

## Article

# Deep Learning-Based Methods for Forecasting Brent Crude Oil Return Considering COVID-19 Pandemic Effect

Seyed Mehrzad Asaad Sajadi <sup>1</sup>, Pouya Khodaei <sup>1</sup>, Ehsan Hajizadeh <sup>1,\*</sup>, Sabri Farhadi <sup>1</sup>, Sohaib Dastgoshade <sup>2</sup>  
and Bo Du <sup>3,\*</sup>

<sup>1</sup> Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, Tehran 15914, Iran

<sup>2</sup> Department of Industrial Engineering, Yazd University, Yazd 89195, Iran

<sup>3</sup> SMART Infrastructure Facility, University of Wollongong, Wollongong, NSW 2522, Australia

\* Correspondence: ehsanhajizadeh@aut.ac.ir (E.H.); bdu@uow.edu.au (B.D.)

**Abstract:** Forecasting return and profit is a primary challenge for financial practitioners and an even more critical issue when it comes to forecasting energy market returns. This research attempts to propose an effective method to predict the Brent Crude Oil return, which results in remarkable performance compared with the well-known models in the return prediction. The proposed hybrid model is based on long short-term memory (LSTM) and convolutional neural network (CNN) networks where the autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroscedasticity (GARCH) outputs are used as features, along with return lags, price, and macroeconomic variables to train the models, resulting in significant improvement in the model's performance. According to the obtained results, our proposed model performs better than other models, including artificial neural network (ANN), principal component analysis (PCA)-ANN, LSTM, and CNN. We show the efficiency of our proposed model by testing it with a simple trading strategy, indicating that the cumulative profit obtained from trading with the prediction results of the proposed 2D CNN-LSTM model is higher than those of the other models presented in this research. In the second part of this study, we consider the spread of COVID-19 and its impact on the financial markets to present a precise LSTM model that can reflect the impact of this disease on the Brent Crude Oil return. This paper uses the significance test and correlation measures to show the similarity between the series of Brent Crude Oil during the SARS and the COVID-19 pandemics, after which the data during the SARS period are used along with the data during COVID-19 to train the LSTM. The results demonstrate that the proposed LSTM model, tuned by the SARS data, can better predict the Brent Crude Oil return during the COVID-19 pandemic.

**Keywords:** CNN; COVID-19; deep learning; energy market; LSTM; return prediction



**Citation:** Sajadi, S.M.A.; Khodaei, P.; Hajizadeh, E.; Farhadi, S.; Dastgoshade, S.; Du, B. Deep Learning-Based Methods for Forecasting Brent Crude Oil Return Considering COVID-19 Pandemic Effect. *Energies* **2022**, *15*, 8124. <https://doi.org/10.3390/en15218124>

Academic Editor: Behnam Zakeri

Received: 25 September 2022

Accepted: 27 October 2022

Published: 31 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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

A major challenge in financial markets is modeling and forecasting the market's future. Since the future returns of financial markets influence many economic goals, forecasting the market return is of utmost importance for making the right investment decision. Predicting the financial market's future direction generally requires the examination of several forecasting modules, risk analysis, and trading strategy. One of the major issues to be addressed in financial markets is the trend of oil return and price in the future, which significantly impacts the global economy. Some recent studies have demonstrated the importance of crude oil price or return prediction [1–4]. This paper aims to use deep learning methods to forecast the Brent Crude Oil return accurately. Given that this natural substance has a tremendous impact on the decisions of many large financial institutions such as banks, manufacturing companies, and refining industries, it is necessary to predict the future trend of oil prices or their returns accurately.

To assess financial volatilities, parametric models have been developed with the advent of ARCH (autoregressive conditional heteroscedasticity) and GARCH (generalized ARCH) models presented by [5,6]. Although financial time series have complex and nonlinear structures, linearity correlation structure is presumed in the classical models. Hence, they may not fit nonlinear structured time series, and linearity assumption can affect the model entirely. However, their application as an input to feed models such as ANN can improve their performance [7]. Hajizadeh, E. improved the model's accuracy using the hybridization of GARCH models and neural networks for the prediction of the euro/dollar exchange rate volatility [8]. Since volatility is one of the most influential parameters on the return, using the GARCH models' output as a new input for the return forecasts could help train the model better.

