaidah [24] showed that the combined model of wavelet transform and neural network has a more appropriate performance in comparison with neural network, ARIMA and GARCH models for forecasting Brent and West Texas crude oil prices. Hirsa et al.[13] used supervised deep neural network for option pricing. Their results showed that deep neural networks are up to 10 times faster than conventional methods. Horvath et al [14] presented a fast calibration technique based on neural network. This method is applicable for a wide range of fluctuations and types of derivatives. In this research, neural networks are used to approximate complex pricing functions that are difficult to analyze with common methods. Du Plooy & Venter [10] compared the pricing performance of two learning networks. They used neural network and bootstrap aggregation network to price European options on the Johannesburg Stock Exchange (JSE). They also compared the numerical accuracy of the finer grid with Monte Carlo simulation in a separate study. When they compared the results, they observed that the bootstrap aggregating method performed better than the neural network and predicted the price without Monte Carlo simulation errors. Zhang et al.[28] presented a hybrid strategy based on deep neural network for short-term electricity price forecasting. More precisely, the proposed technique includes the Categorical Boosting algorithm for feature selection and a two-way neural network with short-term memory for prediction. They used Nord Pool hourly market data from 2018 onwards to check the performance of the model. They compared the performance of their proposed model with the results of MLP neural network, SVM, Ensemble tree, ARIMA and two deep learning based models including LSTM and GRU recurrent unit neural network. The results show that the proposed model has less prediction error than other methods, although its training and prediction require more time.

The following table shows a list of cited foreign articles and researches along with the type of neural network model used in them, the application and advantage of those models in chronological order.

Table 1: neural network models used in the field of market risk hedging, along with detailed application in modeling and advantages

field of excellence	**Application**	**Neural network method and model**	**source**
precision	Better pricing	Multilayer perceptron and radial basis functions	Dindar & Marwala [9]
precision	Improving accuracy in price forecasting	Neural network and support vector regression	Liang et al. [17]
precision	Simplifying option price forecasting	An improvement over the HLP model	Montesdeoca, Niranjan [19]
Calculation speed	Generalization of the pricing function	Gated Neural Network	and Yang colleagues [27]
Advanced compliance	Better pricing	Supervised deep convolutional neural network	Dimitroff et al. [8]
precision	Better management of hedging errors	General neural network	De Gennaro et al. [7]
Calculation speed	Quick calculation of Greek options	Using neural network as a derivation algorithm	Ratku, A., & Neumann [23]
Calculation speed	Calculating the fast path integral	Feedforward neural network	Vadori [26]
precision	More accurate pricing of Greek options and options	Genetic algorithm of re-propagation	Qian et al. [22]

2.2 Internal studies

Naini and Kazemi manesh [15] studied the optimal hedging rate of the oil market during the years 1999 to 2003. Also, in the same year, Yousefi et al. used a wavelet-based model to forecast the price of crude oil in different periods. Behradmehr [2] used wavelet transform to reduce the noise in oil price forecast data. The results showed that the performance of the model has improved in forecasting and it performs better than competing models. Ebrahimi and Ghanbary [11] studied the risk coverage of Iran's oil revenues and showed that by using futures contracts, the amount of risk is significantly reduced. They also calculated the risk and desirability of each of the future contracts and concluded that these contracts with longer maturity are more effective in reducing the risk of oil price fluctuations. Their results showed that the combined model has a better performance in forecasting. Compared to the ARMA model, the combined model has a lower prediction error (RMSE). Fallahpour and Ahmadi [12] investigated the value at risk of oil and gold portfolio with Copula-Garch approach and using data from 2012 to 2017. Their results showed the superiority of this method over traditional methods in the accuracy of risk calculation. Ansari et al [1] studied the optimal rate of risk coverage in the natural gas market from 2000 to 2013 using the futures contract in the stock market. They showed that the use of futures contracts is justified and that the hedging rate increases as their maturity lengthens. Mohammadi et al [18] used neural network and wavelet to predict the daily price of natural gas. Based on the prediction error criterion, the combined model of neural network and wavelet has performed better than the neural network. Pishbahar and Abedi [21] used the copula method to calculate the Value at Risk of a portfolio consisting of dairy products and sugar. Their results showed that the Copula-based Monte Carlo simulation method performed better than other methods.

2.3 Summary of research background and statement of research gaps

The methods based on neural networks, with higher accuracy, speed and computing power than traditional methods, perform better in terms of various criteria in the financial field, including risk coverage, valuation and predictability. Price, interest rate, remaining time to maturity and price volatility are the most important inputs of the neural network. Recurrent neural network, feed forward, convolution and ong short term memory are among the best and most widely used neural network models according to data structure, relationship complexity, time horizon and research type. Among the different types of neural network, the short-term long-term memory model is one of the best in the field of risk coverage. However, this method has not been used in Iran's financial discussions and this article aims to use this tool for the first time. In the country, it provided the ground for future researches and activities in this field. Also, the internal researches are mainly focused on the oil and gas market or other markets, and the upcoming article is different in terms of focusing on the petrochemical market.

3 Risk hedging tools in the Iranian petrochemical market

All transactions in the Iran Commodity Exchange are carried out in the form of the following standardized contracts:

1) Cash contract
2) Predecessor contract
3) Credit agreement
4) Futures contract
5) Standard self-debt contract
6) Option contract
7) Commodity boxes
8) Goods deposit certificate

A review of the underlying assets on which these derivative contracts were issued shows that it is the only standard forward contract that has a history of being used in petrochemical transactions. The list of assets and fees of the standard parallel forward contract includes petrochemical products such as heavy polyethylene, polystyrene, urea granules, urea and PVC. The description of the transaction fees of standard parallel forward contracts, which are mainly concluded with a quarterly maturity (three months), is as follows.

