

Article

A Hybrid Deep Learning Approach for Crude Oil Price Prediction

Hind Aldabagh ¹, Xianrong Zheng ^{2,*} and Ravi Mukkamala ¹

¹ Computer Science Department, Old Dominion University, Norfolk, VA 23529, USA; haldabag@odu.edu (H.A.); rmukkama@odu.edu (R.M.)

² Information Technology & Decision Sciences Department, Old Dominion University, Norfolk, VA 23529, USA

* Correspondence: x1zheng@odu.edu

Abstract: Crude oil is one of the world's most important commodities. Its price can affect the global economy, as well as the economies of importing and exporting countries. As a result, forecasting the price of crude oil is essential for investors. However, crude oil price tends to fluctuate considerably during significant world events, such as the COVID-19 pandemic and geopolitical conflicts. In this paper, we propose a deep learning model for forecasting the crude oil price of one-step and multi-step ahead. The model extracts important features that impact crude oil prices and uses them to predict future prices. The prediction model combines convolutional neural networks (CNN) with long short-term memory networks (LSTM). We compared our one-step CNN–LSTM model with other LSTM models, the CNN model, support vector machine (SVM), and the autoregressive integrated moving average (ARIMA) model. Also, we compared our multi-step CNN–LSTM model with LSTM, CNN, and the time series encoder–decoder model. Extensive experiments were conducted using short-, medium-, and long-term price data of one, five, and ten years, respectively. In terms of accuracy, the proposed model outperformed existing models in both one-step and multi-step predictions.

Keywords: crude oil price prediction; hybrid deep learning; convolution neural networks; long short-term memory networks



Citation: Aldabagh, Hind, Xianrong Zheng, and Ravi Mukkamala. 2023. A Hybrid Deep Learning Approach for Crude Oil Price Prediction. *Journal of Risk and Financial Management* 16: 503. <https://doi.org/10.3390/jrfm16120503>

Academic Editor: Thanasis Stengos

Received: 31 August 2023

Revised: 24 November 2023

Accepted: 24 November 2023

Published: 6 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Forecasting crude oil price is important for many stakeholders, such as governments, companies, and investors. Crude oil is one of the most influential commodities on the global stage, exerting a profound impact on economies, industries, and financial markets worldwide. Its dynamic pricing is subject to complex interactions of geopolitical events, supply and demand dynamics, economic fluctuations, and environmental factors. As such, the ability to anticipate changes in crude oil prices is pivotal for informed decision-making by governments, corporations, and investors. This is a challenging task because of crude oil's high volatility (Saltik et al. 2016), making prices susceptible to sudden fluctuations driven by multiple factors. Developing prediction models for crude oil prices has been the focus of some researchers. The models include traditional econometric/statistical models and complex machine learning models (Jahanshahi et al. 2022).

The econometric/statistical models for crude oil price prediction employ techniques, such as linear regression (LR), multiple linear regression (MLR), random walk (RW), autoregression (AR), autoregression moving average (ARMA), autoregressive integrated moving average (ARIMA), and generalized autoregressive conditional heteroskedasticity (GARCH) (Behmiri and Manso 2013). On the other hand, the machine learning models employ techniques such as support vector machines (SVM) (Cervantes et al. 2020), artificial neural networks (ANN) (Abiodun et al. 2018), convolutional neural networks (CNN) (Krichen 2023), deep belief networks (DBN) (Ghojogh et al. 2021), and recurrent neural networks (RNN) (Ghojogh and Ghodsi 2023).

In this paper, we focus on machine learning models. We propose a hybrid model that combines CNN and LSTM to forecast oil prices. It can make one-step and multi-step oil price predictions. A one-step prediction can forecast the oil price for the next day, while a multi-step prediction can forecast the oil price for the following week. The multi-step prediction is useful in speculating on promising opportunities and minimizing potential risks. For governments, especially those that heavily rely on oil revenues, accurate price forecasts are imperative for fiscal planning. Budgeting, taxation, and public expenditure allocation depend on oil prices. Sound forecasting aids in managing deficits, stabilizing economies, and mitigating potential shocks.

For one-step predictions, the proposed model combines CNN and LSTM models. The CNN model is effective in extracting new features of time series data. The LSTM model is suitable for modeling a long sequence of dependencies. The combined CNN–LSTM model was tested with short-, medium-, and long-term datasets. The results demonstrated the superiority of the proposed model over the existing models. For multi-step predictions, we implemented two models and compared their results. The first model was the vector output model, which is based on LSTM models using multi-step predictions. The second model was an encoder–decoder model, which is also based on LSTM. We tested them on short-, medium-, and long-term datasets. We find that the multi-step CNN–LSTM model is superior to the encoder–decoder LSTM model.

The paper makes three contributions: First, it proposes a hybrid one-step CNN–LSTM model. Second, it extends the one-step model, and proposes a multi-step model. Third, it conducts comprehensive experiments to show its effectiveness. In particular, it compares the hybrid models with various machine learning and ARIMA models on short-, medium-, and long-term datasets.

The rest of the paper is organized as follows. Section 2 reviews the existing methods used for oil price prediction. Section 3 presents a hybrid deep learning model. Section 4 describes the datasets that we used, the evaluation metrics, and the results of our experiments. Finally, Section 5 summarizes our work, states the advantages and limitations of the proposed method, and discusses some future work.

2. Literature Review

In the literature, some researchers use statistical/econometric time series models and machine learning models to predict crude oil prices. Random walk is a process describing a path that includes a set of random steps (Xia et al. 2020). Among statistical models, random walk-based methods have been adopted for oil price prediction (Panopoulou and Pantelidis 2015). Their main drawback is that they oversimplify the complexity of financial markets (Smith 2023). Econometric time series models are quantitative models that use historical data to predict future prices. Among these models, autoregressive integrated moving average (ARIMA)-based models have been used to predict oil prices (Yu et al. 2016). ARIMA models constitute a family of statistical models that offer a framework for understanding and predicting time series data. The three key components of ARIMA models are the autoregressive (AR) component, which captures the serial correlation of the time series data, the integrated (I) component, which accounts for differencing to achieve stationarity, and the moving average (MA) component, which models short-term dependencies of the data. The limitation of the ARIMA model is its limited capability of capturing the nonlinearity of oil prices.

To overcome the shortcomings of econometric algorithms, various machine learning techniques have been suggested, such as support vector machines (SVM) (Fan et al. 2016) and artificial neural networks (ANN) (Hu 2021). SVM is based on the principle of small-sample statistical learning theory. This theory primarily concerns the analysis of limited datasets in the framework of statistical learning principles, and it has applications in tasks, such as pattern classification and nonlinear regression. The SVM algorithm seeks one misalignment mapping from the input space to the output space. This mapping

transforms the data into a feature space, where subsequent linear regression is performed (Guo et al. 2012).

Artificial neural networks are computational systems inspired by human neural networks. Their primary objective is to generate an output pattern based on a given input pattern. ANN possesses an architecture characterized by a vast number of nodes (neurons) and connections, which are distributed in a parallel fashion (Lakshmanan and Ramasamy 2015). The primary advantage of these algorithms is their ability to handle nonlinearity, which makes them popular in forecasting tasks. The ANN technique is suitable for pattern recognition, and so has become the most popular technique in this field.

Deep learning techniques have been used extensively in economics and finance, due to their capability of learning complex patterns in high-dimensional data. Currently, the most frequently used deep learning techniques are convolutional neural networks (CNN) and recurrent neural networks (RNN), including extensions, such as long short-term memory (LSTM) and deep recursive neural networks (DRNN), etc. Li et al. (2019) presented a novel method based on analyzing and text-mining online media, using a CNN. Similarly, Wu et al. (2021) proposed a text-based and big-data-driven technique that employs a CNN model to automatically read crude oil news updates, processing more than 8000 news headlines. Chen and Huang (2021) devised a CNN to predict stock prices, using gold and oil prices. Using RNN, Wang and Wang (2016) forecasted crude oil indices. Cen and Wang (2019) proposed LSTM-based models to predict the fluctuating behaviors of crude oil prices.

Jahanshahi et al. (2022) employed LSTM and bidirectional LSTM (Bi-LSTM) models to predict crude oil prices affected by the Russia–Ukraine war and the COVID-19 pandemic. They tested the models on a dataset collected over 20 years and used seven features, including crude oil opening, closing, intraday highest, and intraday lowest price values. Similarly, Daneshvar et al. (2022) explored LSTM and Bi-LSTM to predict Brent crude oil prices.

3. A Hybrid Deep Learning Model

Before describing our approach, we first provide the background to convolutional neural network (CNN) and long short-term memory (LSTM) architectures. We then present a brief description of the vector output model and the encoder–decoder LSTM model.

3.1. Convolutional Neural Network

Convolutional neural networks were introduced in 1995 by LeCun and Bengio (1998) in the context of computer vision. CNN mimics the perception and learning processes of the human eye in many tasks, such as image processing, natural language processing, face recognition, classification problems, and recommendation systems. They can be very effective at automatically extracting and learning features from one-dimensional sequence data, such as univariate time series data. They are composed of many layers, i.e., the input layer, the convolutional layers, the pooling layers, the fully connected layers, and the output layer. The role of a convolutional layer is to apply a convolution operation on the data, which involves filtering the input data to measure their effect on the data. The size of the filter indicates its coverage. Each filter utilizes a shared set of weights to perform the convolutional operation. Normally, weights are updated during the training process. The output $v_{i,j}^l$ of an input layer represented by an $N \times N$ matrix and a convolution filter represented by an $F \times F$ matrix is calculated by Equation (1):

$$v_{i,j}^l = \delta \left(\sum_{k=0}^{F-1} \sum_{m=0}^{F-1} w_{k,m} V_{i+k,j+m}^{l-1} \right) \tag{1}$$

where $v_{i,j}^l$ is the value at row i and column j in layer l , $w_{k,m}$ is the weight at row k and column m of the filter, and δ is the activation function. The output of the filter is passed to an activation function of the next layer. Common nonlinear activation functions include the ReLU (Rectified Linear Unit) function, which is represented as $f(x) = \max(0, x)$.