Non-parametric models implemented by various approaches such as machine and deep learning can fit much better with the data set than classical linear models [9,10]. For many years, experts have developed extended endeavors to take advantage of AI (Artificial Intelligence) to set up a system that assists traders with decision-making. Racine, J. showed that neural network-based optimization models, a branch of artificial intelligence, can perform far better and maintain flexibility simultaneously [11]. Hajizadeh, E. applied a hybrid model based on EGARCH (Exponential Generalized Autoregressive Conditional Heteroscedasticity) and ANN to predict the stock index volatility of the S&P 500 [12]. Among other modern deep learning methods, CNN is used mainly for automatic feature selection and market forecasting. Like the ANN, CNN is based on artificial intelligence. Di Persio, L. presented a neural network approach to the stocks' trend forecasting [13]. Furthermore, the results of the CNN-based models were compared with different methods, where CNN showed remarkable performance. Since the long-term prediction of time series data is challenging, long short-term memory (LSTM) is another approach that can be used to maintain long-term dependencies. Wu, C.-H. used the LSTM method to forecast the price of Bitcoin [14]. In a similar study, Karakoyun, E. and A. Cibikdiken compared this method with the ARIMA method in predicting the price of Bitcoin [15].

This research attempts to provide a 2D CNN-LSTM that is based on a hybrid model with LSTM and CNN to predict the future return of Brent Crude Oil, taking advantage of each. We have also tried to illustrate the superiority of the presented models by comparing different up-to-date models described in the coming sections. Because of the lack of data after the outbreak of COVID-19, SARS data are added to the COVID-19 data for better training of the neural networks. It is initially assumed and then proved that the hypothesized COVID-19 behavior would be similar to that of SARS. Therefore, SARS data are added to compensate for the lack of data. Finally, Brent Crude Oil's return is forecasted by applying the LSTM model for the period after the COVID-19 pandemic regarding the concerns raised by [16,17] and other developed studies. Moreover, after accurately forecasting the returns of Brent Crude Oil, we sought to propose a trading strategy on the model's outcome. In the following, we apply some new performance measures such as deflated Sharpe ratio which are used to evaluate the financial aspect of our model.

The remainder of this paper is organized as follows: Section 2 discusses the literature review of the methods used in this research. Section 3 describes the methodology of the models that have led us to present our proposed model. In Section 4, the data and their characteristics contribute to show how our proposed model performs better compared with benchmark models through empirical results. Finally, Section 5 discusses the results and provides future research directions.

## 2. Literature Review

This section first reviews the GARCH-based models such as GARCH, fuzzy-GARCH, and ANN-GARCH that are used to predict the volatility of financial data. Then, it puts forth a review of the up-to-date models in the literature which have been utilized to forecast financial markets' return, price, and volatility.

### 2.1. Classical Models

The first approach, which is based on the data from financial time series with stochastic variance and offers volatility intervals instead of precise forecasts, is the ARCH (Autoregressive Conditional Heteroscedasticity). This approach, first introduced by [5], has been in the literature for a long time. Bollerslev, T. developed a generalized model called GARCH (Generalized ARCH) and was the first who examined ARCH models in financial terms, both empirically and theoretically [18]. Bauwens, L. employed multivariate GARCH models because the financial volatilities move along time with assets and markets, resulting in the failure of univariate GARCH models to perform effectively under these conditions [19]. Lam, K.S. and L.H. Tam showed that classical methods such as GARCH are mean-revert and usually built with close price data, possibly resulting in the negligence of the important daily price changes and subsequently leading to data loss and inefficiency [20]. Hence, the range-based autoregressive model was introduced and expanded to play off these weaknesses. Developing on the issue, Wang, L. presented the efficient semi-parametric GARCH model for financial volatility [21]. Maciel, L. introduced fuzzy-GARCH models to predict and model financial volatility [22]. More recently, Sadik, Z.A. proposed the news-augmented GARCH (NA-GARCH) model to forecast stock price volatility, which is the combination of the GARCH model by examining the effects of quantified news sentiment on the movement of stock prices [23]. Finally, Naimy, V. compared the accuracy of GARCH models in assessing the volatility of cryptocurrencies [24]. Many financial time series data show a nonlinear dependency structure, but a linear correlation structure is generally assumed between time-series data in GARCH models; hence, these models do not usually record the nonlinear patterns. As a result, the approximate linear models obtained from them may not be satisfactory in complex problems.