- **Return and Risk Formula**

$$R = \frac{P_t - P_{t-1}}{P_{t-1}} \times 100$$

  where

  - R:Asset return
  - P_t: Price at time t
  - P_{t-1}: Price at time $t-1$

- **Pearson Correlation Coefficient** (to analyze relationships between variables)

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

  where

Table 2: transaction fee of standard parallel forward contract in Iran Commodity Exchange according to the latest update until spring 2023

Fee description	**shopping**	**sale**	**Collect**	**Maximum amount of sales commission**	**Maximum purchase fee amount**	
Brokers' fees	0.0006	0.0006	0.0012	50	50	100
Fee of Iran Commodity Exchange Company	0.0001	0.0001	0.0002	20	20	40
Central depository company fee	0.000015	0.000015	0.00003	6	6	12
Tehran Stock Exchange technology management fee	0.00001	0.00001	0.00002	4	4	8
Collect all	0.000724	0.000724	0.00145			

- ρ_{xy}: Correlation coefficient between variables x and y
- x_i, y_i: Individual data points
- $\bar{x}$, $\bar{y}$: Means of x and y

- **Volatility (Standard Deviation of Returns)**

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(R_i - \bar{R})^2}$$

where

- σ: Volatility
- R_i: Individual returns
- $\bar{R}$: Mean return

4 Methodology

Selection of model type and input variables, number of layers, nodes and their relationship with each other and model parameters are among the defining factors of neural network model architecture. As stated in the background of the research in the second chapter, a variety of neural network models can be used to cover market risk, and factors such as research objectives, data access, data structure, relationship complexity, and time horizon determine the type of neural network model. For this reason, in this section, we go to the most common models in terms of use in the research background, and then, by enumerating the features of the article, we choose the appropriate neural network model.

Between 2017 and 2023, several petrochemical companies were active on the Tehran Stock Exchange. The following are the sample companies of this study, which are among the most important of these companies, along with their trading symbols and relevant markets:

Company Name	**Trading Symbol**	**Market**
Persian Gulf Petrochemical Industries	FARS	Tehran Stock Exchange (TSE) - Secondary Market
Nouri Petrochemical	NOURI	TSE - Secondary Market
Pars Petrochemical	PARS	TSE - Secondary Market
Shazand Petrochemical	SHARAK	TSE - Primary Market (Main Board)
Jam Petrochemical	JAM	TSE - Secondary Market
Pardis Petrochemical	SHEPDIS	TSE - Primary Market (Sub-Board)
Shiraz Petrochemical	SHIRAZ	TSE - Primary Market (Main Board)
Khark Petrochemical	SHKHARK	TSE - Primary Market (Sub-Board)
Fanavaran Petrochemical	SHFAN	TSE - Primary Market (Main Board)
Khorasan Petrochemical	KHORASAN	TSE - Secondary Market
Kermanshah Petrochemical	KERMASHA	TSE - Primary Market (Main Board)
Ghadir Petrochemical	SHGHADIR	TSE - Secondary Market
Bou Ali Sina Petrochemical	BOUALI	TSE - Secondary Market
Tondgouyan Petrochemical	SHGOOYA	Iran Fara Bourse (IFB) - Secondary Market
Arya Sasol Polymer	ARYA	IFB - Secondary Market
Zagros Petrochemical	ZAGROS	IFB - Secondary Market
Marun Petrochemical	MAROON	IFB - Secondary Market
Urmia Petrochemical	SHAROOM	IFB - Primary Market
Amir Kabir Petrochemical	SHKABIR	IFB - Yellow Base Market
Abadan Petrochemical	SHEPETRO	IFB - Yellow Base Market

The choice of the type of model used depends on several factors, including the amount of access and data structure, the time horizon considered, the complexity of the relationships between variables, the necessity or lack of real time decision making, hardware facilities and computing resources.

Price data available for modeling includes daily data in a ten-year period from 2013 to 2023 and the total number of observations is 2639. As a result, data access can be considered at a medium and high level. The data structure is in the form of time series tables, and the assumption of linear relationships between variables is a limiting and unjustified assumption. Considering the current state of forward contracts in the petrochemical market, the considered time horizon is medium-term (three months) and according to the scientific purpose of the article, it is not necessary to make a decision in real time. According to these points, the table below shows the recommendability of different neural network models.

Table 3: examining the fields of using common models based on the characteristics of the article

Model / Index	FNN	CNN	RNN	LSTM
Data access			√	√
Data structure	√	√		√
Complexity of relationships			√	√
time horizon	√			√
Real time decision making	√	√		√

Based on this, it seems that the most recommended model to use for hedging the market risk in the upcoming article is the Long Short Term Memory (LSTM) model.

- **Linear Regression Model**

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

where

  - y: Dependent variable (e. g. risk-hedging performance)
  - x_i: Independent variables (e. g. historical priece, external factors)
  - β_0: Intercept
  - β_i: Coefficients

  - ε: Error term

- **LSTM Cell Operations**

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C)$$
$$C_t = f_t \star C_{t-1} + i_t \star \tilde{C}_t$$
$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \star \tanh(C_t)$$

  where:

  - f_t: Forget gate
  - i_t: Input gate
  - $\tilde{C}_t$: Candidate memory cell
  - C_t: Cell state
  - o_t: Output gate
  - h_t: Hidden state
  - σ: Sigmoid activation function
  - tanh: Hyperbolic tangent activation

4.1 Neural network model input data

The selection of input data will be based on the variables that are conventionally used in the modeling cited in the background of the research. However, the exact selection of input variables for modeling in this article is appropriate to the characteristics of the article, including its goals and data access. As a result, 10 input variables will be used for the neural network model, which are polystyrene price, oil price, natural gas price, chemical fertilizer price, naphtha price, inflation, dollar price, economic growth, export and seasonal data. Price and technical data (such as the price of the product itself, oil price, gas price, naphtha price, chemical fertilizer price) extracted from the Iran Commodity Exchange and macro data (such as economic growth, export, exchange rate, energy consumption in the country and inflation) extracted From the Central Bank of the Islamic Republic of Iran.

