

Article

# Petroleum Price Prediction with CNN-LSTM and CNN-GRU Using Skip-Connection

Gun Il Kim and Beakcheol Jang \* 

Graduate School of Information, Yonsei University, Seoul 03722, Republic of Korea

\* Correspondence: bjang@yonsei.ac.kr

**Abstract:** Crude oil plays an important role in the global economy, as it contributes one-third of the energy consumption worldwide. However, despite its importance in policymaking and economic development, forecasting its price is still challenging due to its complexity and irregular price trends. Although a significant amount of research has been conducted to improve forecasting using external factors as well as machine-learning and deep-learning models, only a few studies have used hybrid models to improve prediction accuracy. In this study, we propose a novel hybrid model that captures the finer details and interconnections between multivariate factors to improve the accuracy of petroleum oil price prediction. Our proposed hybrid model integrates a convolutional neural network and a recurrent neural network with skip connections and is trained using petroleum oil prices and external data directly accessible from the official website of South Korea's national oil corporation and the official Yahoo Finance site. We compare the performance of our univariate and multivariate models in terms of the Pearson correlation, mean absolute error, mean squared error, root mean squared error, and R squared ( $R^2$ ) evaluation metrics. Our proposed models exhibited significantly better performance than the existing models based on long short-term memory and gated recurrent units, showing correlations of 0.985 and 0.988, respectively, for 10-day price predictions and obtaining better results for longer prediction periods when compared with other deep-learning models. We validated that our proposed model with skip connections outperforms the benchmark models and showed that the convolutional neural network using gated recurrent units with skip connections is superior to the compared models. The findings suggest that, to some extent, relying on a single source of data is ineffective in predicting long-term changes in oil prices, and thus, to develop a better prediction model based on time-series based data, it is necessary to take a multivariate approach and develop an efficient computational model with skip connections.

**Keywords:** crude oil price; hybrid model; multivariate; petroleum price; skip connection technique; univariate

**MSC:** 68T01



**Citation:** Kim, G.I.; Jang, B. Petroleum Price Prediction with CNN-LSTM and CNN-GRU Using Skip-Connection. *Mathematics* **2023**, *11*, 547. <https://doi.org/10.3390/math11030547>

Academic Editors: Petr Nikitin, Sergey Korchagin and Sergey Grigoriev

Received: 28 December 2022

Revised: 12 January 2023

Accepted: 16 January 2023

Published: 19 January 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

The price of crude oil is predicted more often than the price of any other commodity in the global market economy because it has significant political and economic impacts. In fact, its complex and irregular price fluctuations affect both living costs at an individual level and the supply of manufactured goods at the industrial level [1–3]. Crude oil is a commodity that is widely used globally, and its price contributes over 50% to petroleum prices [4], which suggests that any changes in crude oil prices will greatly impact petroleum prices. Because of the recent COVID-19 global pandemic, we now experience greater uncertainty in predicting crude oil prices. Therefore, it is necessary to develop an improved prediction model that can capture the swift transitions made in daily time-series data so that individuals and industries alike can predict unusual trends. Given the accessibility of the internet through smart devices such as smartphones and laptops, it has become easier to

acquire large amounts of oil price data. The development of an accurate prediction model based on crude oil prices along with the use of big data resources and machine learning has drawn significant attention because of the advantages of such models in capturing complex relationships among data variables [5–7].

Oil prices and stock prices share similar characteristics in terms of how they are historically recorded. Stock price predictions are classified as very short-term, short-term, and medium-term predictions. Very short-term ranges from a single day to a month; short-term can range from one to two months, and medium-term can range from two to six months [8]. Because short-term prediction refers to predicting the daily changes in the stock price over a month, oil price prediction over a few consecutive days has been studied. Thus, research on short-term prediction models has attracted substantial attention from researchers. Many advanced data-driven methods using machine-learning methods [9] and statistical analysis [10] have achieved much progress in improving prediction models.

While these previously proposed methods perform well, achieving high performance using these methods requires a large amount of historical data from the same oil price index to train the models. Furthermore, because of inconsistent and unpredictable events as well as the complex correlations among variables related to oil prices, it is even more challenging to achieve the highest performance. In such cases, conventional machine learning approaches cannot be trained to effectively capture the finer details of quick transitions. Thus, to overcome or solve the lack of historical training data, a better prediction model should be developed.

Many researchers have introduced various methods of improving single models by introducing newly developed optimizers that are replaced by the traditional predictive models being utilized in different engineering applications. Ref. [11] have experimented in predicting the thermal efficiency and water yield based on tubular solar still by optimizing the hybrid artificial neural network model with humpback whale optimizer (HWO) that is based on a meta-heuristic optimization approach. Unlike the traditional artificial neural network model performance, the combination of the ANN-HWO model reveals much better results in enhancing the thermal performance of the solar still. Ref. [12] have studied ways of predicting the mechanical and micro-structural properties of friction stir processed (FSP) aluminium alloy by introducing the hybrid method of the integration between the grey wolf optimizer (GWO) and the deep-learning-based multilayer perceptron model. The aforementioned authors also introduce and experiment with various input variables such as rotational speed, linear processing speed, and number of passes, while the outputs are grain size, aspect ratio, micro-hardness, and ultimate tensile strength. Ref. [13] researchers surveyed the use of various artificial neural network (ANN) models on solar energy (SE) systems, which covers the basics of the multilayer perceptron neural network model, wavelet-neural network (WNN) models, the function of radial basis, and RNN-based models (i.e., Elman Neural Network). The previous authors summarized in detail several applications applied in different solar devices including solar collectors, solar-assisted heat pumps, solar air, and water heaters, photovoltaic/thermal (PV/T) systems, solar stills, solar cookers, and solar dryers. In this sense, there is a trend of introducing and implementing the hybrid model developed by various authors in various real-world applications [14–19].

While many research studies have been explored in developing hybrid-model approaches, there is also a wide range of research that has been conducted on predicting oil prices. A few studies have explored the impact of external factors on oil prices, as these factors help in forecasting the finer details of irregular price movements. For example, ref. [20] demonstrated that using the prior knowledge data transfer variable on oil prices improves prediction accuracy. Refs. [21–23] used social media news and external information about the oil markets and macroeconomic news to predict the weekly trends in West Texas Intermediate (WTI) oil prices using sentiment analysis. Ref. [24] analyzed the correlation between oil and gold prices and concluded that gold prices affect oil prices. Refs. [25–27] used the support vector machine (SVM) model, which is a conventional machine-learning approach, whereas [28] used the autoregressive-integrated moving aver-

age (ARIMA) model to predict Brent oil prices. Using a deep-learning approach, ref. [4] employed a back-propagation neural network model for predicting oil prices. Although many studies based on oil price prediction focus on using the WTI and Brent indices, few studies have focused on predicting the price of petrol. In South Korea, two types of oil are mainly consumed: (1) petroleum oil and (2) diesel oil for vehicles. Because these prices change in response to the changes in crude oil price, it is often beneficial to make accurate price predictions.