### 2.2. Artificial Intelligence-Based Models

Artificial intelligence-based models have demonstrated superb performance in modeling and forecasting return, price, and volatility. According to [11], neural network-based optimization models, a type of artificial intelligence, can perform far better than the classic GARCH models and indicate better flexibility. Artificial neural network (ANN) can be much more applicable and flexible when the output of the GARCH and ARCH models feeds the networks to predict the volatility and return of the financial market. Hamid, S.A. developed a neural network-based approach to forecast the future price volatility of the S&P 500 [25]. Pérez-Rodríguez, J.V. examined the ANN and STAR (smooth transition auto-regression) models for the prediction of the Spanish Stock Index [26]. In another study, Wang, Y.-H. proposed a nonlinear neural network method to predict the selected price of a stock index [27]. In the same line, [9] presented meta-modeling neural networks in forecasting financial time series. Bildirici, M. and Ö.Ö. Ersin worked on predicting robust GARCH family models integrated with ANN and then used it to predict Istanbul Stock Exchange's return [28]. Roh, T.H. implemented three aggregated financial time-series methods and artificial neural networks in the KOSPI 200 (Korea Composite Stock Price Index 200) [29]. In their study, ref. [12] applied a hybrid model based on EGARCH-ANN to forecast the stock index volatility of the S&P 500. Similarly, Adhikari, R. and R. Agrawal presented an approach that processed the linear part of the financial data set by the Random Walk model and the lasting nonlinear part through a set of ANN and Elman ANN (EANN) models [30]. Finally, Kristjanpoller, W. and M.C. Minutolo investigated the combination of GARCH and ANN approaches for the prediction of gold price volatility [7]. Mohammed, G.T. proposed an innovative fuzzy-EGARCH-ANN model to forecast the stock market volatility [31]. Their model improved the leverage effects and volatility clustering of highly nonlinear financial data compared with the EGARCH model.

Using a support vector machine (SVM) to forecast the financial time series, ref. [32] discovered that SVM performed better than the backpropagation neural network (BPNN). Tang, L.-B. examined the volatility prediction by wavelet-support vector machine (W-SVM),

which is a hybridization of SVM and discrete wavelet transform (DWT) [33]. In their study, Chen, C.-H. investigated the SVM based on GARCH models to predict volatilities [34]. Zhiqiang, G. developed this model by predicting the financial time series through SVM models and locality preserving projection (LPP) while also considering the particle swarm optimization (PSO) algorithm [35]. Later, ref. [36] improved the PSO algorithm by developing the interval volatility prediction using SVM. As shown, the use of these algorithms increased the SVM capabilities. In a study conducted by [37], a new approach to stock price prediction was introduced based on SVM and singular spectrum analysis (SSA). Then, Lu, C.-J. employed the support vector regression (SVR) method [38]. In a more recent work, Sun, H. and B. Yu designed a two-step volatility prediction method from the combination of the SVR and the GARCH model [39].

Although extracting useful features from the financial market is a challenging issue, convolutional neural networks (CNN) have been largely effective in dealing with this problem. This method, among other modern methods, is used mainly for automatic feature selection and market forecasting. Like the ANN method, CNN is based on AI and was used by [40] to identify faces. In another study, Yang, J. applied the deep convolutional neural networks (DCNN) method to identify human activities over multi-channel time series [41]. On the other hand, [13] introduced an artificial neural network approach to forecast the stock market indices to predict trends. As shown by [42], the CNN could use technical indicators for each type of sample but failed to consider correlations between stock markets as another possible source of information. Finally, Hoseinzade, E. and S. Haratizadeh introduced a CNN-based framework for the data collection from various sources such as different markets to extract features for the future prediction of these markets [43].