Figure 1 shows the calculation of $v_{1,1}$ in a matrix of size $E \times E$ at layer l , where $E = N - F + 1$. The process performs the convolution of the input data matrix with a convolutional filter.

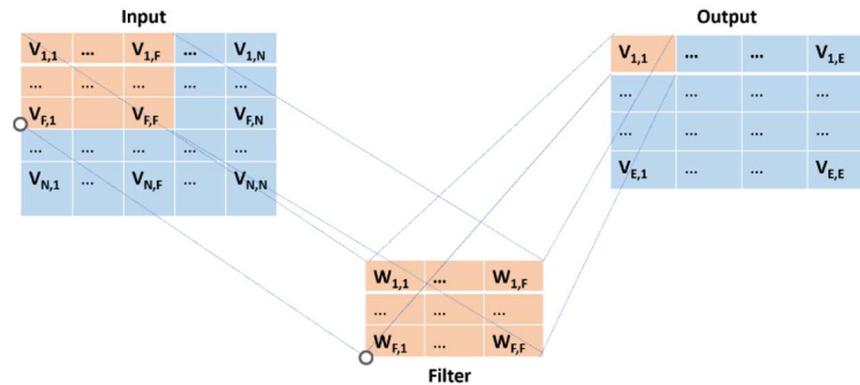


Figure 1. Calculation of the output $v_{1,1}$ by applying a convolution filter $F \times F$ to an input layer represented by the $N \times N$ matrix.

To avoid overfitting in CNN, an additional pooling layer is added. Deep models are more prone to overfitting than shallow models. Max pooling is the most common type of pooling, where the maximum value in a certain window is chosen.

The last step in a CNN is the fully connected layer, which is a multi-layer perceptron (MLP) network. This layer converts the extracted features in the previous layers for the final output. The final output is calculated by Equation (2):

$$v_j^i = \delta \left(\sum_k v_k^{j-1} w_{k,i}^{j-1} \right) \tag{2}$$

where v_j^i is the value of neuron i at the layer j , δ is activation function, and $w_{k,i}^{j-1}$ is the weight of the connection between neuron k from layer $j - 1$ and neuron i from layer j .

3.2. Long Short-Term Memory

LSTM is a special variant of RNN, first introduced by Hochreiter and Schmidhuber (1996). It solves mathematical problems of modeling long sequence dependencies. Figure 2 shows an RNN unit, where X_t denotes the input vector at time t , O_t denotes the output vector at time t , and A_t denotes the hidden state at time t , which is dependent on the input vector and the previous hidden state. U denotes the weights of the hidden layer, V denotes the weights of the output layer, and W denotes the transition weights of the hidden layer. Equations (3) and (4) calculate the output and hidden vectors, respectively, where f is the activation function, which can be sigmoid, tanh, SoftMax, or ReLU.

$$O_t = f(VA_t) \tag{3}$$

$$A_t = f(UX_t + WA_{t-1}) \tag{4}$$

The original, fully connected RNN experiences the gradient vanishing issue in modeling long time series. To solve this problem, LSTM replaces the ordinary node in a hidden layer with a memory cell with a complex internal gate structure. This structure provides a powerful learning capability to LSTM. Because it can extract features automatically and incorporate exogenous variables very easily, LSTM is expected to do well in crude oil price prediction. LSTM overcomes the problem of the gradient vanishing issue of RNN. It is well suited for dealing with long-term dependency problems. The detailed structure of the model is shown in Figure 3, where the cell state C is used to record the long-term status of the sequence and the hidden state h is used for the current status of the sequence. The first

step is the forget gate layer, which decides which information will be discarded from the cell state. This task is accomplished using a sigmoid layer, whose output value is between 0 and 1. The value determines the degree of forgetting the input information, where 0 means completely forgetting and 1 means the opposite. It takes h_{t-1} and x_t as input and outputs a number in the range $[0, 1]$, as shown in the following Equation (5):

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5}$$

where x_t is the input vector of the memory cell at time t , and h_{t-1} is the value of the memory cell at time $t - 1$. W_f and U_f are weight matrices and b_f is a bias vector. The next steps are the input gate layer and tanh layer, which decide which information will be stored in the memory cell state. The input gate layer is a sigmoid layer, and it decides which values will be updated. The tanh layer creates a vector of new candidate values \hat{C} , as shown in Equations (6) and (7):

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{6}$$

$$\hat{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

where W_i , W_c , U_i , and U_c are weight matrices and b_i and b_c are bias vectors. The third step is to update the old cell state C_{t-1} into the new cell state C_t using Equation (8).

$$C_t = i_t \times \hat{C}_t + f_t \times C_{t-1} \tag{8}$$

The final step is generating the output based on a filtered cell state through two stages comprising Equations (9) and (10).

$$O_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{9}$$

$$h_t = O_t \tanh(C_t) \tag{10}$$

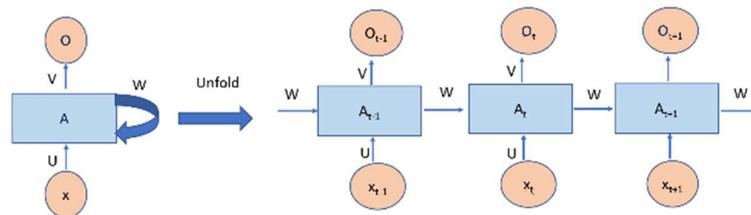


Figure 2. An unrolled recurrent neural network.

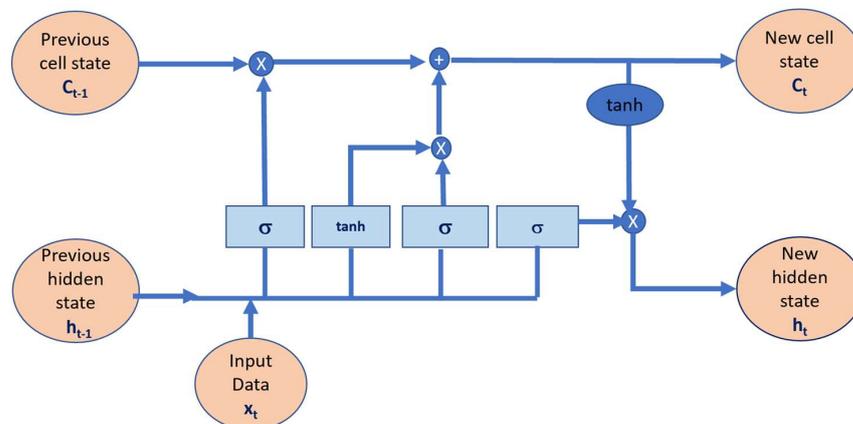


Figure 3. One memory cell of a long short-term memory network.

3.3. The Hybrid Model Architecture

The proposed hybrid model combines a CNN and LSTMs to forecast daily oil prices. The CNN model effectively uncovers and acquires novel features in time series data. On the other hand, the LSTM model excels in capturing extended sequential dependencies. This combined CNN–LSTM model is good at time-based analysis and abstracting meaningful features. Its widespread applications include computer vision and natural language processing with highly satisfactory results (Liang et al. 2020). Our crude oil price prediction model learns a function that maps a sequence of past observations, i.e., past oil prices, as input to an output observation, i.e., the future oil price. As such, the sequence of observations must be transformed into multiple samples, from which the LSTM can learn. We divide the sequence into multiple input/output samples, where three-time steps are used as input and one time step is used as output, for one-step prediction. We experimented with three LSTM models—vanilla LSTM, stacked LSTM, and the proposed hybrid model.

The vanilla LSTM model is composed of only one single hidden layer LSTM unit and an output layer for prediction. The number of LSTM units in the hidden layer is 50. The model is trained using the Adam stochastic gradient descent and optimized using the mean square error loss function. The stacked LSTM is composed of multiple LSTM hidden layers stacked on top of each other. We defined our model with two hidden layers, each with 50 LSTM units.

Our hybrid model is composed of a CNN and an LSTM model, where the CNN is used to interpret sub-sequences of input that together are provided as input to the LSTM. Figure 4 presents the architecture of our hybrid model. We first format our training sample, using samples of an $n \times k$ matrix as input to the convolution layer. We split our time series data into input/output samples with four steps as input and one as output. Each sample can then be split into two sub-samples, each with two-time steps. The CNN can interpret each sub-sequence of two-time steps and provide a time series of interpretations of the sub-sequences to the LSTM model as input.

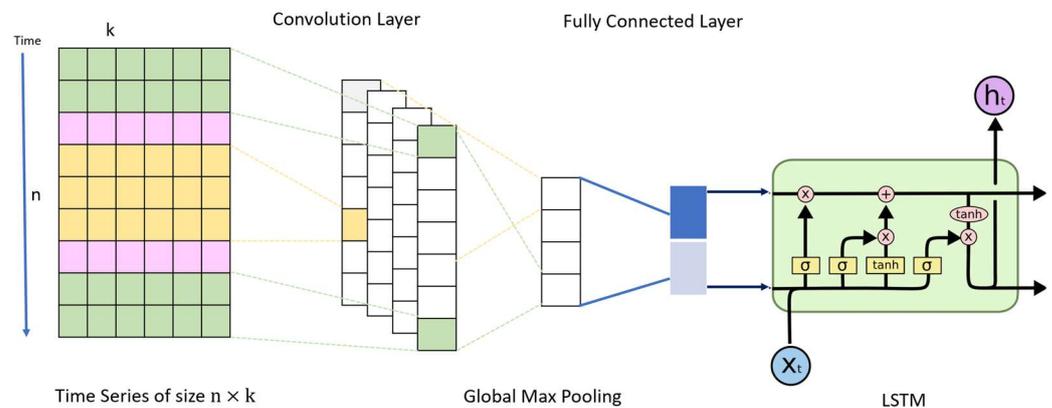


Figure 4. The proposed hybrid model.