However, unexpected or sudden changes in oil prices caused by unusual events can drastically affect the accuracy of the predictions. In such cases, recent studies have explored the use of hybrid models to predict the precise details of changes in crude oil prices. For example, ref. [29] proposed a hybrid model consisting of a genetic algorithm (GA) and neural network (NN) to predict the WTI crude oil price. Ref. [30] proposed AdaBoost-LSTM and AdaBoost-GRU (Gated Recurrent Unit) models to improve the accuracy of crude oil price prediction.

There has been a growth in the use of hybrid models to improve the forecasting of crude oil prices using external data. However, studies using such models have focused on predicting crude oil prices via a univariate single-step approach using conventional deep-learning methods. These methods lack medium- and long-term prediction ability; however, it is more important to forecast oil prices for the next several days to prepare for and mitigate the effects of any sudden changes.

Therefore, in this study, we aim to predict the petrol price in South Korea using the data accessible from the official website of South Korea's national oil corporation [31] and the Yahoo Finance site [32]. We also use influential factors to predict the price with better accuracy and explanation; we compare univariate and multivariate methods. For short-term prediction, we set a prediction period of 10 days and consider a multi-step approach. We propose a novel model, inspired by DenseNet architecture, that employs skip connections to improve the existing convolutional neural network (CNN)-long short-term memory (LSTM) model by capturing the finer details and accuracy of petrol price movements. In our experiments, we evaluated our proposed model and compared it with other models using several evaluation metrics (Pearson correlation, mean absolute error, root mean squared error, and R squared ( $R^2$ )). The results clearly show that our proposed model outperforms the conventional models. The following bullet points represent the main contribution in our study.

## 2. Materials and Methods

### 2.1. Proposed Model

This section is subdivided into five subsections that describe the: (1) multivariate input, (2) CNN-based architecture, (3) LSTM/GRU-based architecture, (4) skip connection technique, and (5) output layer.

### 2.2. Multivariate Input Sequences

Our data are a multivariate time-series sequence and consist of a total of 7 variables: (1) petroleum price (normal), (2) petroleum price (premium), (3) diesel price, (4) WTI crude oil price, (5) Brent oil price, (6) Dubai oil price, and (7) heating oil price. Each row of our data is a vector that represents the sequential data of the variable and varies in length according to the number of past timesteps ( $t$ ) in our input layer. Including more past timesteps in the data increases the sample size; therefore, prediction accuracy is increased because more data are used. We then input our data into the CNN layers.

### 2.3. CNN

CNN models have shown great performance in several image datasets [33]. The CNN architecture consists of two types of layers: (1) the convolutional layers and (2) the pooling layers, which are specifically designed to filter and extract useful features such as the channels, color, and resolution from images.

For every convolutional layer, there is a convolution kernel (filter) that acts as a small window, sliding from left to right sequentially over a sub-region to extract the minimal but crucial aspects of a given image through complicated convolution operations. It is followed by a nonlinear activation function, which is a rectified linear unit (ReLU) and a pooling layer.

In every pooling layer, the CNN subsamples and extracts specific values from the convolved features and reduces the dimensions of the matrix to extract the unique features from the raw input image data. For example, the max-pooling method computes the maximum value from every shift of the window, and this represents the most critical feature characteristics within that kernel. Hence, the pooling layer produces matrices with different dimensions repeatedly processed layer by layer to produce the final pooled output value before the fully connected layer is used. A CNN architecture is appropriate for capturing the robust features for our time-series data while avoiding iterative expansion in the dimensions of the matrix. After the CNN layers, we then process our data using an RNN, which considers the long-term features of the data.

#### 2.4. RNN

Many researchers used the LSTM model [34] to predict sequential data due to its powerful mechanisms for overcoming problems such as the vanishing/exploding gradient problem, which is frequently faced in the RNN approach. In particular, in terms of time-series datasets, several research have shown that, compared to the traditional time-series models such as ARIMA (AutoRegressive Integrated Moving Average), deep learning models such as LSTM are far superior at predicting with better accuracy. This is due to the fact that deep-learning models are capable of identifying structures and patterns in data, such as the non-linearity and complexity in time-series forecasting, whereas the ARIMA lacks such capability. Despite ARIMA's non-stationary and univariate characteristics, in which it is effective when there is a strong trend or seasonality within the data, petroleum oil prices fluctuate in a non-cyclical and nonlinear trend. Not knowing the external factors that shift the petrol oil prices to change can become problematic in assessing accuracy when using the ARIMA method. In comparison, LSTM-based models are capable of preserving and training the features of given data for a much longer period of time. This can become crucial and advantageous when it comes to being able to control the length of time durations, which are difficult to see in the ARIMA models [35–37]. Initially, the LSTM architecture shares some of the characteristics of the RNN architecture but was designed to avoid problems with long-term dependency. In contrast to the RNN design, a vanilla LSTM model has three gates (i.e., input, forget, and output gates) that are controlled by the cell state and the activation function. The following equations represent the steps of the overall process of a vanilla LSTM model:

$$\begin{aligned} \text{forget gate}(f_t) &= \delta_g(W_f * [h_{t-1}, X_t] + b_f) \\ \text{input gate}(i_t) &= \delta_g(W_i * [h_{t-1}, X_t] + b_i) \\ \text{output gate}(o_t) &= \delta_g(W_o * [h_{t-1}, X_t] + b_o) \\ \text{current state}(g_t) &= \tanh(W_c * [h_{t-1}, X_t] + b_c) \\ \text{cell state}(C_t) &= f_t \odot C_{t-1} + i_t \odot g_t \\ \text{hidden state}(h_t) &= o_t \odot \tanh(C_t) \end{aligned}$$

where  $\delta_g$  is the sigmoid function,  $\tanh$  is the hyperbolic tangent activation function,  $W_f$ ,  $W_i$ ,  $W_o$  and  $W_c$  represent the weight matrices of each input gate,  $b_f$ ,  $b_i$ ,  $b_o$ , and  $b_c$  represent the bias matrices of each gate,  $h_{t-1}$  is the information for the t-1 units stored in the memory, and  $x_t$  is the input to the memory cell.

Using the above equations, the model evaluates and updates each weight by determining whether to forget the past value according to the forget gate and the state of the current

cell using the result of the sigmoid function (i.e., “0” for forget and “1” for remember). The model provides the final output value by using the tanh activation function on the sum of the hidden state and output gate. A GRU model [38] uses the same principles as the LSTM model in terms of its mechanisms; however, it uses two gates (i.e., reset and update) that require fewer parameters, resulting in much lower computational cost and higher speed than the LSTM approach. Because the memory gate is not used, the GRU memory and hidden layers consider only data points from the recent past to predict the next target value, which is theoretically a much more efficient method than that of the LSTM model. The following equations express the overall mechanisms of the GRU model:

$$\text{update gate}(Z_t) = \delta(W_z * [h_{t-1}, X_t] + b_z)$$

$$\text{reset gate}(r_t) = \delta(W_r * [h_{t-1}, X_t] + b_r)$$

$$\text{current state}(\tilde{h}_t) = \tanh(W * [r_t, h_{t-1}])$$

$$\text{hidden state}(h_t) = (1 - Z_t) * h_{t-1} + Z_t * \tilde{h}_t$$

The reset gate calculates how much of the previous information in memory should be forgotten when given the new input value. In contrast, the update gate calculates how much of the previous information needs to be stored based on the number of time steps relative to the future time steps.