4.2 Model Implementation of short-term long memory

The implementation of the model is done in the Python software environment. The implementation of the model includes several different phases. The first phase is related to equipping the analytical device with tools suitable for the purposes of the article; The second phase is related to entering data into the system; The third phase of data division for training and validation; The fourth phase is focused on building the model, training and validation, optimization, and finally the final phase is dedicated to receiving and evaluating the outputs and comparing them with the traditional model.

4.3 Data segmentation for training and validation in the model

The data partition ratio is often determined through testing and cross-validation, where different ratios are tried and the performance of the model is checked. The proportion of training data can vary and is usually chosen between 20% and 40%. As a result, we will run the model with three training ratio scenarios of 20%, 30% and 40% and we will check its results. In addition to different ratios of training, the previous time period to influence the target variable is implemented in the present with three scenarios of 50, 100 and 200 days.

4.4 Model construction, training and validation

In this section, the construction of long-short-term memory model, training and performance evaluation of the model will be discussed. Parameters that determine the architecture of the model such as the number of layers, the number of neurons in each layer, and hyper-parameters of the model such as Epochs, which specifies the number of times the model is repeated over the entire training dataset, and Batch-size, which determines the number of samples per gradient update, will be tested by checking the outputs of the model. For example, start with a small size like 8 and gradually increase it, while monitoring the performance of the model, looking for the point at which increasing the batch size no longer improves the performance of the model or leads to worse performance. will be

We implement the short-term long-term memory model with three scenarios of 1 layer, 2 layers and 3 layers and with the number of neurons 32, 64 and 128 and examine the outputs. The following code shows the construction, training and validation of a long-short-term memory hedging model with an inner layer with 64 neurons. The model is repeated 10 times in the entire training set, and the number of samples before updating the weights is 32.

4.5 LSTM Model Formulas

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. Below are the key formulas that define the operations within an LSTM cell:

1. **input-gate**

   The input gate determines how much of the new information should be stored in the cell state. Where:

   $$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)$$

   - i_t: Input gate activation vector at time t
   - σ: Sigmoid activation function
   - W_i: Weight matrix for the input gate
   - h_{t-1}: Hidden state from the previous time step
   - x_t: Input at time t
   - b_i: Bias vector for the input gate

2. **Forget Gate**

   The forget gate decides what information to discard from the cell state.

   $$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)$$

   Where:

   - f_t: Forget gate activation vector at time t
   - W_f: Weight matrix for the forget gate
   - b_f: Bias vector for the forget gate

3. **Cell State Update**

   The cell state is updated by combining the previous cell state and the new candidate values.

   $$\tilde{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C)$$

   $$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

   Where:

   - $\tilde{C}_t$: Candidate cell state at time t
   - tanh: Hyperbolic tangent activation function
   - W_c: Weight matrix for the cell state
   - b_c: Bias vector for the cell state

- C_t: Updated cell state at time t
- $\odot$: Element-wise multiplication (Hadamard product)

4. **Output Gate**

The output gate controls how much of the cell state is exposed to the next hidden state.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot \tanh(C_t)$$

- o_t: Output gate activation vector at time t
- W_o: Weight matrix for the output gate
- b_o: Bias vector for the output gate
- h_t: Hidden state at time t

Summary of Variables:

- x_t: Input at time t
- h_t: Hidden state at time t
- C_t: Cell state at time t
- i_t, f_t, o_t: Input, forget, and output gate activations
- W_i, W_f, W_C, W_o: Weight matrices
- b_i, b_f, b_C, b_o: Bias vectors

These formulas collectively define the operations within a single LSTM cell, enabling it to learn and remember long-term dependencies in sequential data.

5 Results

The essence of this section is to evaluate the performance of different versions of long-term memory with different parameters and hyperparameters, which in a feedback process helps to analyze the performance of the model and reach the optimal output.

5.1 Hyperparametric optimization

To begin with, you can start by finding the optimal amount of Epochs and Batch-size hyper-parameters and the number of model layers. The working method here is to compare the results based on three performance indicators: coefficient of determination (R^2), root mean square error (RMSE) and mean absolute error (MAE). The following table is related to the outputs of long-short-term memory models with one layer for different values of Epochs (the number of times the model is repeated in the learning set).

- **Mean Absolute Error (MAE)**

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

Where:

  - y_i: Actual value
  - $\hat{y}_i$: Predicted value
  - n: Number of observations

- **Root Mean Squared Error (RMSE)**

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

- **R-squared (Coefficient of Determination)**

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

- **Diebold-Mariano Test (to compare predictive accuracy)**

$$DM = \frac{\bar{d}}{\sqrt{\frac{2\pi\hat{y}_0}{T}}}$$

Where:

- $\bar{d}$: Mean loss differential
- $\hat{y}_0$: Variance of the loss differential
- T: Sample size

Table 4: performance evaluation of one-layer long-short-term memory model based on different values of Epochs

Epochs	8	16	32	64
R^2	0.60	0.64	0.69	0.36
RMSE	0.21	0.19	0.18	0.22
MAE	0.16	0.15	0.14	0.16

As can be seen from the table above, the performance of the model improves at first by increasing the number of repetitions of the model for learning (that is, reducing RMSE and MAE and increasing R^2 until the best performance corresponds to the number of 32 Epochs, but after that the performance of the model weakens. It becomes until the explanatory power of the model reaches half in 64 epochs. The following table also shows the outputs corresponding to different number of epochs for the model with two layers.

Table 5: evaluation of the performance of the two-layer long-short-term memory model based on different values of Epochs

Epochs	8	16	32	64
R^2	0.65	0.73	0.71	0.36
RMSE	0.20	0.18	0.20	0.20
MAE	0.16	0.14	0.15	0.16

What emerges from the outputs of this table is that, firstly, the performance of the two-layer model is better than the performance of the one-layer model in all scenarios. Secondly, the model with the number of Epochs 16 has the best performance among the two-layer short-term long memory models. The following table also shows the outputs corresponding to different number of Epochs for the three-layer model.