All experiments were run on a PC with a 1.8 GHz CPU and 64 GB RAM. The CNN and LSTM were implemented in Python 3.7.4 via Keras 2.4.3. The neural networks were trained using the Nadam algorithm with a default learning rate of 0.001.

3.4. Multi-Step Prediction

A time series forecasting problem that requires a prediction of multiple time steps into the future can be referred to as multi-step time series forecasting. Specifically, it is a problem where the forecast horizon or interval is more than one time step. There are two types of LSTM models that can be used for multi-step forecasting—the vector output model and the encoder–decoder model.

3.4.1. Multi-Step Vector Output LSTM Model

Our multi-step LSTM model predicts a week (i.e., predictions for seven days) into the future. LSTM directly outputs a vector that can be interpreted as a multi-step forecast. We extended our proposed CNN–LSTM model to include multi-step prediction. The input to the CNN–LSTM model is a vector consisting of a series of days, and the output of the model is a vector containing the price prediction for the following seven days. The multiple output strategy entails the construction of a unified model capable of performing one-shot predictions for the entire forecast sequence, as shown in Figure 5, where an input in the range [1, n] days is fed to the CNN–LSTM units, with an expected output of a sequence of n days predictions.

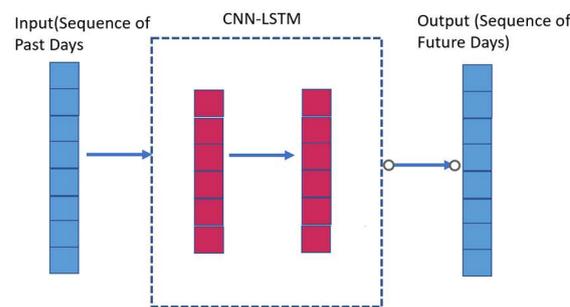


Figure 5. The vector output LSTM model.

3.4.2. Encoder–Decoder LSTM Model

The encoder–decoder LSTM model adopts the autoencoder paradigm (Baldi 2012). It is suitable for addressing the task of multi-step time series forecasting, where both input and output sequences are involved. The problem is commonly referred to as a sequence-to-sequence (seq2seq) problem, and the model is specifically designed to solve problems, such as text translation from one language to another.

The model architecture consists of two distinct sub-models, namely the encoder and the decoder, each playing a crucial role in the overall functioning of the model. The encoder, as its name implies, is responsible for processing and absorbing the input sequence. It uses a vanilla LSTM model as the default choice for the encoder. However, alternative encoder models, including stacked LSTMs, bidirectional LSTMs, and CNN-based models, can be employed based on the specific requirements and characteristics of the input sequence.

The primary objective of the encoder is to generate a fixed-length vector that encapsulates the model’s interpretation of the input sequence. This vector serves as a meaningful representation of the input information and is subsequently utilized by the decoder component to generate the desired output sequence, as shown in Figure 6. The encoder input is the oil price for a sequence of days, and the decoder output is a sequence of predicted prices.

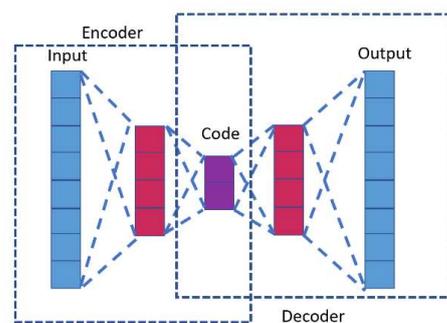


Figure 6. The encoder–decoder LSTM model.

4. Experimental Evaluation and Result Analysis

In this section, we evaluate the performance of our hybrid forecasting model using several evaluation criteria, and compare it with other oil price prediction techniques in the literature.

4.1. Dataset Description

We downloaded three recent crude oil datasets from MarketWatch for ten years (i.e., from 2013 to 2022). The type of oil is WTI crude oil. It represents the benchmarked North America Oil Price. We split the datasets into three sub-datasets. The three sub-datasets are composed of the daily prices of WTI crude oil.

- The first sub-dataset is a long-term period dataset, which spans 10 years from January 2013 to December 2022, constituting 2521 data points. Figure 7 shows the evolution of the time series data. Table 1 provides the descriptive analysis for this sub-dataset.
- The second sub-dataset is a medium-term period dataset, which spans five years from January 2018 to December 2022, constituting 1262 data points. Figure 8 shows the evolution of the time series data. Table 2 provides the descriptive analysis for this sub-dataset.
- The third sub-dataset is a short-term period dataset, which covers only January to December 2022, constituting 251 data points. Figure 9 shows the evolution of the time series data. Table 3 provides the descriptive analysis for this sub-dataset.

Table 1. Statistical properties of crude oil prices (2013–2022).

Mean	Std. Dev.	Skewness	Kurtosis
65.78	22.49	0.44	0.82

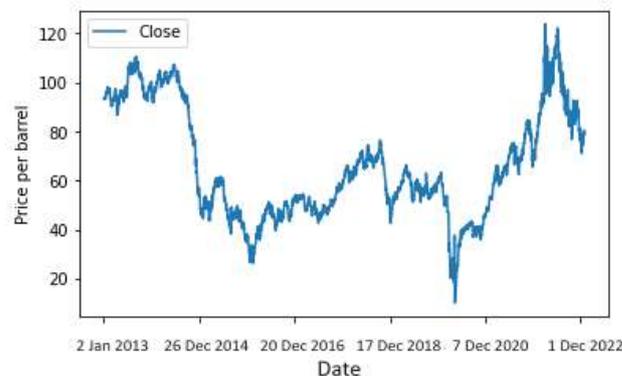


Figure 7. Daily crude oil prices for the long-term period.

Table 2. Statistical properties of crude oil prices (2018–2022).

Mean	Std. Dev.	Skewness	Kurtosis
64.75	19.77	0.39	0.40

Table 3. Statistical properties of crude oil prices (2022).

Mean	Std. Dev.	Skewness	Kurtosis
94.3	12.36	0.36	−0.77

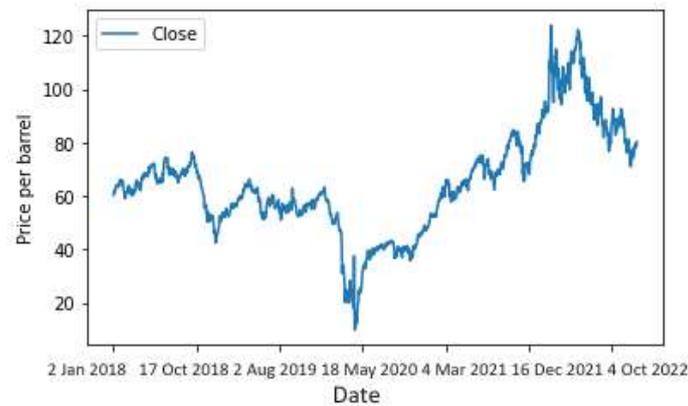


Figure 8. Daily crude oil prices for the medium-term period.

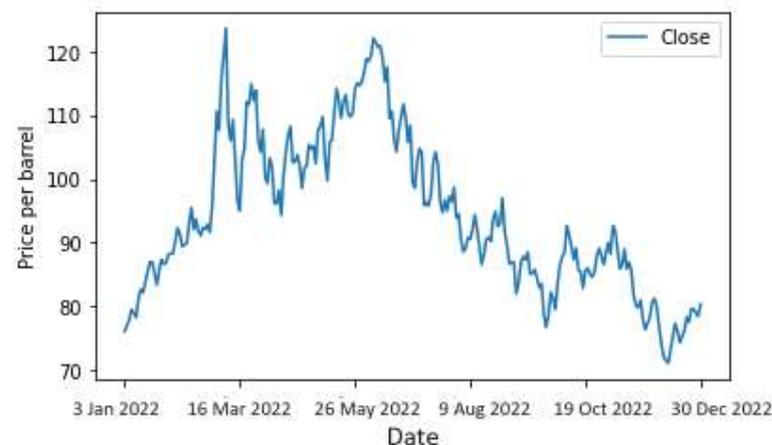


Figure 9. Daily crude oil prices for the short-term period.

Our observations are summarized in the following:

- (1) The mean crude oil prices in the long- and medium- terms were close to USD 65 per barrel, while they increased in the short-term period during 2022, hitting a mean of USD 94. This can be explained by the fact that, during the initial months of 2022, crude oil prices surged to levels surpassing USD 120 per barrel, marking the highest price in the 10-year period. These elevated prices were considered as a potential source of inflationary pressure on economic growth. This scenario stands in contrast to the sharp decline in crude oil prices observed during the Spring of 2020, which was a direct response to the onset of the COVID-19 pandemic.
- (2) The price distribution is not normal, since the skewness is greater than zero and kurtosis is less than three, which yields to skewness towards right with thickened tails.
- (3) Fluctuations in oil prices exhibit diverse magnitudes and durations, implying the possible presence of a dynamic nonlinear nature of the data. This suggests the need for nonlinear models capable of accommodating these irregularities.

Each dataset was split into two parts, with 70% of the dataset used for training and 30% used for testing. Figure 10 shows the training and testing data for the three sub-datasets. The blue color represents the samples for training the model, while the orange color represents the samples for testing the model.