### 2.5. Skip Connection

The use of skip connections has been shown to solve the degradation in performance of deeper models, which are unable to perform much better than shallow models because the increase in the layer depth in the architecture causes the performance to drop dramatically [39]. Ref. [39] compared 20-layer and 56-layer networks trained on the CIFAR-10 image dataset, reporting that the training error and test error demonstrate the problem of this degradation.

Skip connections are known to solve such problems. There are two methods of adding skip connections: (1) addition and (2) concatenation. The addition method was introduced in the ResNet architecture, in which [39] proposed the core idea of updating the back-propagation through the identity function by adding a matrix consisting of one row of an identity matrix. Through such a method, the gradient can simply be multiplied by one, which preserves the gradient. The following mathematical equation can be solved using its partial derivative (gradient) given a loss function:

$$\begin{aligned} \frac{\partial L}{\partial x} &= \frac{\partial L}{\partial H} \frac{\partial H}{\partial x} \\ &= \frac{\partial L}{\partial H} \left( \frac{\partial F}{\partial x} + 1 \right) \\ &= \frac{\partial L}{\partial H} \frac{\partial F}{\partial x} + \frac{\partial L}{\partial H} \end{aligned}$$

Unlike the ResNet model, a skip connection model based on concatenation retains information learned from the initial input throughout its entire architecture, all the way to its final output layer. This method ensures that the maximum amount of information is passed on to each layer and effectively builds a rich number of feature channels in the last layers. This approach is more compact, and the features are reusable. DenseNet [40] is an example of skip connections based on concatenation. Figure 1 shows an example of feature reusability by concatenating five convolutional layers.

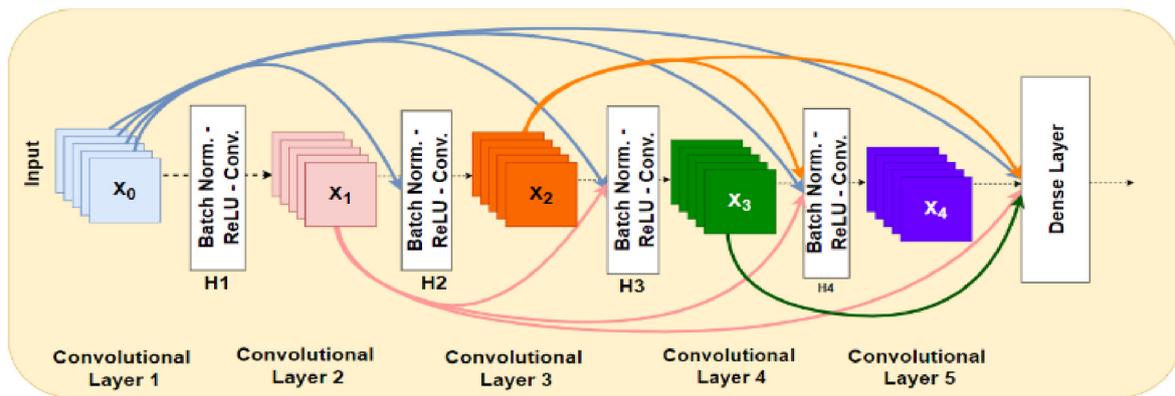


Figure 1. DenseNet architecture consisting of five concatenated convolutional layers.

In this study, we adopt concatenated skip connections. Therefore, in our proposed hybrid models, CNN-LSTM and CNN-GRU, we add concatenated skip connections on top of the hybrid model. At present, skip connections are a standard mechanism implemented in many convolutional architectures. Therefore, similar to the DenseNet architecture, we implement a modified version of the skip connection layer just before the fully connected layers (i.e., the dense layers). Our proposed models are designed as follows: The first proposed model is CNN-LSTM-Skip, which consists of two convolutional 1-dimensional (1D) layers of  $512 \times 512$  filters with a kernel size of 3, and two max-pooling layers of pool size of  $3 \times 2$ , followed by two LSTM layers consisting of 512 units. We then use flatten layers as our last encoder layer and the initial input layer of the dense layers to concatenate the data and produce the skip connection layer. Finally, five dense layers with the leaky-ReLU activation function are applied. The second proposed model is CNN-GRU-Skip, comprising two convolutional layers of  $512 \times 256$  filters with a kernel size of 1 and two max-pooling layers with a pooling size of 3, followed by two GRU layers of 256 units each. We apply the concatenated skip connection layer as the last encoder layer and the initial input layer of the dense layers and employ six dense layers with the leaky-ReLU activation function. In both proposed models, we predict 10 days of petrol prices. The complete proposed model architectures are shown in Figure 2, which shows the steps of how the input data are shifted with respect to time, where each row of cells represents a single day. In this manner, a univariate model would only include a single column of cells that varies with respect to changes over time. In our experiments, we compared univariate and multivariate versions of the various models mentioned previously in our proposed model.

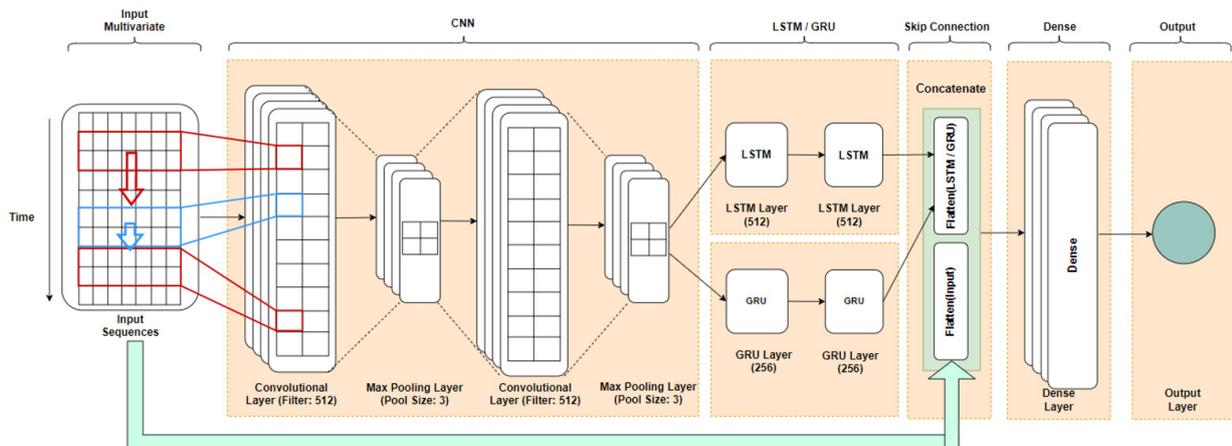


Figure 2. Proposed models (CNN-LSTM-Skip and CNN-GRU-Skip).