Table 6: performance evaluation of the three-layer long-short-term memory model based on different values of Epochs

Epochs	8	16	32	64
R^2	0.90	0.67	0.75	0.59
RMSE	0.17	0.18	0.17	0.20
MAE	0.13	0.14	0.14	0.16

The outputs of this table show that the performance of the three-layer model is always better than the performance of the one-layer model, but compared to the two-layer models, this superiority is somewhat relative, which means that out of the four modes, the performance is better in three modes, but in one mode, it corresponds to 16 epochs. The performance is also weaker. Also, the results of Table 6 show that the best performance in the three-layer models corresponds to 8 epochs.

So far, we have seen that if all the conditions are constant and Epochs can be changed, the two-layer and three-layer models clearly have better performance than the one-layer models. Three-layer models also generally perform better than two-layer models, and the best performance recorded was for the three-layer model with 8 epochs. Of course, the process of hyper-parametric optimization is not limited here because the Batch-size variable, which represents the number of initial sampling in the model, has not been investigated yet. The following table shows the different outputs of long-short-term memory models of one layer with 32 Epochs and with different Batch-size values. Unlike

Table 7: performance evaluation of single-layer long-short-term memory model with 32 epochs based on different values of Batch-size

Epochs	8	16	32	64
R^2	0.73	0.34	0.68	0.69
RMSE	0.20	0.22	0.18	0.18
MAE	0.15	0.17	0.14	0.14

Epochs, changes in Batch-size hyperparameter do not have a uniform effect on changes in performance evaluation indicators. For example, if we consider the performance index R^2, then 8 Batch-size has the best performance, and if we consider the index RMSE or MAE, then 16 Batch-size has the best performance, and the overall performance of the model increases in proportion to the increase in Batch-size It has a fluctuating state that makes decision making more complicated.

Table 8: performance evaluation of the two-layer long-short-term memory model with 32 epochs based on different values of Batch-size

Epochs	8	16	32	64
R^2	0.36	0.49	0.78	0.71
RMSE	0.22	0.19	0.19	0.20
MAE	0.17	0.15	0.13	0.15

The table above shows the results of the two-layer models, which indicates that in the two scenarios of 16 and 32 Batch-size, the performance of the two-layer model is better than that of one layer, while in the case of 8 Batch-size, the performance of the one-layer model is better, and in the case of 64 Batch -size cannot be judged in any way, so it can be said that the two-layer model has relatively better performance. In this table, it can also be seen that the performance of the model is always improved by increasing the Batch-size, and the best performance is recorded for 64 Batch-size. The following table also shows the performance of the three-layer long-short-term memory model for different numbers of Batch-sizes.

Table 9: performance evaluation of the three-layer long-short-term memory model with 32 epochs based on different values of Batch-size

Epochs	8	16	32	64
R^2	0.92	0.73	0.71	0.75
RMSE	0.17	0.18	0.18	0.17
MAE	0.14	0.13	0.14	0.14

The results of the above table contain these points that, firstly, the performance of the three-layer model is clearly better than the two-layer and one-layer models; Secondly, the performance of the three-layer model has a sinusoidal state with respect to Batch-size changes, so that with the increase of Batch-size, it first decreases and then increases.

The results of Tables 2 to 7 can be evaluated in such a way that short-term long memory models with more layers and fewer Epochs perform relatively better. As a result, we focus on the hyperparametric optimization of the model in the framework of the three-layer model and we continue to try to reach the optimal amount of hyperparameters according to the outputs of different scenarios. The following tables show the R^2, RMSE and MAE of the three-layer model for different combinations of Epochs and Batch-size.

Table 10: performance evaluation of three-layer long-short-term memory models based on R^2 for different values of Batch-size and Epochs

Batch-size \ Epochs	8	12	16	24	32	48	64
8	0.83	0.78	0.81	0.62	0.79	0.53	0.37
12	0.87	0.70	0.78	0.71	0.54	0.63	0.25
16	0.84	0.78	0.65	0.77	0.62	0.34	0.49
24	0.85	0.62	0.76	0.70	0.61	0.35	0.35
32	0.86	0.65	0.78	0.78	0.68	0.67	0.66
48	0.81	0.82	0.64	0.75	0.77	0.65	0.62
64	0.87	0.90	0.89	0.83	0.74	0.60	0.44

Table 11: performance evaluation of three-layer long-short-term memory models based on RMSE for different values of Batch-size and Epochs

Batch-size \ Epochs	8	12	16	24	32	48	64
8	0.17	0.17	0.17	0.18	0.18	0.20	0.21
12	0.17	0.19	0.17	0.18	0.19	0.18	0.23
16	0.17	0.17	0.18	0.17	0.18	0.21	0.22
24	0.17	0.19	0.18	0.18	0.19	0.21	0.21
32	0.17	0.18	0.17	0.17	0.18	0.18	0.23
48	0.17	0.17	0.18	0.18	0.17	0.18	0.20
64	0.17	0.16	0.17	0.17	0.18	0.19	0.19

Table 12: performance evaluation of three-layer long-short-term memory models based on MAE for different values of Batch-size and Epochs

Batch-size \ Epochs	8	12	16	24	32	48	64
8	0.13	0.14	0.13	0.14	0.14	0.16	0.17
12	0.13	0.14	0.15	0.13	0.14	0.17	0.17
16	0.13	0.14	0.15	0.13	0.14	0.17	0.17
24	0.13	0.14	0.14	0.14	0.15	0.16	0.17
32	0.13	0.13	0.14	0.13	0.14	0.14	0.19
48	0.13	0.13	0.15	0.14	0.13	0.14	0.15
64	0.13	0.13	0.13	0.13	0.14	0.14	0.15

The results obtained from examining the above tables are as follows: first, the performance of the model is significantly affected by the choice of model hyper-parameters; For example, R^2 fluctuates between 0.24 and 0.89. The performance of the model is usually improved by increasing Batch-size and decreasing Epochs, and the three-layer model with 12 Epochs and 64 Batch-size showed the best performance with the highest R^2 and the lowest RMSE and MAE. Due to the location of this point in the corner of the table, we also tested the model with a higher number of Batch-size to see if there is a significant improvement in the performance of the model. The results showed that the performance of the model did not significantly improve with Batch-sizes of more than 64, and the performance of the model became weaker with the increase of Batch-size. As a result, we consider the three-layer model with 12 epochs and 64 batch-size as the optimal model in terms of hyperparameters.