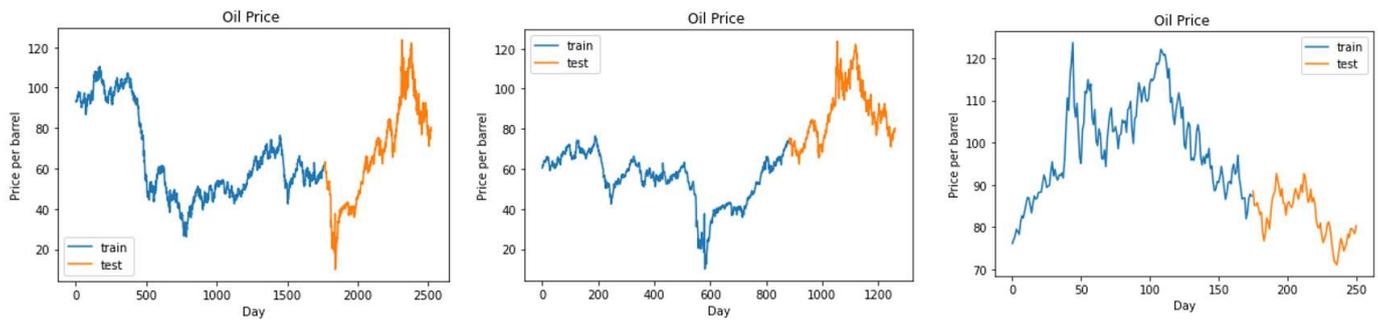


Figure 10. The training and testing data for long-, medium-, and short-term datasets.

4.2. Evaluation Criteria

We used two standard performance metrics to measure the difference between the actual and predicted oil prices. The root mean square error (RMSE) and the mean absolute percentage error (MAPE) have often been used (Zhang 2023). The first metric RMSE quantifies the difference between the actual and the predicted prices. If $y_1, y_2, y_3, \dots, y_n$ are the actual prices and $y'_1, y'_2, y'_3, \dots, y'_n$ are the corresponding predicted prices, then the RMSE is calculated using Equation (11).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - y'_i)^2}{n}} \tag{11}$$

The RMSE is by far the most frequently used metric for measuring the performance of models predicting commodity prices. It is a criterion that gives a higher weight to larger absolute errors.

The second metric MAPE compares the results of different models. The MAPE, when expressed as a percentage, is calculated using Equation (12).

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left(\left| \frac{y_i - y'_i}{y_i} \right| \right) \tag{12}$$

4.3. Parameter Tuning and Optimization

For parameter tuning and optimization, we experimented with different optimizers and activation functions during the training phase. The optimizers that we used included Adam, Adadelta, Adagrad, Adamax, Nadam, Ftrl, and RMSprop optimizers. We tested with ReLU, Softplus, Softsign, tanh, SELU, and ELU activation functions. Table 4 shows the RMSE obtained using each optimizer and activation function. From this table, we can conclude that the best optimizer is Nadam and the best activation function is ReLU.

Table 4. RMSE of various optimizers versus activation functions. The lowest RMSE is highlighted in the bold font.

	Adam	Nadam	Adadelta	Adagrad	Adamax	Ftrl	RMSprop
ELU	2.45	2.42	3.82	3.36	2.39	3.5	3.06
ReLU	2.47	2.36	3.48	3.35	2.5	3.5	3.16
SELU	2.6	2.4	3.54	2.76	2.45	3.27	3.05
tanh	3.10	2.95	36.62	56.77	2.97	53.91	3.06
Softplus	2.51	2.44	3.68	3.10	2.42	3.37	2.55
Softsign	3.11	3.36	40.32	59.91	3.32	61.70	4.05

4.4. Experimental Results

4.4.1. One-Step-Ahead Prediction

To evaluate the effectiveness of our proposed model, we compared it with two other models based on long short-term memory (LSTM). The first model was vanilla LSTM. It has a single layer of LSTM units, and the output layer is used for the oil price prediction. The second model was stacked LSTM, where multiple hidden LSTM layers are stacked with one on top of another.

To verify the efficacy of our proposed model, we compared it with other three benchmark models, namely ARIMA, SVM, and CNN. Our model was a hybrid one composed of a CNN and an LSTM. We used CNN, because it can automatically extract features from one-dimensional sequence data. Tables 5–7 show RMSE and MAPE for the three sub-datasets.

Table 5. RMSE and MAPE of each model on the long-term dataset. The lowest RMSE and MAPE are highlighted in the bold font.

Dataset (2013–2022)	RMSE	MAPE
Vanilla LSTM	2.51	3.0%
Stacked LSTM	2.52	3.0%
CNN–LSTM	2.36	2.7%
SVM	2.87	3.9%
CNN	2.54	2.9%
ARIMA	2.50	2.8%

Table 6. RMSE and MAPE of each model on the medium-term dataset. The lowest RMSE and MAPE are highlighted in the bold font.

Dataset (2018–2022)	RMSE	MAPE
Vanilla LSTM	2.88	2.3%
Stacked LSTM	2.88	2.2%
CNN–LSTM	2.75	2.1%
SVM	19.7	12.9%
CNN	2.82	2.3%
ARIMA	3.06	2.5%

Table 7. RMSE and MAPE of each model on the short-term dataset. The lowest RMSE and MAPE are highlighted in the bold font.

Dataset (2022)	RMSE	MAPE
Vanilla LSTM	2.72	2.7%
Stacked LSTM	2.81	2.8%
CNN–LSTM	2.18	2.2%
SVM	2.58	2.6%
CNN	2.35	2.4%
ARIMA	2.35	2.3%

Table 5 shows the results of long-term predictions. Here, we can observe that vanilla, stacked LSTM, and CNN models yielded similar performances. The proposed CNN–LSTM model improved the performance by obtaining the lowest RMSE. Similar behavior was

observed with MAPE. The SVM and ARIMA models had the lowest accuracy with higher error rates.

Table 6 shows the results of the medium-term prediction. While the medium-term RMSE values were worse than the long-term RMSE values, the MAPE values for the medium term were better. Once again, the proposed CNN–LSTM model yielded better accuracy in predictions. The SVM model had a poor performance compared to the other models when tested on the medium-term datasets. This was because the medium-term dataset is between the years 2018 to 2022 and the testing data were from the last 30% of the time period, which approximately covered 2021 and 2022. The SVM model could not predict the spikes in oil prices during the Russia–Ukraine crisis. Deep learning models tend to have better performances with high volatility rates.

Table 7 shows the results of the short-term prediction. These results fell between the long- and medium-term intervals. From Tables 5–7, we can observe that the hybrid CNN–LSTM model outperformed the other models on the three sub-datasets. The RMSE and MAPE values of the CNN–LSTM model were the lowest among the models.

Figures 11–13 depict the actual oil prices versus the predicted ones on the three sub-datasets, using the hybrid model. In Figure 11, we can observe the almost perfect match between the predicted and actual values, except for the days between 50 and 70. As can be seen from the figure, the abnormal deviation for this time interval corresponded to the high fluctuation in oil prices during the COVID-19 period.

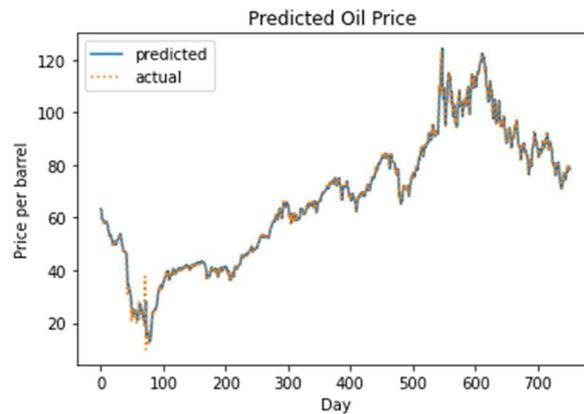


Figure 11. The actual versus the predicted oil price using the hybrid model on the long-term dataset.

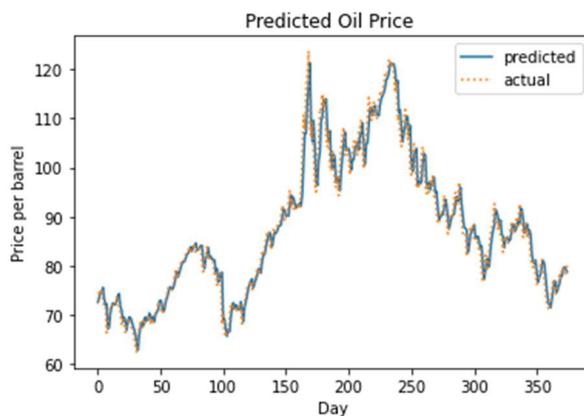


Figure 12. The actual versus the predicted oil price using the hybrid model on the medium-term dataset.

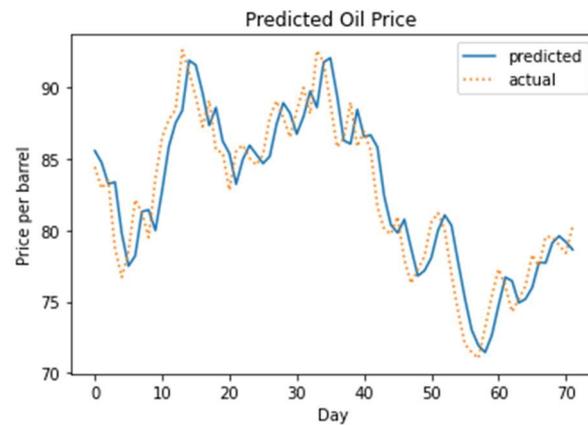


Figure 13. The actual versus the predicted oil price using the hybrid model on the short-term dataset.