## 2.6. Data Preprocessing

In this section, we introduce the dataset we used and explain how its data were processed before the experiment. Our dataset was extracted from the official website of South Korea's national oil corporation [31] and the official Yahoo Finance site [32] using a Python-based application programming interface. The national oil corporation website provides South Korea's daily petrol oil prices, including regular, premium, and diesel prices, while the Yahoo Finance site provides a range of stock market prices for stockholders and investors, including the daily stock prices, exchange traded funds, exchange rates, and oil prices. Among the many types of oil, petroleum oil, which includes premium and diesel oil, are the most-used oils worldwide. In addition, from many lists of oil prices, WTI, Brent, and Dubai, in order, are popular globally. In this study, we also included heating oil because it is one type of petroleum oil and is used in many homes and buildings to produce heat during the winter season. Hence, our dataset consists of 7 columns of variables, and a total of 3623 rows of values. Because our dataset includes a series of dates, we assume that our dataset is based on a daily time series. Since our study is focused on time-series prediction, we adjusted our time period so that all variables in the data have timestamps with a fixed duration between 1 January 2012 and 1 December 2021. There were no missing values in our data. It was necessary to normalize the values of our data to efficiently optimize and train our models. Therefore, we used min–max scaling to convert our values to fall within the range of 0 and 1 for efficient computation.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

In a multi-step prediction, we must consider a sequence of past input values (i.e.,  $X_1, X_2, \dots, X_n$ ), where  $n$  is the number of past data points used to predict the next  $t$  days (i.e.,  $X_{n+1}, X_{n+2}, \dots, X_{n+t}$ ). This method is also known as the window-slicing or forward-moving method; we set our window size to 14 (which we found to be optimal for these data) to predict the next 10 days of petrol prices. Before determining the optimal hyperparameter settings for our model, other settings were tested in which the number of past data points ranged from 7 to 31 days and the number of predicted days ranged from 1 to 21 days. We split our dataset into training, validation, and test datasets using a ratio of 7:1:2, respectively, because validation was also used in our study. The performance of the model on the validation set was used to check the training of the models throughout the experiments. We matched the columns with the target variable ( $y$ ) and the input variables. As our focus is on multivariate multi-step prediction, we must alter our data dimensions so that they fit our models. Thus, we must reshape our datasets so that they can be input to our initial layers, which have three dimensions (i.e., samples, time steps, and variables). In our case, the number of time steps and variables were fixed to 10 and 7, respectively, as we are using up to 14 days of historical data to predict the next 10 days of prices using 7 variables.

Six petrol price forecasting models using LSTM-based structures were analyzed. Specifically, the first two forecasting models were univariate and multivariate versions of the vanilla LSTM model (i.e., LSTM(U) and LSTM(M), respectively). The next two forecasting models were univariate and multivariate versions of a hybrid-model design that combined the CNN and LSTM models (i.e., CNN-LSTM(U) and CNN-LSTM(M), respectively). The last two forecasting models are univariate and multivariate versions of our proposed model with the skip connections applied on top of the hybrid model (i.e., CNN-LSTM-Skip Connection(U) and CNN-LSTM-Skip Connection(M), respectively). Using the same designs and architectures, we also implemented an additional six forecasting models based on GRUs. To evaluate our models, we used four evaluation metrics.

We initially trained for 100 epochs with a dropout ratio of 10% and found that the overfitting problem occurred. We then increased our number of epochs to 500 for each model and removed the dropout technique to improve performance. We also used Adam optimization [41] and a fixed learning rate of  $3 \times 10^{-4}$ . The use of Adam optimization is very

computationally efficient, uses little memory, and is appropriate for non-stationary datasets such as the petrol price data considered in this study. Compared to other optimization functions such as RMSProp [42] and AdaGrad [43], Adam optimization is advantageous in terms of having a bias-correction term that enables it to achieve convergence much better with small steps, which RMSProp optimization lacks. Moreover, although AdaGrad optimization works well for sparse gradients, it is incredibly slow in terms of computation. In this sense, Adam optimization combines the advantages of both RMSProp and AdaGrad, and thus, it is ideal for our sparse and non-stationary dataset of petrol prices. Despite tuning our hyperparameter settings, there were insufficient performance improvements to meet our requirements. Therefore, we increased the number of units in each neural network layer and stacked another layer in the network to improve the performance of our proposed models. All our networks were trained using Google’s TensorFlow 2.6.0 in Python 3.7 using Jupyter Notebook. We used a single GPU (RTX 3090) for faster computation.

2.7. Evaluation Metrics

We used four performance metrics to evaluate our proposed models (i.e., LSTM-Skip and GRU-Skip) with other neural network models and evaluate our performance and errors. These metrics are the mean absolute error (MAE), root mean squared error (RMSE), Pearson correlation, and R squared (R<sup>2</sup>). The MAE and RMSE metrics are widely used. Although these metrics are easy to compute, it is difficult to interpret the effectiveness of our models due to the differences in the units of the values of the time-series variables, as these were normalized before processing [44]. We therefore used the Pearson correlation equation to determine the suitability of our evaluation metrics. The following mathematical equations are used to calculate the first two evaluation metrics:

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

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

where n is the number of observations, y<sub>i</sub> is the i<sup>th</sup> observed value,  $\hat{y}_i$  is the corresponding predicted value, and y is the true value (or ground-truth value). RMSE is simply the root of the mean squared error, and the Pearson correlation is:

$$Pearson\ Correlation = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

where x and y represent the independent and dependent variables, respectively. We calculated how well these sets of data are related to each other. Finally, the coefficient of determination (R<sup>2</sup>) provides us with an overall guideline for tracking the goodness of fit of our proposed model throughout the next n-days of prediction. It is calculated as follows:

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}}$$

$$= 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2}$$

where y<sub>i</sub> is the vector of actual values,  $\hat{y}_i$  is the vector of predicted values, and  $\bar{y}_i$  is the mean value of the y values. Therefore, the sum of squared regression is the residuals obtained from y-points.