5.2 Performance evaluation of the three-layer long-short-term memory model for different parameters and comparison with linear regression

As stated in the previous part of this chapter, a three-layer model that is trained 12 times during the period of time and the number of initial samplings is 64 is the most suitable option from the point of view of performance evaluation indicators. What follows is to examine the performance of the selected model in different parametric scenarios and compare this performance with the linear regression model. Two important parameters are decisive in this section; The first parameter is the ratio of training data and the second parameter is the length of the previous effective time period. The following tables show the performance of the three-layer model in different parametric scenarios (training rate 20%, 30%, 40% and the effective previous influence period of 50, 100 and 200 days) and its comparison with the performance of the linear regression model (regression outputs in section 6- 5 have been calculated)

Table 13: performance evaluation of three-layer long short-term memory models based on R^2 for different values of training rate and effective time period, the asterisked value is the output of the linear regression model.

Training rate / Effective time period	%20	%30	%40	
50	-	0.92	0.74	
100	-	0.90	0.85	
200	-	0.88	0.83	
				$0.62^{\star}$

Table 14: performance evaluation of three-layer long-short-term memory models based on RMSE for different values of training rate and effective time period, the starred value is the output of the linear regression model

Training rate / Effective time period	%20	%30	%40	
50	-	0.16	0.18	
100	-	0.16	0.16	
200	-	0.16	0.15	
				$0.23^{\star}$

Table 15: performance evaluation of three-layer long-short-term memory models based on MAE for different values of training rate and effective time period, the asterisked value is the linear regression output.

Training rate / Effective time period	%20	%30	%40	
50	-	0.12	0.13	
100	-	0.13	0.13	
200	-	0.12	0.12	
				$0.20^{\star}$

The results of the above tables contain important points; Among other things, the short-term long-term memory model with a training ratio of 20% is basically unable to learn at a level that can lead to conclusions regarding the relationship of variables. Also, models with a training level of 30% generally perform better in terms of R^2. Also, their performance in terms of RMSE and MAE indicators is more or less better with the level of education of 40%. The effect of changes during the previous effective time period (Sequence-length) is mostly uniform in the sense that in terms of the R^2 performance index at the educational level of 30%, there are very few changes and the highest level of performance is related to the 50-day period, but at the level Education is 40% of the best performance related to the 100-day effective course period. From the point of view of the RMSE index, the performance of the model improves or at least does not deteriorate with the increase of the effective period. Also, from the point of view of the MAE index, the performance of the model with an effective historical period of 200 days is the best. Finally, it should be noted that the outputs of the short-term long-term memory model are better than the outputs of the linear regression model in all parametric scenarios and based on all three performance evaluation indicators.

5.3 Calculation of the optimal rate of risk coverage

The following tables show the optimal rate of risk coverage based on the three-layer short-term long-term memory model in different parametric scenarios. According to the more regular form of contracts used in the petrochemical market. three-month forward contracts, we calculate the optimal rate of risk coverage seasonally in the period from spring 2017 to winter 2023.

Table 16: the optimal rate of risk coverage in three-layer short-term long-term memory models, spring 2017 to winter 2020

	Spring 2017	Summer 2017	Autumn 2017	Winter 2018	Spring 2018	Summer 2018	Autumn 2018	Winter 2019	Spring 2019	Summer 2019	Autumn 2019	Winter 2020
LSTM30%50	0.47	0.42	0.45	0.47	0.53	0.48	0.59	0.50	0.64	0.69	0.76	0.86
LSTM30%100	0.48	0.43	0.46	0.46	0.54	0.53	0.57	0.51	0.63	0.68	0.74	0.84
LSTM30%200	0.51	0.43	0.45	0.45	0.59	0.54	0.54	0.54	0.61	0.69	0.73	0.81
LSTM40%50	0.49	0.42	0.39	0.46	0.47	0.50	0.52	0.52	0.60	0.67	0.75	0.69
LSTM40%100	0.44	0.45	0.43	0.41	0.52	0.53	0.54	0.52	0.61	0.66	0.68	0.75
LSTM40%200	0.46	0.41	0.44	0.43	0.48	0.52	0.58	0.54	0.61	0.69	0.66	0.70

Table 17: the optimal rate of risk coverage in three-layer short-term long-term memory models, spring 2020 to winter 2023

	Spring 2017	Summer 2017	Autumn 2017	Winter 2018	Spring 2018	Summer 2018	Autumn 2018	Winter 2019	Spring 2019	Summer 2019	Autumn 2019	Winter 2020
LSTM30%50	0.65	0.62	0.65	0.91	0.71	0.77	0.81	0.84	0.83	0.94	0.96	0.99
LSTM30%100	0.67	0.65	0.70	0.83	0.75	0.76	0.77	0.85	0.81	0.89	0.93	0.96
LSTM30%200	0.70	0.61	0.68	0.81	0.77	0.75	0.79	0.86	0.84	0.87	0.97	1/01
LSTM40%50	0.73	0.83	0.71	0.79	0.78	0.86	0.86	0.85	0.85	0.98	0.83	0.85
LSTM40%100	0.77	0.85	0.69	0.78	0.73	0.89	0.88	0.90	0.91	0.95	0.85	0.87
LSTM40%200	0.76	0.87	0.70	0.81	0.70	0.93	0.89	0.90	0.89	0.97	0.90	0.89

5.4 Regression model implementation and results

In the calculations related to this section, historical data on the price of the base asset (polystyrene) and the corresponding forward contract prices are collected, and then it is time for the regression analysis, where the dependent variable is the forward price and the independent variable is the spot price of the product. This coefficient shows how much of the product should be covered by concluding the forward contract for each product unit.

$$F = \beta_0 + \ \beta_1 P + \varepsilon \tag{1}$$

Before starting the examination of the final results of the regression, it is possible to analyze the components of the regression disturbance to ensure the normality of the distribution of variables and homogeneity of variance. Appendix 1 depicts the distribution diagram of disturbance components, which shows that the condition of normal distribution is satisfied.