Since the long-term prediction of time series is a very challenging issue, long short-term memory (LSTM) is another approach that can be used to maintain long-term dependencies. The LSTM method was used by [14] to predict the price of Bitcoin. In a similar study, ref. [15] compared LSTM with the ARIMA method in predicting the price of Bitcoin. Similarly, Siami-Namini, S. examined the ARIMA and LSTM methods in time-series predictions [44]. Kim, H.Y. and C.H. Won predicted Korea Composite Stock Price Index 200 (KOSPI 200) stock market price volatility using LSTM models and GARCH [45]. To predict the financial time-series models, ref. [46] used the complete ensemble empirical mode decomposition with adaptive noise and the LSTM method. In the same vein, Tomar, A. and N. Gupta predicted the spread of the COVID-19 virus in China using the LSTM method [47]. Finally, Livieris, I.E. employed the CNN-LSTM hybrid method to forecast the gold market [48].

Classical methods have a linear nature, making their employment in complex and non-linear situations inappropriate. Accordingly, the current study takes advantage of AI models to address this problem. Furthermore, the output of the GARCH and ARCH models feeds the neural networks. Since the CNN model is very efficient in extracting valuable features from the primary data and the LSTM model helps identify short-term and long-term dependencies [48], a combination of these two models has been used in this study to predict Brent Crude Oil prices.

### 3. Materials and Methods

#### 3.1. Methodology

This section discusses the CNN and the LSTM basic concepts separately and in detail, after which the developed hybrid model is presented in the forthcoming section. In the end, we demonstrate the effectiveness of our proposed model by testing it with a simple trading strategy.

##### 3.1.1. CNN

CNN is one of the deep learning models, the components of which include an input layer, convolutional layers, nonlinear activation function, pooling layers, and a fully connected layer. In most cases, the input layer data of this network is one-dimensional, two-dimensional, or three-dimensional. This paper converts the features of the time series

of Brent Oil into two-dimensional inputs to take advantage of the CNN network. The approach taken for converting them into 2D inputs will be presented in the coming sections.

### Convolutional Layers

Convolution layers work by applying filters to each sample input and passing them to the next layer that is convoluted by filter size. These newly generated kernels, containing valuable information obtained from the primary data, will make the input into the following layers. Finally, they produce a matrix with high-level features as an output of convolutional layers. The freshly obtained matrix is usually given as an input to the fully connected layer after having completed the convolution process and passing through the max pooling layers. According to [49], the input data should have a matrix form; therefore, data processing continues by taking the two-dimensional image (matrices). Since convolution operations aim to extract high-level features, more than one convolution layer is typically used. The first layer of convolution usually extracts the low-level features and will be able to extract the high-level features by adding the subsequent layers of the network. Hence, the network uses two layers of convolution to extract high-level features in the proposed model. More precisely, as discussed in the study by [48], convolution layers convolute the input data. Thus, a convolution kernel, which has dimensions of  $3 \times 3$  or  $5 \times 5$  in most cases, is applied to the initial matrix. The kernel of the convolution used in this model is  $3 \times 3$ , which slides from the upper left of the image on the matrix, with a specified stride length (the stride length is considered one in this research). Each time the multiplication occurs across the kernels, whose entries are updated and changed after each network training, this process continues until the kernel covers the whole input matrices. The CNN model of this research used the padding technique, which is commonly employed to control the shrinkage of dimension after applying filters larger than  $1 \times 1$ . Moreover, it avoids losing information at the boundaries [50].

### Max Pooling Layer

Like convolution layers, this layer is used to extract useful features from the convolved matrix, which is accomplished by reducing the dimensions of the matrix. This layer extracts the maximum amount of matrix elements covered each time by the kernel. It also helps the network become more robust, the output of which will eventually be a matrix with lower dimensions.

### 3.1.2. LSTM Model

LSTM is a branch of the recurrent neural network (RNN) model with the capacity of learning long-term dependencies. Sometimes, we need past information and data trends to make a more reliable prediction. RNNs can use long-term information in processing current data; however, they can learn only a limited number of short-term dependencies, and the RNN networks lose their ability to recall past information at longer distances. LSTM can be considered an extension to RNN that can connect past information with present ones and maintain long-term data dependencies during data processing. While there is one processing layer in RNN, we have four processing units in LSTM. Each LSTM block consists of a memory cell as well as input, forget, and output gates. Gates are a way to enter information voluntarily. Gates in the LSTM network assist information processing with the sigmoid activation function where the output is between 0 and 1. The forget gate controls the flow of information from the previous step, which determines whether to use the previous information and, if yes, how much of it should be affected in the next layers. The input gate controls the new information, which examines whether we use this current information in the process, and if so, how much of it should be used. Finally, the output gate examines how much information from the previous and current time steps should be combined and transferred to the next step.