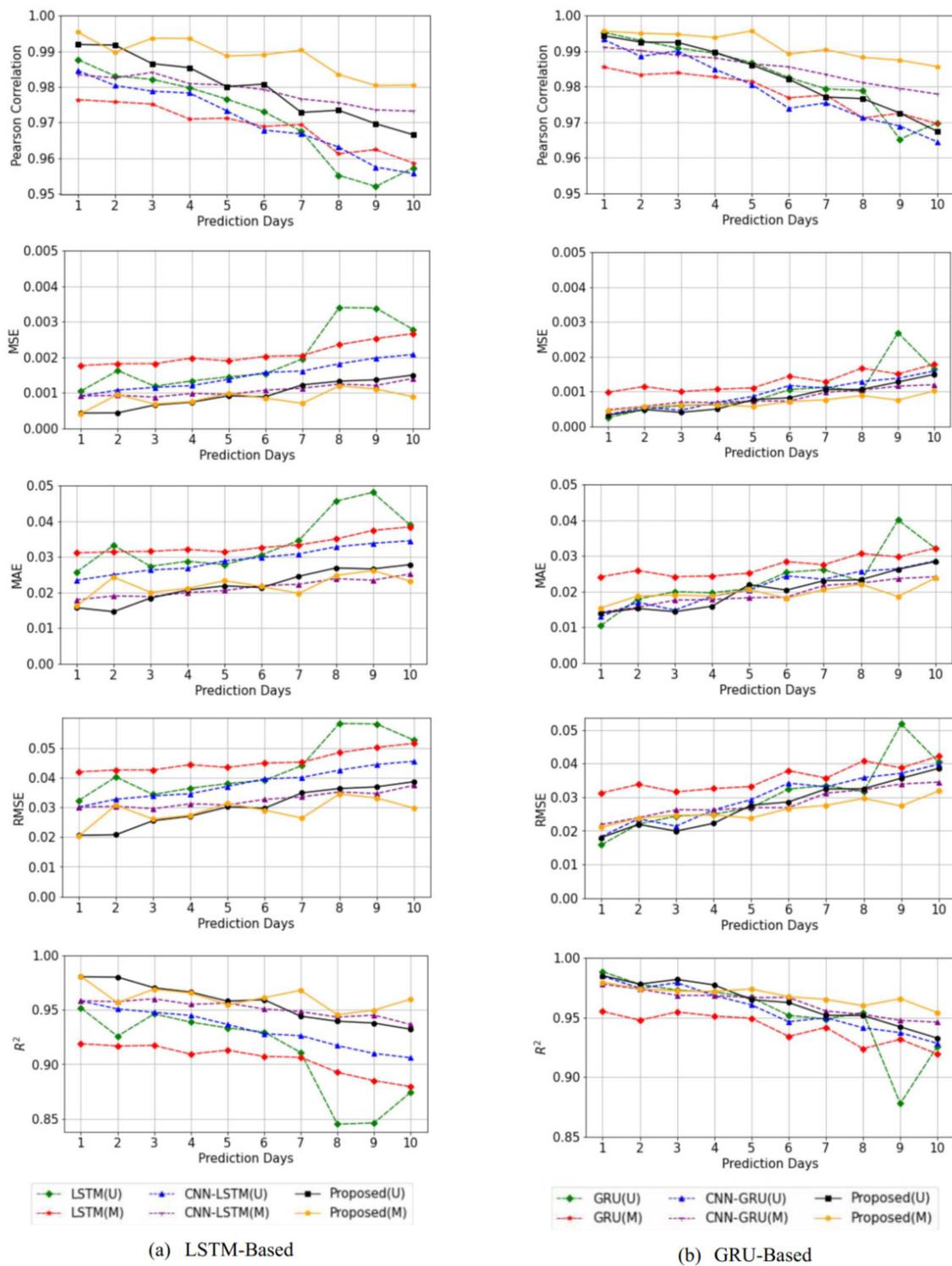
### 3. Results

After tuning our hyperparameter settings, we compared the baseline and hybrid models with our final proposed models. Our proposed models outperformed the baseline models and the hybrid models with respect to all four metrics. We evaluated the performance for the two separate cases of models based on LSTM and GRU. Our setup was evaluated by increasing the prediction period from 1 to 10 days, with the window size fixed to 14 past days. Figure 3 compares results of the baseline, hybrid, and proposed models based on LSTM and GRU. We also demonstrate that a multivariate model is better at predicting the petrol price than a univariate model.

In the case of LSTM, the univariate and multivariate proposed models were compared with the baseline and hybrid models. For 1-day prediction, our multivariate proposed model outperformed other models, obtaining the lowest RMSE and the highest correlation and  $R^2$  scores. A minor difference in the MAE of 0.005 showed that our univariate proposed model performed better than the multivariate proposed model by a small margin. For 2-day to 5-day predictions, the results show that the univariate proposed model performed better than the multivariate proposed model by a small margin. Despite this small difference, the results for the 3-day to 5-day predictions show that the multivariate proposed model maintained the highest Pearson correlation score throughout that period, indicating that our multivariate proposed model maintained its strong performance when compared with other models. Furthermore, throughout the 6-day to 10-day predictions, our multivariate proposed model maintained the highest performance, outperforming other models in almost all four evaluation metrics. There was an approximately 1.8–2.2% difference in the Pearson correlation in the 9-day and 10-day predictions of the univariate and multivariate proposed models. Our multivariate proposed model can maintain its high performance for longer prediction periods than can the other models.

In the case of GRU, the comparison of the baseline, hybrid, and proposed models varied according to whether the model was univariate or multivariate. Despite the 1-day prediction results, almost all predictions for the next 9 days reveal that our proposed model outperformed the baseline and hybrid models. For the 1-day prediction, the univariate baseline model outperformed the other models, yielding the smallest errors and highest  $R^2$  score; the multivariate proposed model obtained its highest correlation score for this prediction period. However, the prediction results for the next 9 days show that our proposed model outperformed the other models. A small difference was observed when comparing the univariate and multivariate proposed models. The 1-day to 5-day prediction shows that the univariate proposed model outperformed the multivariate proposed model by a small margin in all metrics except for the Pearson correlation score.

Our multivariate proposed model yielded the best Pearson correlation scores throughout the prediction period. The results for predictions for 6 days and more reveal that the multivariate proposed model outperformed the other models, including the univariate proposed model. A comparison of the performance results of the proposed models reveals that the univariate proposed models based on both LSTM and GRU maintained the highest overall performance over short-term periods (i.e., 1 to 5 days), whereas the multivariate proposed models obtained the best performance in all four evaluation metrics over medium-term periods (i.e., 6 to 10 days). Table 1 reports the performance of the LSTM-based models, and Table 2 reports the performance of the GRU-based models for 1-day to 10-day predictions. The highest (or lowest) values for each prediction period and each evaluation metric are shown in bold.



**Figure 3.** Comparison of the performance results of the LSTM and GRU models as a function of forecasting period up to 10 days.