Also, Brosh Pagan's test is used to check the heterogeneity of variance, and the results indicate that there is no statistically significant relationship between the components of the discharge and the spot price of the product (Appendix 2). The calculated performance evaluation indicators are as follows.

Table 18: performance evaluation indicators of linear regression model, spring 2017 to winter 2023

R^2	RMSE	MAE
0.62	0.23	0.20

Although these results indicate the significant explanatory power of the regression equation to explain the changes in the futures price based on the spot price, it clearly implies a lower explanatory power and a higher error rate in the regression model compared to the neural network model. The following tables also show the calculated amount of the optimal coefficient of risk coverage for the three-month periods from spring 2017 to winter 2023. The results of the initial regression estimation for the optimal risk coverage rate (β_1) for the seasons of the time period are as described in the table below.

Table 19: the optimal rate of risk coverage based on the linear regression model, spring 2017 to winter 2020

	Spring 2017	Summer 2017	Autumn 2017	Winter 2018	Spring 2018	Summer 2018	Autumn 2018	Winter 2019	Spring 2019	Summer 2019	Autumn 2019	Winter 2020
β_1	0.57	0.54	0.47	0.42	0.44	0.49	0.59	0.56	0.59	0.59	0.60	0.62

Table 20: optimal rate of risk coverage based on the linear regression model, spring 2020 to winter 2023

	Spring 2017	Summer 2017	Autumn 2017	Winter 2018	Spring 2018	Summer 2018	Autumn 2018	Winter 2019	Spring 2019	Summer 2019	Autumn 2019	Winter 2020
β_1	0.74	0.84	0.69	0.79	0.69	0.89	0.89	0.91	0.89	0.97	0.89	0.89

5.5 Comparison of the optimal coefficient of risk coverage

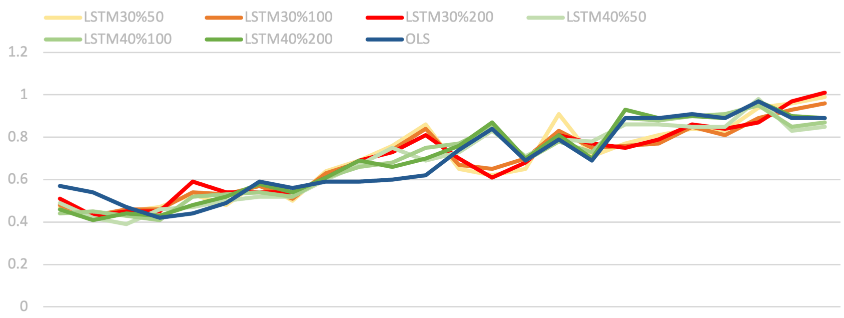


Chart 2. Seasonal optimal rate of risk coverage in three-layer long-short-term memory models and linear regression model, spring 2017 to winter 2023

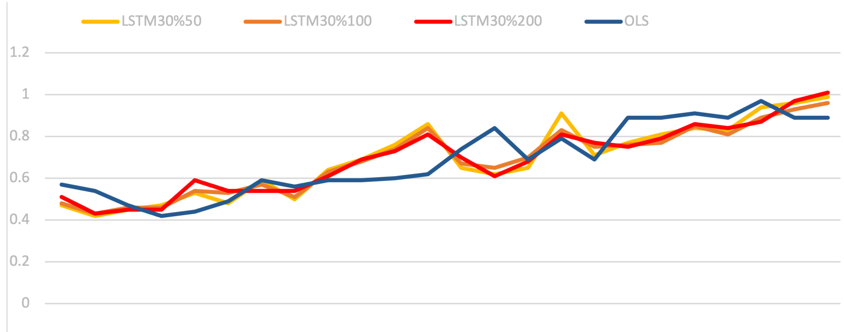


Chart 3. Seasonal optimal rate of risk coverage in three-layer short-term long memory models with 30% training ratio and linear regression model, spring 2017 to winter 2023

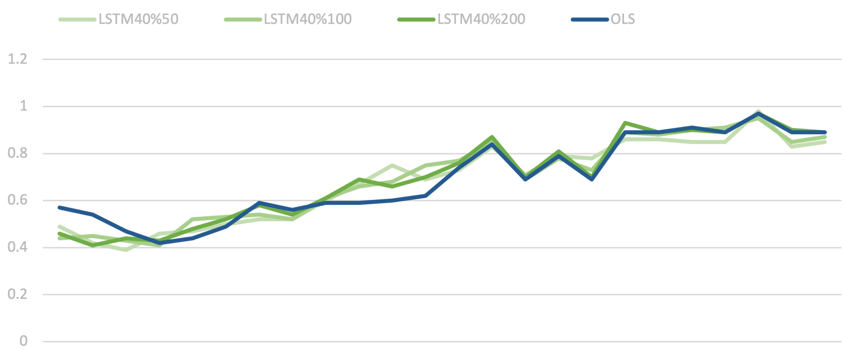


Chart 4. seasonal optimal rate of risk coverage in three-layer short-term long memory models with 40% training ratio and linear regression model, spring 2017 to winter 2023