These features help LSTM with controlling the flow of information and learning long-term dependencies [51,52]. Figure 1 shows the architecture of an LSTM network block. The symbols used in the model and their definitions are presented in Table 1.

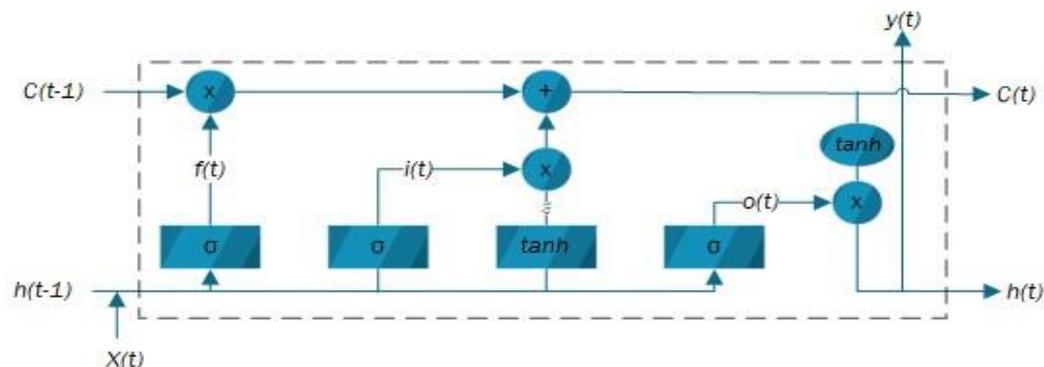


Figure 1. Each unit of LSTM Architecture.

Table 1. LSTM Terms.

Definition	Symbol
The input gate	$i_t$
The forget gate that controls the previous information	$f_t$
The second gate that controls the new information	$C_t^*$
The state of memory at the time t	$C_t$
Output gate which manages the information and could be used as the memory cell output	$o_t$
The input	$x_t$
The hidden state that constitutes the memory cell output	$h_t$
Weight matrices	$U_* \text{ \& } W_*$
The bias term vectors	$B_*$
The sigmoid function	$\sigma$
The component-wise multiplication operator	$\odot$

It should be noted that  $C_t$  and  $h_t$  in each layer of LSTM are entered as inputs to the next layer. At the forget gate, the  $f_t$  function output is a number of zero or one, indicating which flow of information should be stopped or continued. A value of one means that the value of the  $C_{t-1}$  state cell is entirely moved to  $C_t$ , and a value of zero means that the  $C_{t-1}$  state cell information is cleared and none of it is taken to the  $C_t$  state cell. Equation (1) shows the calculation of each unit of LSTM:

$$\begin{aligned}
 i_t &= \sigma(U_i x_t + W_i h_{t-1} + b_i) \\
 f_t &= \sigma(U_g x_t + W_g h_{t-1} + b_g) \\
 C_t^* &= \tanh(U_c x_t + W_c h_{t-1} + b_c) \\
 C_t &= f_t \odot C_{t-1} + i_t \odot C_t^* \\
 o_t &= \sigma(U_o x_t + W_o h_{t-1} + b_o) \\
 h_t &= o_t \odot \tanh(C_t)
 \end{aligned}
 \tag{1}$$

### 3.1.3. Trading Strategy

The proposed model offers the closing price of the next day, and trades are executed accordingly. Hence, if the Brent Crude Oil return of the previous day is less than the predicted value, a long trade is taken. However, if the Brent Crude Oil return of the previous day is greater than the predicted value, a short-sell trade will be considered. In

addition, as long as the direction of return does not change, the buying or selling position is opened. In other words, it will not be closed at the end of the day to prevent the increase of transaction costs, while the transaction cost is reduced as a percentage of profit.