In Figure 12, we plotted the actual and predicted oil prices from 2018 to 2022. The approximate range for the training data, which was 70% of this interval, corresponded to the years 2018, 2019, and 2020, leaving two years for the testing data, namely, 2021 and 2022. Our model was able to capture the price changes, even during the recent Russia and Ukraine conflict, with an excellent performance.

Figure 13 shows the actual and predicted values of oil prices. The oil price trend was recognized by our model with minor deviations. As discussed above, it can be concluded that the proposed model performs better than the benchmark models.

Further analysis was conducted to verify the obtained results. We ran a simple moving average (SMA) on the output of machine learning models, including SVM, CNN, LSTM, and CNN–LSTM. Moving averages are considered one of the main indicators in technical analysis. The SMA is the average over a specified period. We calculated a series of averages of fixed-size subsets of the total set of the predicted output. Figure 14a shows the simple moving average of the actual prices versus the predicted prices of SVM, CNN, LSTM, and CNN–LSTM models on the short-term dataset, where the window size is 3 days. Similarly, Figure 14b shows the simple moving average of the actual versus the predicted output of the models on the medium-term dataset, where the window size is 300 days. The SMA of SVM is relatively far from the actual SMA, when compared with other models. It clearly shows that the trend was captured by the models except SVM. In some intervals, CNN and LSTM models were closer to the actual SMA, but the CNN–LSTM model showed a better overall performance.

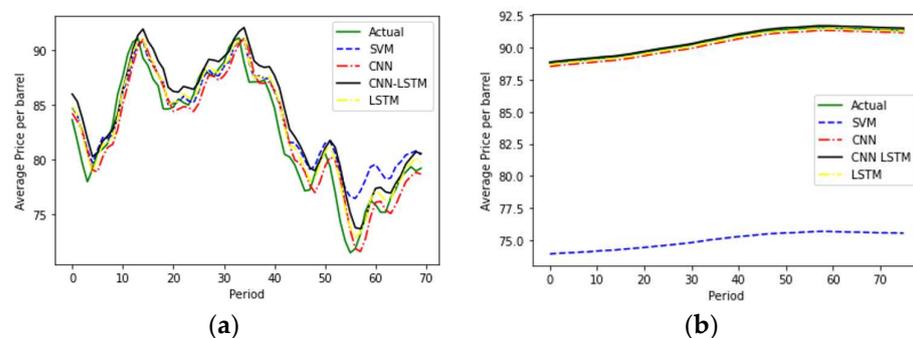


Figure 14. (a) Simple moving average of the actual prices versus the predicted prices on the short-term dataset. (b) Simple moving average of the actual prices versus the predicted prices on medium-term dataset.

Figure 15 shows the simple moving average of the actual versus the predicted output of the models for the long-term dataset, where the window size is 50 days. As the accuracy was high and it was very hard to spot the differences among different models, we enlarged the

first and the last intervals of the SMA. The overall computed SMA is plotted in Figure 15a with two rectangles delineating the enlarged intervals. Figure 15b–d shows zoom plots of the intervals from point 0 to 150 and Figure 15e–g shows zoom plots of the intervals from point 550 to 700. The figures show the superiority of the CNN–LSTM model, where the black line (i.e., the SMA of the CNN–LSTM predicted prices) is closest to the green line (i.e., the SMA of actual prices).

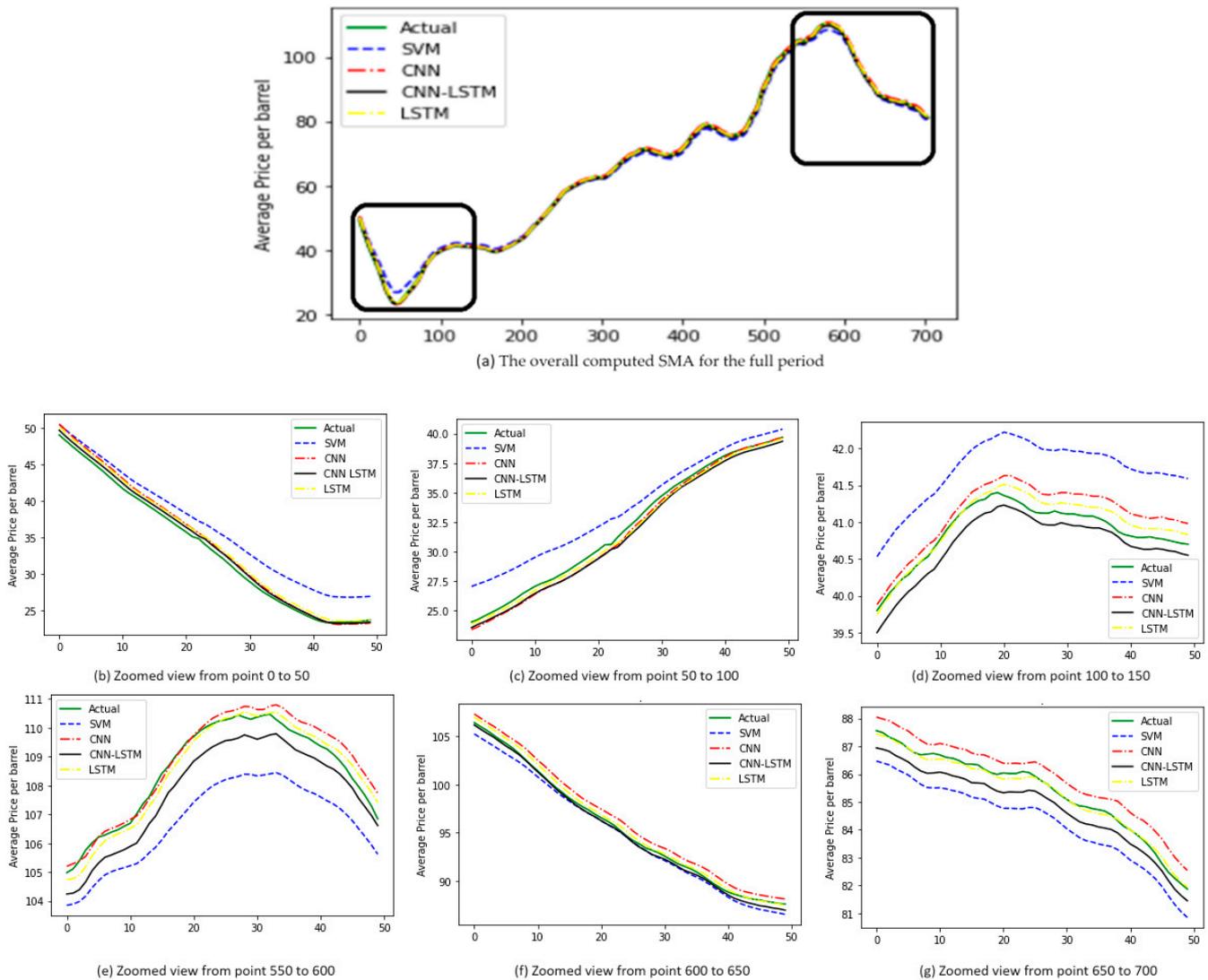


Figure 15. Simple moving average of the actual prices versus the predicted prices on the long-term dataset with an enlarged view of six time-intervals.

4.4.2. Multi-Step-Ahead Prediction

In the experiments, we focused on the comparison among several deep learning models, by calculating their average performance. We conducted several experiments of different multi-step vector output LSTM models. They are stacked LSTM, CNN, and CNN–LSTM vector output models. Tables 8–13 summarize the results, using multi-step vector output and the encoder–decoder LSTM models. The RMSE of $t+1$ to $t+7$ for the long-, medium-, and short-term datasets are illustrated in Tables 8–10. The lowest RMSE is highlighted in the bold font.

Table 8. RMSE of each model on the long-term dataset over seven consecutive days. The lowest RMSE is highlighted in the bold font.

Day	1	2	3	4	5	6	7
CNN	3.65	4.22	4.66	5.04	5.32	5.63	5.83
LSTM	2.79	3.59	4.19	4.52	4.93	5.24	5.49
CNN–LSTM	2.54	3.29	3.89	4.39	4.87	5.21	5.48
Encoder–Decoder	2.48	3.54	5.01	7.00	9.86	13.90	21.08

Table 9. RMSE of each model on the medium-term dataset over seven consecutive days. The lowest RMSE is highlighted in the bold font.

Day	1	2	3	4	5	6	7
CNN	4.02	4.71	5.27	5.81	6.16	6.37	6.58
LSTM	3.15	3.96	4.83	5.48	6.08	6.37	6.86
CNN–LSTM	2.74	3.83	4.58	5.19	5.78	6.21	6.46
Encoder–Decoder	2.74	4.00	4.72	5.43	5.97	6.36	6.61

Table 10. RMSE of each model on the short-term dataset over seven consecutive days. The lowest RMSE is highlighted in the bold font.

Day	1	2	3	4	5	6	7
CNN	3.59	4.13	4.65	5.04	5.29	5.46	5.73
LSTM	4.36	5.21	6.52	6.30	6.81	7.43	7.35
CNN–LSTM	2.60	3.43	4.10	4.64	4.96	5.28	5.49
Encoder–Decoder	2.42	4.04	5.14	6.17	7.11	8.00	8.60

Table 11. MAPE of each model on the long-term dataset over seven consecutive days. The lowest MAPE is highlighted in the bold font.