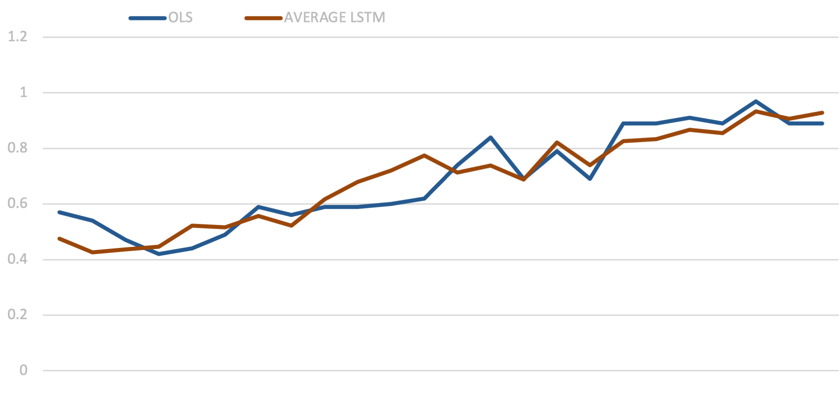


Chart 5. Seasonal optimal average risk coverage rate of three-layer short-term long memory models and linear regression model, spring 2017 to winter 2023

The above tables and graphs show that, firstly, the optimal rate of risk coverage in the six years leading to the winter of 2023 has mainly increased and reaches from around 0.5 in the spring of 2017 to around 0.8 in the winter of 2023. Convergence between the outputs of the neural network with the same training rate but different effective previous periods is observed, and at the same time, more convergence is observed between the outputs of the linear model and the neural network with a training rate of 40% than with the neural network with a training rate of 30%. The outputs of the neural network models are close to each other at the beginning, but eventually divergences occur between them. The two major divergences between the neural network models are related to the divergence between the models with different training ratios in the spring to autumn of 2020 and the summer and autumn of 2021. On the other hand, the convergence of outputs among models with the same training ratio is an evident phenomenon.

One of the trends that can be seen in the graphs above at several points in time is the occurrence of a significant increase in the optimal risk coverage rate of the neural network model with a training ratio of 30% in the spring of 2018 and the winter of 2020 and the occurrence of similar increases with The time interval is 2 to 3 seasons in the outputs of the traditional model and 40% models. Considering the better results of the model with a training ratio of 30% based on the performance evaluation indicators, perhaps this time gap can be attributed to the ability to better adapt and react faster to market changes, which can be the result of better training of the neural network model with a training ratio of 30% .

6 Discussion and analysis

The better performance of the neural network model compared to the traditional linear regression model is in full agreement with the results of foreign research such as Pagnottoni [20], Carverhill et al [6], Barunıkova et al [3] and Fallahpour and Ahmadi [12]. Models based on neural network and artificial intelligence have a higher ability to analyze non-uniform relationships and analyze a greater amount of quantitative and qualitative information compared to traditional models, and their increasing use is the engine driving the fifth wave of the industrial revolution.

Among the recommendations for future studies, it can be mentioned to investigate why the optimal rate of risk coverage increases in the polystyrene market. In the continuation of this path, we can examine whether this result is repeated for other markets as well. Can this trend be attributed to the increase in market risk during the period under review? In this case, what factors have led to an increase in market risk in recent years? These are all questions that can be targets of future research. Also, the difference between the outputs of neural network models and linear regression, along with the difference between the preferences of market participants, raises the question of what factors influence the choice of the appropriate approach by market participants? Considering the limitation of the model implemented in this article in using news as an input variable, it can be recommended to use more comprehensive modeling in the future that is able to understand and analyze textual data. These advanced models can better analyze contextual information related to news and expectations and provide more accurate results, which ultimately help improve forecasts and decisions.

This section can be concluded by recommending petrochemical companies to use the neural network approach to hedge market risk. Considering the recent developments in the field of artificial intelligence and deep learning, using these technologies for purposes such as predicting market behavior and risk management seems necessary and inevitable. This approach has higher power, speed and accuracy than traditional methods and can provide petrochemical companies with more advanced tools for more effective risk analysis and management.

7 Conclusion

The type of contracts and commodity derivatives used in the Iran Commodity Exchange, including the conditions, fees and scope of the covered products, were also investigated, which indicates that the history of using commodity derivatives in the Iranian petrochemical market is based on the use of seasonal forward contracts for products including polyethylene. Heavy, polystyrene, heavy polyethylene, PVC, urea and urea granules are returned. In the following, appropriate modeling was done to cover the risk in the context of this market. The result was the use of a three-layer long-short-term memory model that performed better than the linear regression model in all final versions. Also, models with a 30% training ratio have relatively better performance than models with a 40% training ratio. The investigation of the polystyrene market shows that the optimal rate of seasonal risk coverage during the period of six years from the spring of 2017 to the winter of 2023 has been increasing. It also seems that the neural network model with a training ratio of 30% compared to other models has the ability to adapt faster in response to market changes, which can be due to its better training.

References

[1] Z. Ansari Ardali, M. H. Musavi and H. Kordbacheh, *Estimating the optimal rate of risk hedge and the difference that optimal hedging makes in the natural gas market*, **53** (2017), no. 53, 35-60.

[2] N. Behradmehr, *Crude oil price prediction using wavelet smoothing and artificial neural network*, Energy Economy Studies **5** (2008), no. 18, 81-98.

[3] M. Barunıkova and J. Barunık, *Neural networks as a semiparametric option pricing tool*, Bulletin of the Czech Econometric Society **18** (2011), no. 4, 66-83.

[4] J. Bennell and C. Sutcliffe, *Black–Scholes versus artificial neural networks in pricing FTSE 100 options*, Intelligent Systems in Accounting, Finance & Management: International Journal **12** (2004), no. 4, 243-260.

[5] H. Buehler, L. Gonon, J. Teichmann and B. Wood, *Deep hedging*. Quantit. Fin. **19** (2019), no. 8, 1271-1291.