**Table 1.** Comparison of the performance results for LSTM-based models for 1-day to 10-day predictions.

Models	1-Day Prediction				2-Day Prediction				4-Day Prediction			
	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>
LSTM (Univariate)	0.0257	0.0323	0.9877	0.9519	0.0332	0.0403	0.9831	0.9258	0.0287	0.0365	0.9798	0.9388
LSTM (Multivariate)	0.0311	0.0420	0.9764	0.9189	0.0314	0.0426	0.9758	0.9169	0.0321	0.0444	0.9710	0.9094
CNN-LSTM (Univariate)	0.0234	0.0301	0.9846	0.9582	0.0250	0.0328	0.9804	0.9509	0.0268	0.0346	0.9783	0.9449
CNN-LSTM (Multivariate)	0.0179	0.0301	0.9834	0.9584	0.0190	0.0304	0.9825	0.9576	0.0199	0.0312	0.9809	0.9552
Proposed Model (Univariate)	0.0157	0.0206	0.9919	0.9805	0.0146	0.0208	0.9917	0.9802	0.0207	0.0270	0.9854	0.9664
Proposed Model (Multivariate)	0.0161	0.0203	0.9954	0.9810	0.0243	0.0308	0.9897	0.9567	0.0211	0.0274	0.9936	0.9656
Models	5-Day Prediction				6-Day Prediction				7-Day Prediction			
	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>
LSTM (Univariate)	0.0278	0.0380	0.9766	0.9335	0.0306	0.0392	0.9730	0.9297	0.0346	0.0441	0.9676	0.9108
LSTM (Multivariate)	0.0314	0.0435	0.9712	0.9130	0.0326	0.0450	0.9689	0.9072	0.0333	0.0452	0.9695	0.9064
CNN-LSTM (Univariate)	0.0288	0.0371	0.9733	0.9367	0.0299	0.0396	0.9679	0.9281	0.0308	0.0401	0.9668	0.9263
CNN-LSTM (Multivariate)	0.0205	0.0309	0.9805	0.9562	0.0219	0.0327	0.9792	0.9509	0.0223	0.0336	0.9766	0.9485
Proposed Model (Univariate)	0.0218	0.0302	0.9801	0.9581	0.0214	0.0298	0.9808	0.9594	0.0245	0.0349	0.9729	0.9442
Proposed Model (Multivariate)	0.0233	0.0313	0.9886	0.9548	0.0217	0.0291	0.9890	0.9612	0.0197	0.0264	0.9903	0.9681
Models	8-Day Prediction				9-Day Prediction				10-Day Prediction			
	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>
LSTM (Univariate)	0.0456	0.0583	0.9552	0.8449	0.0481	0.0581	0.9521	0.8460	0.0390	0.0527	0.9573	0.8742
LSTM (Multivariate)	0.0350	0.0485	0.9613	0.8926	0.0374	0.0503	0.9624	0.8850	0.0384	0.0516	0.9587	0.8795
CNN-LSTM (Univariate)	0.0328	0.0426	0.9631	0.9172	0.0338	0.0445	0.9575	0.9099	0.0345	0.0456	0.9557	0.9060
CNN-LSTM (Multivariate)	0.0238	0.0352	0.9756	0.9434	0.0234	0.0347	0.9736	0.9452	0.0252	0.0374	0.9732	0.9366
Proposed Model (Univariate)	0.0269	0.0364	0.9735	0.9396	0.0266	0.0369	0.9697	0.9379	0.0278	0.0387	0.9666	0.9323
Proposed Model (Multivariate)	0.0247	0.0345	0.9835	0.9456	0.0261	0.0332	0.9874	0.9498	0.0232	0.0298	0.9885	0.9599

**Table 2.** Comparison of the performance results for GRU-based models for 1-day to 10-day predictions.

Models	1-Day Prediction				2-Day Prediction				4-Day Prediction			
	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>
GRU (Univariate)	0.0104	0.0158	0.9953	0.9885	0.0179	0.0220	0.9930	0.9778	0.0196	0.0249	0.9894	0.9715
GRU (Multivariate)	0.0241	0.0313	0.9855	0.9550	0.0259	0.0338	0.9833	0.9477	0.0244	0.0326	0.9827	0.9512
CNN-GRU (Univariate)	0.0129	0.0182	0.9933	0.9847	0.0170	0.0236	0.9885	0.9746	0.0187	0.0262	0.9849	0.9684
CNN-GRU (Multivariate)	0.0142	0.0218	0.9911	0.9781	0.0155	0.0239	0.9901	0.9738	0.0177	0.0262	0.9881	0.9685
Proposed Model (Univariate)	0.0139	0.0180	0.9943	0.9851	0.0152	0.0219	0.9925	0.9780	0.0159	0.0222	0.9897	0.9773
Proposed Model (Multivariate)	0.0153	0.0210	0.9956	0.9796	0.0187	0.0238	0.9950	0.9741	0.0187	0.0247	0.9938	0.9719
Models	5-Day Prediction				6-Day Prediction				7-Day Prediction			
	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>
GRU (Univariate)	0.0208	0.0269	0.9867	0.9667	0.0254	0.0324	0.9825	0.9518	0.0261	0.0335	0.9793	0.9487
GRU (Multivariate)	0.0252	0.0332	0.9815	0.9493	0.0284	0.0379	0.9768	0.9341	0.0275	0.0357	0.9776	0.9416
CNN-GRU (Univariate)	0.0204	0.0293	0.9806	0.9606	0.0244	0.0342	0.9739	0.9464	0.0234	0.0331	0.9754	0.9498
CNN-GRU (Multivariate)	0.0183	0.0269	0.9864	0.9667	0.0184	0.0269	0.9855	0.9668	0.0217	0.0312	0.9834	0.9554
Proposed Model (Univariate)	0.0219	0.0276	0.9861	0.9649	0.0204	0.0286	0.9821	0.9625	0.0230	0.0325	0.9770	0.9518
Proposed Model (Multivariate)	0.0205	0.0238	0.9957	0.9740	0.0181	0.0266	0.9891	0.9675	0.0205	0.0276	0.9904	0.9652
Models	8-Day Prediction				9-Day Prediction				10-Day Prediction			
	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>	MAE	RMSE	CORR	R <sup>2</sup>
GRU (Univariate)	0.0229	0.0317	0.9788	0.9542	0.0401	0.0518	0.9652	0.8777	0.0321	0.0405	0.9698	0.9258
GRU (Multivariate)	0.0307	0.0409	0.9712	0.9238	0.0297	0.0388	0.9725	0.9316	0.0321	0.0422	0.9696	0.9193
CNN-GRU (Univariate)	0.0257	0.0359	0.9712	0.9413	0.0264	0.0371	0.9689	0.9372	0.0286	0.0398	0.9645	0.9282
CNN-GRU (Multivariate)	0.0223	0.0322	0.9811	0.9526	0.0236	0.0339	0.9794	0.9476	0.0241	0.0345	0.9779	0.9460
Proposed Model (Univariate)	0.0233	0.0326	0.9766	0.9516	0.0262	0.0357	0.9725	0.9421	0.0284	0.0386	0.9674	0.9326
Proposed Model (Multivariate)	0.0220	0.0297	0.9882	0.9598	0.0186	0.0274	0.9874	0.9658	0.0237	0.0319	0.9856	0.9540

#### 4. Discussion

First, we compared our proposed models based on LSTM and GRU separately. We also compared univariate and multivariate predictions of the petroleum oil price. Using external factors to predict petroleum oil price improves the performance of our proposed models.