Day	1	2	3	4	5	6	7
CNN	4.25%	4.94%	5.45%	5.96%	6.53%	6.98%	7.34%
LSTM	3.25%	4.17%	4.84%	5.23%	5.73%	6.15%	6.47%
CNN–LSTM	2.75%	3.66%	4.52%	5.09%	5.60%	6.01%	6.49%
Encoder–Decoder	2.66%	4.08%	5.11%	5.79%	6.37%	6.87%	7.34%

Table 12. MAPE of each model on the medium-term dataset over seven consecutive days. The lowest MAPE is highlighted in the bold font.

Day	1	2	3	4	5	6	7
CNN	3.38%	3.98%	4.38%	4.74%	5.08%	5.43%	5.75%
LSTM	2.56%	3.36%	4.10%	4.62%	5.11%	5.33%	5.96%
CNN–LSTM	2.20%	3.16%	3.79%	4.29%	4.63%	5.06%	5.47%
Encoder–Decoder	2.28%	3.49%	4.11%	4.73%	5.10%	5.48%	5.81%

Table 13. MAPE of each model on the short-term dataset over seven consecutive days. The lowest MAPE is highlighted in the bold font.

Day	1	2	3	4	5	6	7
CNN	3.61%	4.15%	4.67%	5.24%	5.44%	5.61%	5.88%
LSTM	4.13%	4.96%	6.29%	5.98%	6.56%	7.10%	6.96%
CNN–LSTM	2.55%	3.38%	3.88%	4.31%	4.63%	5.07%	5.39%
Encoder–Decoder	2.34%	3.87%	4.76%	5.77%	6.72%	7.63%	8.27%

The MAPE of t+1 to t+7 for the long-, medium-, and short-term datasets are illustrated in Tables 11–13. The lowest MAPE is highlighted in the bold font. Here, we can observe that the performance of the vector output CNN–LSTM model was superior to other models in terms of both RSME and MAPE.

We also compared the performance of the CNN–LSTM and the encoder–decoder model on the t+1 and t+7 days. Figures 16 and 17 show the accuracy of the price prediction for the first day, using the vector output CNN–LSTM model and the encoder–decoder model on the long-term data set, respectively.

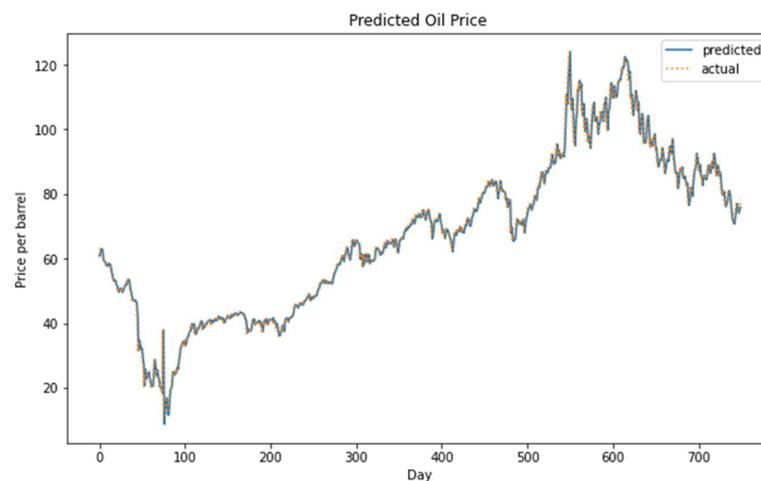


Figure 16. The actual versus the predicted oil price, using the vector output CNN–LSTM model on the long-term dataset for the t+1 day price prediction.

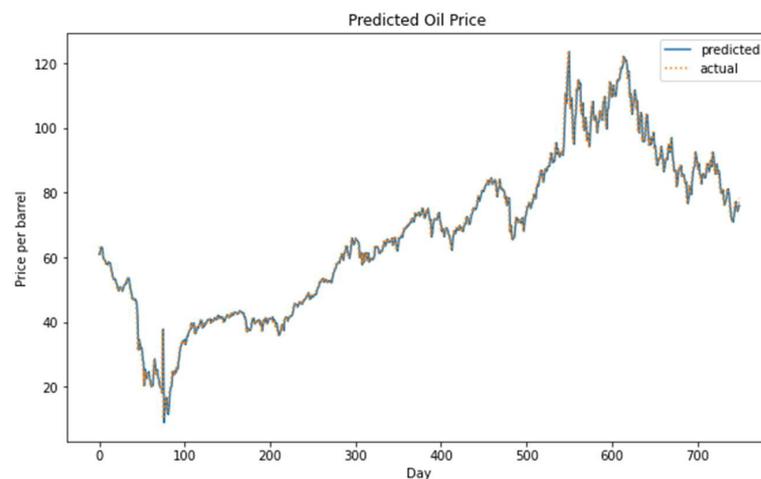


Figure 17. The actual versus the predicted oil price, using the encoder–decoder model on the long-term dataset for the t+1 day price prediction.

Figures 18 and 19 show the accuracy for the seventh day, using the two models on the long-term dataset, respectively. We note that the accuracy for the first day was higher than the seventh, which was expected. Nonetheless, the prediction accuracy for the seventh day remained high and acceptable.

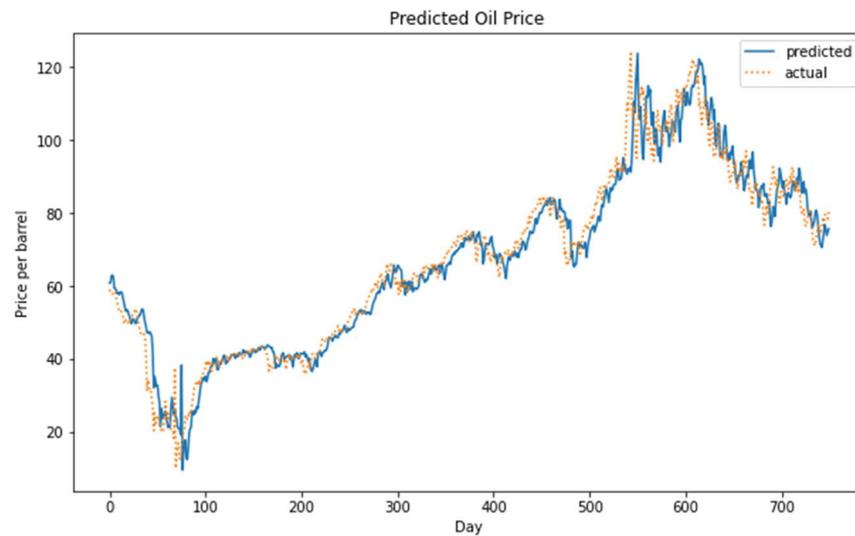


Figure 18. The actual versus the predicted oil price, using the vector output CNN–LSTM model on the long-term dataset for the $t+7$ day price prediction.

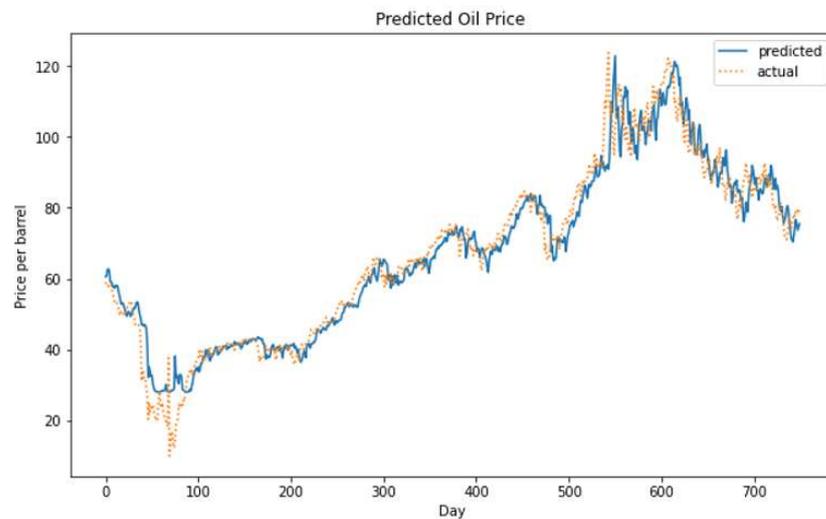


Figure 19. The actual versus the predicted oil price, using the encoder–decoder LSTM model on the long-term dataset for the $t+7$ day price prediction.

When we observed the accuracy of both models, we clearly spotted the superiority of the vector output CNN–LSTM model over the encoder–decoder model for the seventh day. Figures 20 and 21 show the accuracy of the price prediction for the first day, using the vector output CNN–LSTM model and the encoder–decoder model on the medium-term data set, respectively.

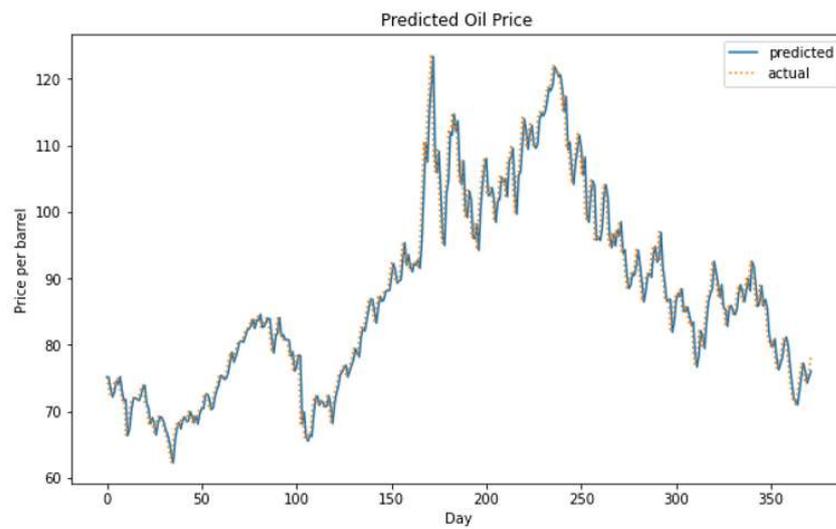


Figure 20. The actual versus the predicted oil price, using the vector output CNN–LSTM model on the medium-term dataset for the $t+1$ day price prediction.