[6] A. P. Carverhill and T. H. Cheuk, *Alternative Neural Network Approach for Option Pricing and Hedging*, Available at SSRN 480562, 2003, University of Hong Kong, School of Business, Hong Kong.

[7] L. De Gennaro Aquino and C. Bernard, *Bounds on multi-asset derivatives via neural networks*, International Journal of Theoretical and Applied Finance **23** (2020), no. 8, 2050050.

[8] G. Dimitroff, D. Roeder and C. P. Fries, *Volatility Model Calibration with Convolutional Neural Networks*, Available at SSRN 3252432, 2018, Ludwig Maximilian University of Munich (LMU), Germany.

[9] Z. A. Dindar and T. Marwala, *Option pricing using a committee of neural networks and optimized networks*, IEEE International Conference on Systems, Man and Cybernetics **1** (2004), no. 2, 434-438.

[10] R. Du Plooy and P. J. Venter, *A comparison of artificial neural networks and bootstrap aggregating ensembles in a modern financial derivative pricing framework*, Journal of Risk and Financial Management **14** (2021), no. 6, 254-268.

[11] M. Ebrahimi, A. Ghanbary, *Hedging of oil revenues by using future contracts: A case study of Iran*, Economics Research **9** (2009), no. 34, 173-204.

[12] S. Fallahpour and E. Ahmadi, *Estimating value at risk of portfolio of oil and gold by copula-GARCH method*, Financial Research Journal **16** (2014), no. 2, 309-326, doi: 10.22059/jfr.2014.50711.

[13] A. Hirsa, T. Karatas amd A. Oskoui, *Supervised deep neural networks (DNNs) for pricing/calibration of vanilla/exotic options under various different processes*, (2019), arXiv preprint arXiv:1902.05810.

[14] B. Horvath, A. Muguruza and M. Tomas, *Deep learning volatility: A deep neural network perspective on pricing and calibration in (rough) volatility models*, Quantitative Finance **21** (2021), no. 1, 11-27.

[15] A. Jalali Naini and M. Kazemi Manesh, *Investigating changes in the optimal rate of risk coverage in the oil market*, Quarterly Journal of Energy Economics Studies **1** (2013), no. 1, 3-27.

[16] S. Kumar, *A review of neural network applications in derivative pricing, hedging and risk management*, Academy of Marketing Studies Journal **27** (2023), no. 3, 1-24.

[17] X. Liang, H. Zhang, J. Xiao and Y. Chen, *Improving option price forecasts with neural networks and support vector regressions*, Neurocomputing **72** (2009), no. 13, 3055-3065, https://doi.org/10.1016/j.neucom.2009.03.015.

[18] T. Mohammadi, A. Taklif and S. Zamani, *Natural gas price forecasting using a combination of wavelet transform and artificial neural network*, Iranian Economic Research Quarterly **22** (2015), no. 7, 1-26.

[19] L. Montesdeoca and M. Niranjan, *Extending the feature set of a data-driven artificial neural network model of pricing financial options*, 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece **2** (2016), no. 4, 1-6, doi: 10.1109/SSCI.2016.7850014.

[20] P. Pagnottoni, *Neural network models for Bitcoin option pricing*, Frontiers in Artificial Intelligence **2** (2019), no. 5. 1-9, doi: 10.3389/frai.2019.00005.

[21] E. Pishbahar and S. Abedi, *Measuring portfolio value at risk: The application of copula approach*, Journal of Financial Engineering and Securities Management **8** (2016), no. 30, 55-73.

[22] L. Qian, J. Zhao and Y. Ma, *Option pricing based on GA-BP neural network*, Procedia Computer Science **199** (2022), no. 10, 1340-1354, https://doi.org/10.1016/j.procs.2022.01.170.

[23] A. Ratku and D. Neumann, *Derivatives of feed-forward neural networks and their application in real-time market risk management*, OR Spectrum **44** (2022), no. 3, 947-965, https://doi.org/10.1007/s00291-022-00672-1.

[24] A. Shabri and R. Samsudin, *Daily crude oil price forecasting using hybridizing wavelet and artificial neural network model*, Hindawi Publishing Corporation Mathematical Problems Engineering **20** (2014), no. 14, 201402, 1-10, https://doi.org/10.1155/2014/201402.

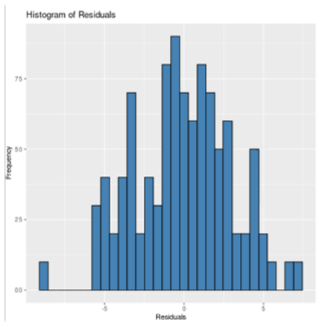
[25] J. Tvedt, *Commodity market flexibility and financial derivatives*, Journal of Commodity Markets **18** (2020), no. 10, 100094.

[26] N. Vadori, *Calibration of derivative pricing models: A multi-agent reinforcement learning perspective*, Proceedings of the Fourth ACM International Conference on AI in Finance **3** (2022), no. 8, 253-260, https://doi.org/10.1145/3604237.3626837.

[27] Y. Yang, Y. Zheng and T. Hospedales, *Gated neural networks for option pricing: Rationality by design*, In Proceedings of the AAAI Conference on Artificial Intelligence **31** (2017), no. 1, 103-120.

[28] F. Zhang, H. Fleyeh and C. Bales, *A hybrid model based on bidirectional long short-term memory neural network and Catboost for short-term electricity spot price forecasting*, Journal of the Operational Research Society **73** (2022), no. 2, 301-325.

Appendix 1



Statistical distribution diagram of disturbance components of the regression model

Appendix 2

```
[('Lagrange multiplier statistic', 5.633),
 ('p-value', 0.426),
 ('f-value', 1.563),
 ('f p-value', 0.519)]
```

The results of Broush Pagan's test to check the homogeneity of variance in the regression model for calculating the optimal rate of risk coverage