### 3.2. Comparative and Proposed Models

This section will first review the benchmark models used to compare the performance and then dive into the details of the proposed model. In each sub-section, a brief review of the selected model is given, and the approach of this study to take advantage of them for forecasting the return is then discussed.

#### 3.2.1. ANN and ANN-PCA

ANN contains a set of interconnected neurons in different layers that exchange information with each other. This network consists of three primary layers, including an input layer, hidden layers, and an output layer. In the input layer, the information of each neuron, which is a vector, is connected to the hidden layer neurons by a connection (synapse). The random weights are assigned to each neuron of the input layer connecting to the hidden layer units. The input vectors are multiplied by the corresponding weights of each connection, and the nonlinear activation function is used. Therefore, the weights of the hidden layer components are updated through backpropagation derivatives. Then the output of all processing units from each layer is given as input to the next layer. If the number of layers is large, this process is called deep neural network (DNN).

In this study, the ANN model is used in a similar way to the proposed models by [12]. The PCA is applied to reduce the size of the input dataset so that it does not omit any variables but generates new variables with more important information achieved from original inputs. Another advantage is that the new variables are independent and have no correlation [53]. In the current study, we choose two principal components that can justify 0.95 of data variability. The PCA considered in this study is similar to that used by [54]. Consequently, the ANN is utilized to predict oil return, after which we use the PCA technique and its output with the ANN network in the second method. The model proposed by [55] is used to implement the hybrid PCA-ANN method.

#### 3.2.2. CNN-ANN and LSTM

Another comparative model is the CNN-ANN hybrid model based on the model developed by [56]. The output of the CNN network is input to the ANN part of the model containing fully connected layers. The first fully connected layer has 200 neurons, while the second dense layer, the third layer, and the fourth layer contain 100, 60, 10 neurons respectively. Each layer uses the RELU activation function, after which the output of the CNN-ANN hybrid model is obtained. Figure 2 shows the proposed CNN-ANN architecture. The LSTM method was implemented drawing upon the studies by [51] for the data before COVID-19 and [47] for the COVID-19 data.

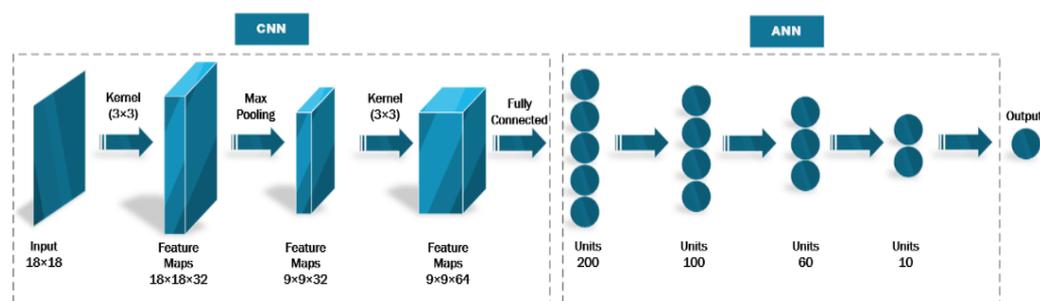


Figure 2. Proposed CNN-ANN Model Architecture.

### 3.2.3. The Proposed 2D CNN-LSTM

In the following, the proposed 2D CNN-LSTM is explained.

At first, the inputs are converted into 2D images, as performed in the CNN-ANN model, by considering every 18 days as the rows of the matrix and the features as the columns. As mentioned above, each  $18 \times 18$  is considered an image (matrix), and these inputs feed the models. The CNN model is very efficient in extracting useful features from the primary data, while the LSTM model is useful for detecting long-term and short-term dependencies [48]. The current study has combined these two models taking into account the capabilities of both in predicting Brent Crude Oil return. Our proposed model is based on the CNN-LSTM hybrid model, in which we have used two layers of convolution and one layer of max pooling. The first and second layers of convolution consist of 32 and 64 filters, respectively. The CNN network outputs, which are valuable features, are connected to the fully connected layer, whose output is given as a new input to the LSTM network. The output of this network is then given to another fully connected layer, and finally, the 2D CNN-LSTM output will have one dimension. Figure 3 illustrates the flowchart of the main steps of our study. Figure 4 illustrates the hybrid model of these two networks. Table 2 shows information about the characteristics of each of these methods. Python implements the models, and the library used in coding deep networks is PyTorch.