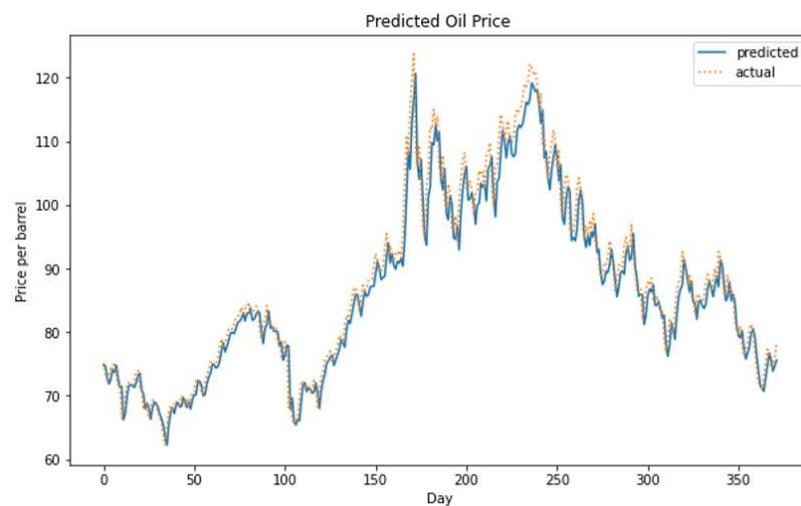


Figure 21. The actual versus the predicted oil price, using the encoder–decoder model on the medium-term dataset for the $t+1$ day price prediction.

Figures 22 and 23 show the accuracy of the price prediction for the seventh day, using the two models on the medium-term dataset, respectively.

Again, the superiority of the vector output CNN–LSTM model over the encoder–decoder model for both the first and seventh days is clear. Figures 24 and 25 show the accuracy of the price prediction for the $t+1$ day, using the vector output CNN–LSTM model and the encoder–decoder model on the short-term dataset, respectively.

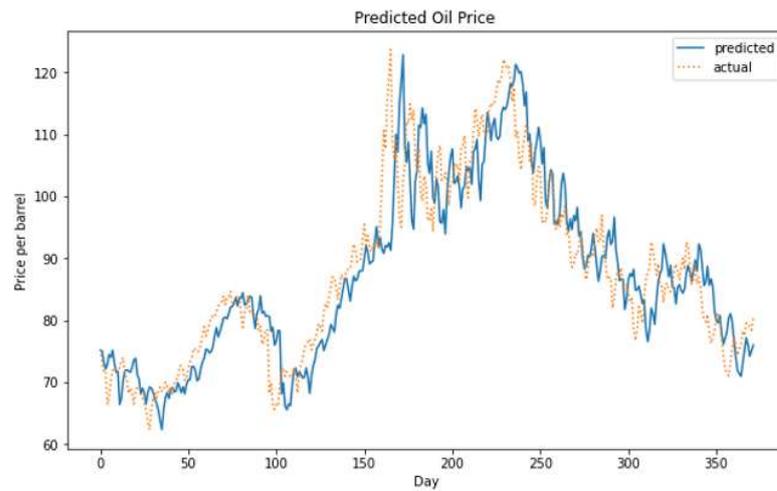


Figure 22. The actual versus the predicted oil price, using the vector output CNN–LSTM model on the medium-term dataset for the t+7 day price prediction.

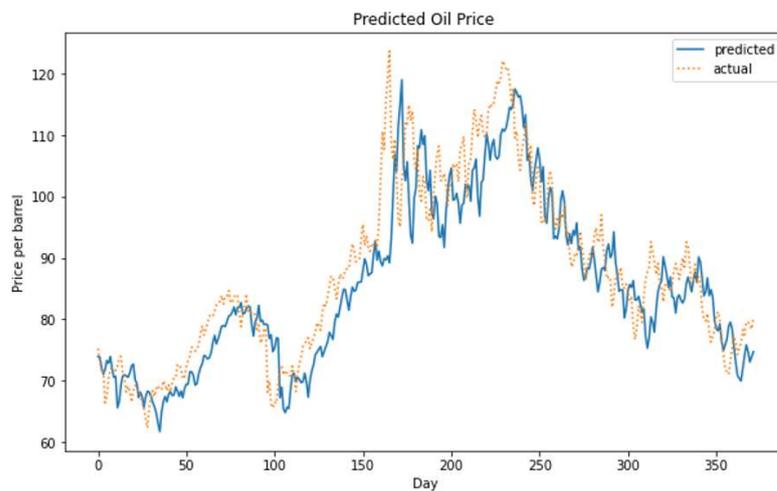


Figure 23. The actual versus the predicted oil price, using the encoder–decoder LSTM model on the medium-term dataset for the t+7 day price prediction.

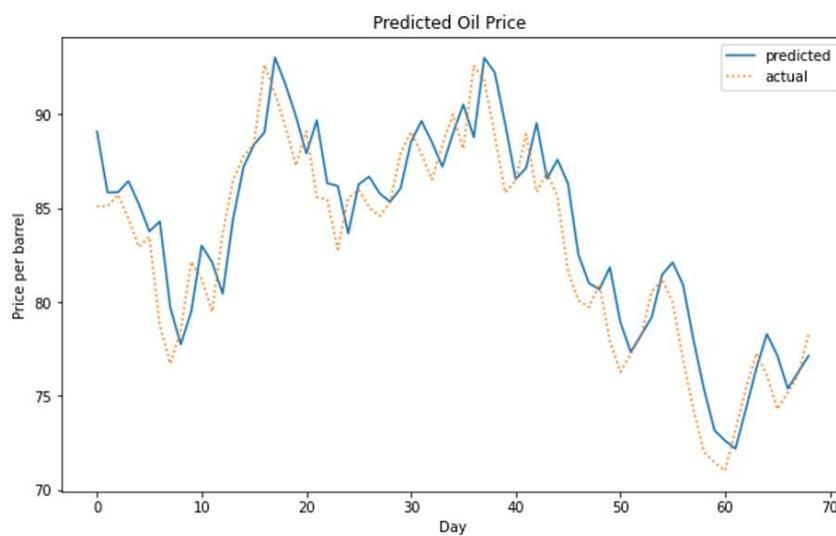


Figure 24. The actual versus the predicted oil price, using the vector output CNN–LSTM model on the short-term dataset for the t+1 day price prediction.

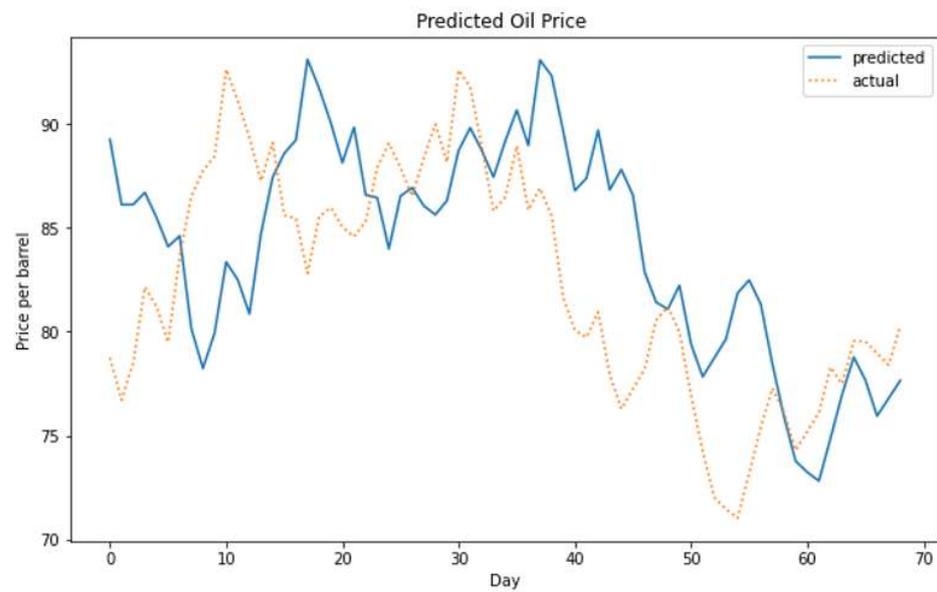


Figure 25. The actual versus the predicted oil price, using the encoder–decoder model on the short-term dataset for the $t+1$ day price prediction.

Figures 26 and 27 show the accuracy of the price prediction for the $t+7$ day, using the vector output CNN–LSTM model and the encoder–decoder model on the short-term dataset, respectively.

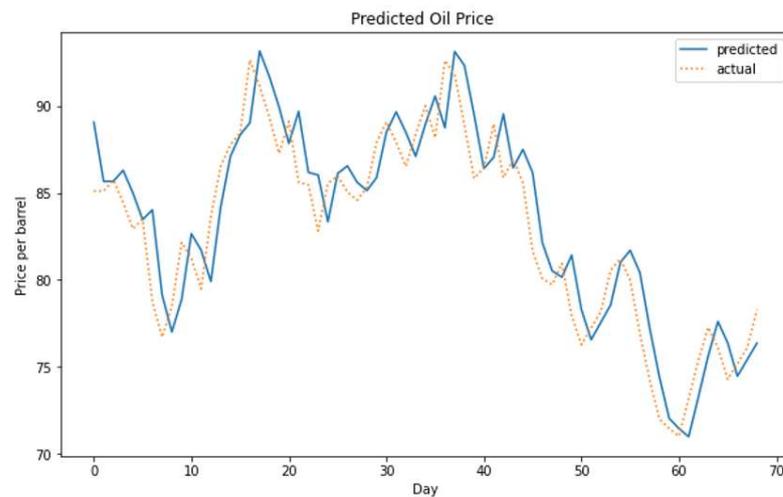


Figure 26. The actual versus the predicted oil price, using the vector output CNN–LSTM model on the short-term dataset for the $t+7$ day price prediction.