In the case of the LSTM models, we obtained two findings in this study. The first finding is that when comparing the baseline and the hybrid models, the univariate model is more effective than the vanilla LSTM approach. However, when improving the baseline model, external factors play a significant role in the prediction of petroleum prices. In this case, by taking the hybrid model approach and adding a CNN architecture to each

baseline model, we found that the convolutional layers can capture and integrate the convolutional features of the petroleum price and its related variables together better than the vanilla LSTM model, which lacks the ability to combine multiple features into one during processing. As Table 1 shows, the multivariate hybrid model can perform better than the univariate vanilla LSTM, yielding much lower error scores and higher  $R^2$  scores throughout the 10-day prediction period. Our second observation is that although both the univariate and multivariate proposed models outperform the other models, they can also be compared in terms of prediction period. Specifically, over the 10-day period for the predictions, the results show that the univariate proposed model is more accurate than the multivariate proposed model for periods of up to five days. However, for periods of six or more days, the performance of the multivariate proposed model is much better. This emphasizes that although the univariate proposed model can better predict the price for the next couple of days, it cannot maintain its accuracy for long or ensure that it will predict the trend from six days onwards unless some external factors are taken into consideration. In this sense, utilizing external data can guide the prediction and increase accuracy. Table 1 shows that the univariate proposed model has lower error scores for the first predictions up to 5 days, whereas the multivariate proposed model performs better on the 6-day to 10-day period predictions. Using external factors to predict short-term prices may be noisy and unnecessary, but it is beneficial for long-term predictions. Petroleum prices may not quickly and dramatically fluctuate within a few days because it may take several days (or even weeks) for a noticeable change caused by other influencing factors to be observed.

In the case of GRU, we made different observations. The univariate vanilla GRU model obtained the lowest error scores on 1-day predictions of petroleum prices. Interestingly, it outperformed even the proposed LSTM-based model. Theoretically, when comparing between the LSTM and GRU models, when there is less training data, the GRU model can marginally outperform the LSTM model, and it also computes much faster due to having less complexity, unlike the LSTM equation that needs to determine whether to update or not on the previous time-steps. Instead, the GRU model can capture multiple past time-steps at once, though only a few multiple steps at a time, and thus, due to having small sets of data in this study with fewer than 4000 rows of data for each variable, it does not require sophisticated and long-term memory units. Therefore, the GRU model can perform as well as the LSTM, given our set of data, and for this reason, it surpassed the LSTM performance [45]. However, for 2-day to 4-day predictions, the univariate proposed model outperformed it. The GRU model is ideal for a single-step approach. Moreover, comparing Tables 1 and 2, for predictions 5 days and longer, the GRU-based multivariate proposed model outperforms the LSTM-based proposed model, indicating that the GRU-based multivariate proposed model is clearly the best model evaluated in our study, thus proving that external factors are beneficial in long-term prediction.

## 5. Conclusions

Previous studies on crude oil price prediction have focused on various methods such as those for finding external factors that correlate with the crude oil price, conventional machine learning methods, and more recently, hybrid models of ensemble learning algorithms. However, few studies have attempted to improve both methods of predicting petroleum oil prices. Indeed, many researchers have focused on predicting the WTI and Brent crude oil prices. In this study, a novel oil price prediction model using skip connections was proposed as a novel solution for predicting petroleum prices in South Korea instead of using the more standardized oil price index.

Our proposed LSTM- and GRU-based models both showed significantly better performance than the existing models, namely, the vanilla LSTM and GRU models as well as the hybrid models combining the CNN architecture. Furthermore, our proposed models contribute improvements by considering other factors that are important in predicting swift changes in oil prices. We evaluated the model performance using four evaluation metrics to clearly demonstrate that the proposed model outperforms other methods, while

also confirming that the external data we collected indeed provide the finer details needed for predicting swift changes in petrol prices. Because using external data can improve the prediction performance, we plan to introduce other variables such as text-related data like news articles and social media-sourced texts that can provide extensive and contextual information about how oil prices change in a more descriptive manner. We leave this direction of research as our future work.

**Author Contributions:** G.I.K.: Conceptualization, writing original draft, review and editing, data curation, formal analysis, visualization. B.J.: Conceptualization, supervision, funding acquisition. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Yonsei University Research Fund of 2022-22-0115.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: <https://www.opinet.co.kr/user/dopospdrg/dopOsPdrgSelect.do>. It can also be found here: <https://finance.yahoo.com/>.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

ARIMA	AutoRegressive Integrated Moving Average
BPNN	Back-propagation Neural Network
COVID-19	Coronavirus disease 2019
CIFAR-10	Canadian Institute For Advanced Research
CNN	Convolutional Neural Network
GA	Genetic Algorithm
GRU	Gated Recurrent Units
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
NN	Neural Network
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
ReLU	Rectified Linear Units
SVM	Support Vector Machine
WTI	West Texas Intermediate

## References

- Alquist, R.; Kilian, L. *What Do We Learn from the Price of Crude Oil Futures?*; CEPR, International Macroeconomics (Topic): Washington, DC, USA, 2007.
- Kilian, L. Not All Oil Price Shocks are Alike: Disentangling Demand and Supply Shocks in the Crude Oil Market. CEPR Discussion Paper No. 5994, December 2006. Available online: <https://ssrn.com/abstract=975262> (accessed on 2 January 2022).
- Liang, C.; Wei, Y.; Li, X.; Zhang, X.; Zhang, Y. Uncertainty and crude oil market volatility: New evidence. *Appl. Econ.* **2020**, *52*, 2945–2959. [[CrossRef](#)]
- Abdullah, S.N.; Zeng, X. Machine learning approach for crude oil price prediction with Artificial Neural Networks-Quantitative (ANN-Q) model. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 1–8 July 2010.
- Zhao, Y.; Li, J.; Yu, L. A deep learning ensemble approach for crude oil price forecasting. *Energy Econ.* **2017**, *66*, 9–16. [[CrossRef](#)]
- Singleton, K.J. Investor Flows and the 2008 Boom/Bust in Oil Prices. *Manag. Sci.* **2014**, *60*, 300–318. [[CrossRef](#)]
- Baumeister, C.; Guérin, P.; Kilian, L. Do High-Frequency Financial Data Help Forecast Oil Prices? The Midas Touch at Work. *Cap. Mark. Asset Pricing Valuat. EJournal* **2013**, *31*, 238–252. [[CrossRef](#)]
- Kamble, R.A. Short and long term stock trend prediction using decision tree. In Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 15–16 June 2017; pp. 1371–1375.
- Kang, S.H.; Kang, S.; Yoon, S. Forecasting volatility of crude oil markets. *Energy Econ.* **2009**, *31*, 119–125. [[CrossRef](#)]
- Naderi, M.; Khamsehchi, E.; Karimi, B. Novel statistical forecasting models for crude oil price, gas price, and interest rate based on meta-heuristic bat algorithm. *J. Pet. Sci. Eng.* **2019**, *172*, 13–22. [[CrossRef](#)]