Although Figures 24–27 show the close accuracies of the two models, the vector output CNN–LSTM model had higher accuracy than the encoder–decoder model. We can conclude that the proposed vector output CNN–LSTM model yielded higher accuracy than other models in the paper, including the encoder–decoder model, when applied on short-, medium-, and long-term datasets.

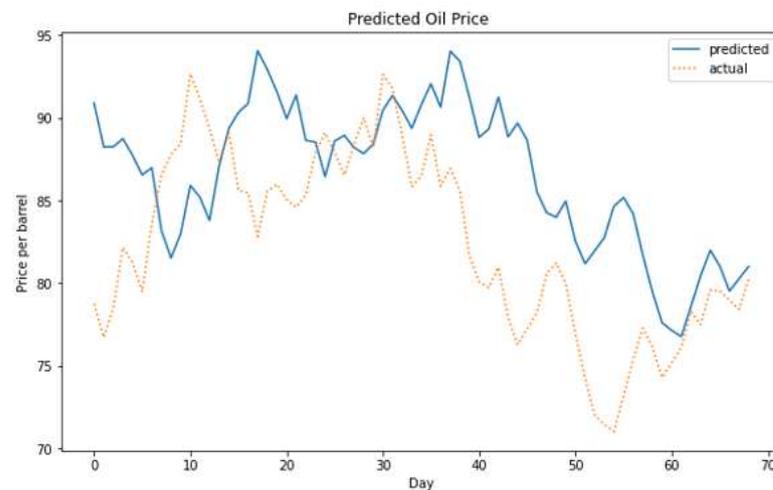


Figure 27. The actual versus the predicted oil price, using the encoder–decoder LSTM model on the short-term dataset for the $t+7$ day price prediction.

5. Conclusions

This paper proposed a model based on a CNN and an LSTM to predict the WTI crude oil price. Due to its high volatility, it is difficult to predict crude oil prices. We used two deep learning models, namely, CNN and LSTM, which are useful in modeling nonlinear dynamics. A hybrid model was presented that combines the two deep learning models. The obtained accuracy for the experimental models was high, but the CNN–LSTM model had a slightly better performance. The models compared in our research are good prediction models in time series cases. However, the CNN–LSTM model had the lowest RMSE and MAPE among the four models. This indicated the effectiveness of the CNN–LSTM model, compared to other models used for the WTI crude oil market.

Publicly available crude oil price data were used to train the models. Our experiments included short-, medium-, and long-term datasets. Investors can use the model trained on long-term samples to develop a long-term investment plan, and they can use the one trained on short-term samples to make a short-term investment decision.

In addition to investigating daily price prediction, we extended our study to include the prediction of multiple steps up to seven days into the future. Our experiments included the vector output CNN–LSTM model and the encoder–decoder LSTM model. We tested the two models on short-, medium-, and long-term datasets. The two models had close accuracies, but the vector output CNN–LSTM model outperforms the encoder–decoder LSTM model.

In our experiments, there are four stages in the model to predict oil prices. In the future, we plan to investigate the impact of the number of stages during model training on the overall model performance. We also aim to combine the proposed model with optimization algorithms, and investigate their effectiveness in increasing prediction accuracy. In addition, the information from different online media sources could be integrated into the system.

Author Contributions: Conceptualization, H.A., X.Z. and R.M.; methodology, H.A., X.Z. and R.M.; software, H.A.; investigation, H.A., X.Z. and R.M.; writing—original draft preparation, H.A.; writing—review and editing, X.Z. and R.M.; supervision, X.Z. and R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data can be downloaded from MarketWatch (<https://www.marketwatch.com/investing/future/cl.1>, accessed on 31 August 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Abiodun, Oludare Isaac, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 4: e00938. [CrossRef] [PubMed]
- Baldi, Pierre. 2012. Autoencoders, unsupervised learning, and deep architectures. *Proceedings of ICML Workshop on Un-supervised and Transfer Learning*, PMLR 27: 37–49.
- Behmiri, Niaz Bashiri, and José Ramos Pires Manso. 2013. Crude Oil Price Forecasting Techniques: A Comprehensive Review of Literature. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2275428 (accessed on 23 November 2023).
- Cen, Zhongpei, and Jun Wang. 2019. Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer. *Energy* 169: 160–71. [CrossRef]
- Cervantes, Jair, Farid García-Lamont, Lisbeth Rodríguez, and Asdrubal Lopez-Chau. 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing* 408: 189–215. [CrossRef]
- Chen, Yu-Chen, and Wen-Chen Huang. 2021. Constructing a stock-price forecast CNN model with gold and crude oil indicators. *Applied Soft Computing* 112: 107760. [CrossRef]
- Daneshvar, Amir, Maryam Ebrahimi, Fariba Salahi, Maryam Rahmaty, and Mahdi Homayounfa. 2022. Brent crude oil price forecast utilizing deep neural network architectures. *Computational Intelligence and Neuroscience* 2022: 6140796. [CrossRef] [PubMed]
- Fan, Liwei, Sijia Pan, Zimin Li, and Huiping Li. 2016. An ICA-based support vector regression scheme for forecasting crude oil prices. *Technological Forecasting and Social Change* 112: 245–53. [CrossRef]
- Ghojogh, Benyamin, Ali Ghodsi, Fakhri Karray, and Mark Crowley. 2021. Restricted boltzmann machine and deep belief network: Tutorial and survey. *arXiv* arXiv:2107.12521.
- Ghojogh, Benyamin, and Ali Ghodsi. 2023. Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey. Available online: <https://arxiv.org/abs/2304.11461> (accessed on 23 November 2023).
- Guo, Xiaopeng, DaCheng Li, and Anhui Zhang. 2012. Improved support vector machine oil price forecast model based on genetic algorithm optimization parameters. *AASRI Procedia* 1: 525–30. [CrossRef]
- Hochreiter, Sepp, and Jürgen Schmidhuber. 1996. LSTM can solve hard long time lag problems. Paper presented at 9th International Conference on Neural Information Processing Systems, Denver, CO, USA, December 3–5; pp. 473–79.
- Hu, Zhenda. 2021. Crude oil price prediction using CEEMDAN and LSTM-attention with news sentiment index. *Oil and Gas Science and Technology* 76: 28. [CrossRef]
- Jahanshahi, Hadi, Süleyman Uzun, Sezgin Kaçar, Qijia Yao, and Madini O. Alassafi. 2022. Artificial intelligence-based prediction of crude oil prices using multiple features under the effect of Russia–Ukraine war and COVID-19 pandemic. *Mathematics* 10: 4361. [CrossRef]
- Krichen, Moez. 2023. Convolutional neural networks: A survey. *Computers* 12: 151. [CrossRef]
- Lakshmanan, Indhurani, and Subburaj Ramasamy. 2015. An artificial neural-network approach to software reliability growth modeling. *Procedia Computer Science* 57: 695–702. [CrossRef]
- LeCun, Yann, and Yoshua Bengio. 1998. Convolutional networks for images, speech, and time-series. In *The Handbook of Brain Theory and Neural Networks*. Edited by Michael A. Arbib. Cambridge, MA: MIT Press.
- Li, Xuerong, Wei Shang, and Shouyang Wang. 2019. Text-based crude oil price forecasting: A deep learning approach. *International Journal of Forecasting* 35: 1548–60. [CrossRef]
- Liang, Shengbin, Bin Zhu, Yuying Zhang, Suying Cheng, and Jiangyong Jin. 2020. A double channel CNN-LSTM model for text classification. Paper presented at 2020 IEEE 22nd International Conference on High Performance Computing and Communications, IEEE 18th International Conference on Smart City, and IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Cuvu, Fiji, December 14–16; pp. 1316–21.
- Panopoulou, Ekaterini, and Theologos Pantelidis. 2015. Speculative behaviour and oil price predictability. *Economic Modelling* 47: 128–36. [CrossRef]
- Saltik, Omur, Suleyman Degirmen, and Mert Ural. 2016. Volatility modelling in crude oil and natural gas prices. *Procedia Economics and Finance* 38: 476–91. [CrossRef]
- Smith, Tim. 2023. Random Walk Theory: Definition, How It's Used, and Example. *Investopedia*. Available online: <https://www.investopedia.com/terms/r/randomwalktheory.asp> (accessed on 23 November 2023).
- Wang, Jie, and Jun Wang. 2016. Forecasting energy market indices with recurrent neural networks: Case study of crude oil price fluctuations. *Energy* 102: 365–74. [CrossRef]
- Wu, Binrong, Lin Wang, Sheng-Xiang Lv, and Yu-Rong Zeng. 2021. Effective crude oil price forecasting using new text-based and big-data-driven model. *Measurements* 168: 108468. [CrossRef]
- Xia, Feng, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan, and Xiangjie Kong. 2020. Random walks: A review of algorithms and applications. *IEEE Transactions on Emerging Topics in Computational Intelligence* 4: 95–107. [CrossRef]
- Yu, Lean, Wei Dai, and Ling Tang. 2016. A novel decomposition ensemble model with extended extreme learning machine for crude oil price forecasting. *Engineering Applications of Artificial Intelligence* 47: 110–21. [CrossRef]
- Zhang, Junhao. 2023. Crude oil price prediction based on multiple ensemble learning algorithms. *BCP Business & Management* 38: 444–51.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.