11. Moustafa, E.B.; Hammad, A.H.; Elsheikh, A.H. A new optimized artificial neural network model to predict thermal efficiency and water yield of tubular solar still. *Case Stud. Therm. Eng.* **2022**, *30*, 1–14. [[CrossRef](#)]
12. Khoshaim, A.B.; Moustaga, E.B.; Bafakeeh, O.T.; Elsheikh, A.H. An Optimized Multilayer Perceptrons Model Using Grey Wolf Optimizer to Predict Mechanical and Microstructural Properties of Friction Stir Processed Aluminum Alloy Reinforced by Nanoparticles. *Coatings* **2021**, *11*, 1476. [[CrossRef](#)]
13. Elsheikh, A.H.; Sharshir, S.W.; Elaziz, M.A.; Kabeel, A.E.; Guilan, W.; Zhang, H. Modeling of solar energy systems using artificial neural network: A comprehensive review. *Sol. Energy* **2019**, *180*, 622–639. [[CrossRef](#)]
14. Elsheikh, A.; Showaib, E.; Asar, A. Artificial neural network based forward kinematics solution for planar parallel manipulators passing through singular configuration. *IEEE Int. Conf. Robot. Autom.* **2013**, *2*, 2.
15. Elsheikh, A.H.; Shangmugan, S.; Sathyamurthy, R.; Thakur, A.K.; Issa, M.; Panchal, H.; Muthuramalingam, T.; Kumar, R.; Sharifpur, M. Low-cost bilayered structure for improving the performance of solar stills: Performance/cost analysis and water yield prediction using machine learning. *Sustain. Energy Technol. Assessments* **2022**, *49*, 1–142. [[CrossRef](#)]
16. Jani, D.B.; Mishra, M.; Sahoo, P.K. Application of artificial neural network for predicting performance of solid desiccant cooling systems—A review. *Renew. Sustain. Energy Rev.* **2017**, *80*, 352–366. [[CrossRef](#)]
17. Karabacak, K.; Cetin, N. Artificial neural networks for controlling wind–PV power systems: A review. *Renew. Sustain. Energy Rev.* **2014**, *29*, 804–827. [[CrossRef](#)]
18. Kashyap, Y.; Bansal, A.; Sao, A.K. Solar radiation forecasting with multiple parameters neural networks. *Renew. Sustain. Energy Rev.* **2015**, *49*, 825–835. [[CrossRef](#)]
19. Siami-Irdemoosa, E.; Dindarloo, S.R. Prediction of fuel consumption of mining dump trucks: A neural networks approach. *Appl. Energy* **2015**, *151*, 77–84. [[CrossRef](#)]
20. Cen, Z.; Wang, J. Crude oil price prediction model with long short-term memory deep learning based on prior knowledge data transfer. *Energy* **2019**, *169*, 160–171. [[CrossRef](#)]
21. Oussalah, M.C.; Zaidi, A.H. Forecasting Weekly Crude Oil Using Twitter Sentiment of U.S. Foreign Policy and Oil Companies Data. In Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 7–9 July 2018; pp. 201–208.
22. Li, J.; Xu, Z.; Xu, H.; Tang, L.; Yu, L. Forecasting Oil Price Trends with Sentiment of Online News Articles. *Asia Pac. J. Oper. Res.* **2016**, *34*, 1081–1087. [[CrossRef](#)]
23. Sadik, Z.A.; Date, P.; Mitra, G. Forecasting crude oil futures prices using global macroeconomic news sentiment. *IMA J. Manag. Math.* **2020**, *31*, 191–215. [[CrossRef](#)]
24. Beckmann, J.; Czudaj, R.L. Oil and gold price dynamics in a multivariate cointegration framework. *Int. Econ. Econ. Policy* **2013**, *10*, 453–468. [[CrossRef](#)]
25. Xie, W.; Yu, L.; Xu, S.; Wang, S. A New Method for Crude Oil Price Forecasting Based on Support Vector Machines. In *International Conference on Computational Science; Part IV, LNCS 3994*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 444–451.
26. Khashman, A.; Nwulu, N.I. Intelligent prediction of crude oil price using Support Vector Machines. In Proceedings of the 2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMII), Smolenice, Slovakia, 27–29 January 2011; pp. 165–169.
27. Shurong, L.; Yulei, G. Crude Oil Price Prediction Based on a Dynamic Correcting Support Vector Regression Machine. In *Abstract and Applied Analysis*; Hindawi: London, UK, 2013; pp. 1–7.
28. Xiang, Y.; Zhuang, X. Application of ARIMA Model in Short-Term Prediction of International Crude Oil Price. *Adv. Mater. Res.* **2013**, *798–799*, 979–982. [[CrossRef](#)]
29. Chiroma, H.; Abdulkareem, S.; Herawan, T. Evolutionary Neural Network model for West Texas Intermediate crude oil price prediction. *Appl. Energy* **2015**, *142*, 266–273. [[CrossRef](#)]
30. Busari, G.A.; Lim, D.H. Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance. *Comput. Chem. Eng.* **2021**, *155*, 107513. [[CrossRef](#)]
31. Opinet. Available online: <https://www.opinet.co.kr/user/dopospdrg/dopOsPdrgSelect.do> (accessed on 29 December 2021).
32. Yahoo Finance. Available online: <https://finance.yahoo.com/> (accessed on 2 January 2022).
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2012**, *60*, 84–90. [[CrossRef](#)]
34. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
35. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. A Comparison of ARIMA and LSTM in Forecasting Time Series. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018.
36. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM. *arXiv* **2019**, arXiv:1911.09512.
37. Siami-Namini, S.; Namin, A.S. Forecasting Economics and Financial Time Series: ARIMA vs. LSTM. *arXiv* **2018**, arXiv:1803.06386.
38. Chung, J.; Gülçehre, Ç.; Cho, K.; Bengio, Y. Gated Feedback Recurrent Neural Networks. *arXiv* **2015**, arXiv:1502.02367.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

40. Huang, G.; Liu, Z.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
41. Kingma, D.P.; Ba, J.L. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. In Proceedings of the 2015 IEEE Conference on International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
42. Tieleman, T.; Hinton, G. *Lecture 6.5—RMSProp*, COURSERA: *Neural Networks for Machine Learning*; Technical Report; University of Toronto: Toronto, ON, Canada, 2012.
43. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
44. Chatfield, C. Apples, oranges and mean square error. *Int. J. Forecast.* **1988**, *4*, 515–518. [[CrossRef](#)]
45. Yang, S.; Zhou, Y.; Yu, X. LSTM and GRU neural network performance comparison study taking Yelp review dataset as an example. In Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Qingdao, China, 12–14 June 2020; pp. 98–101.

**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.