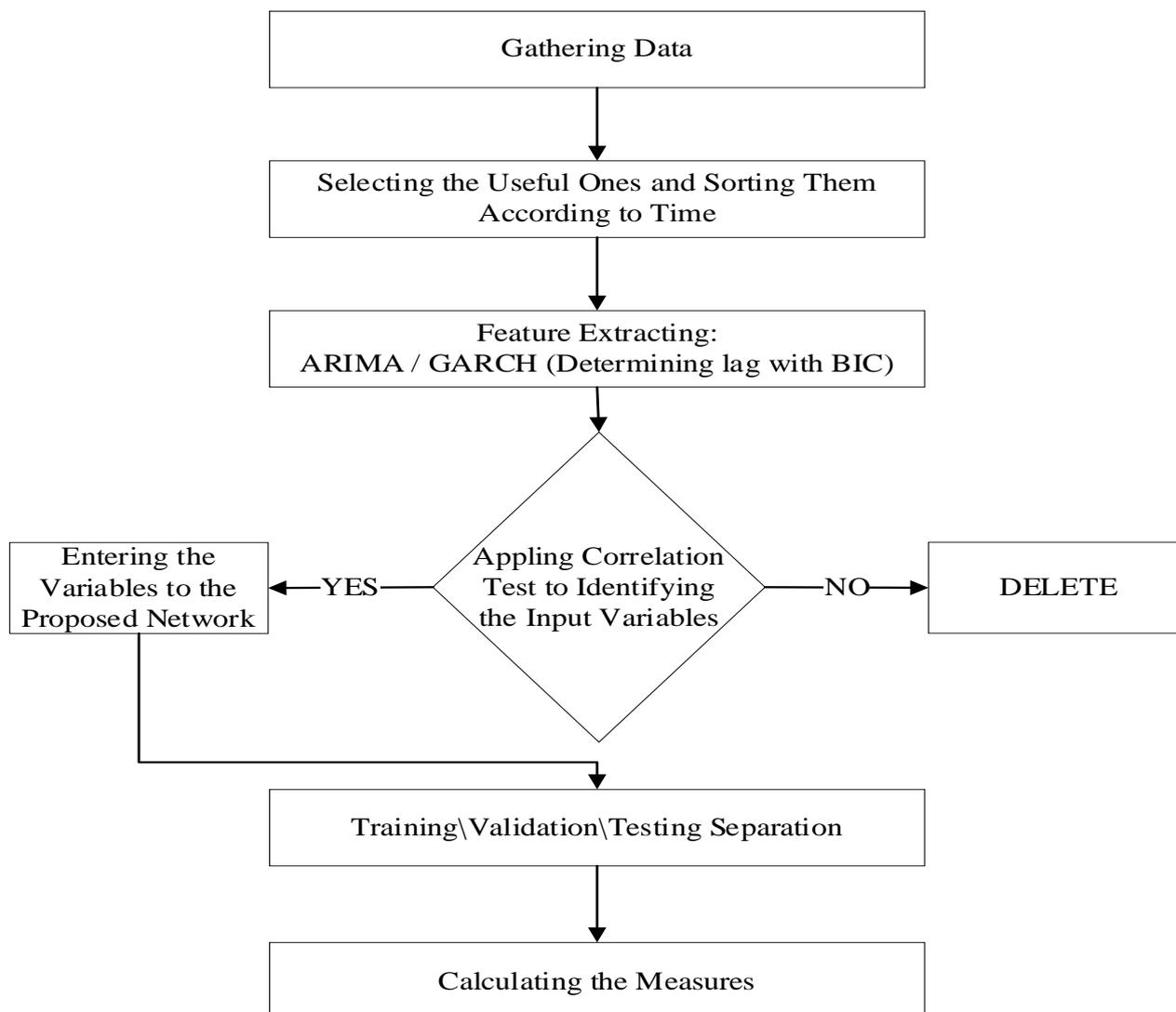
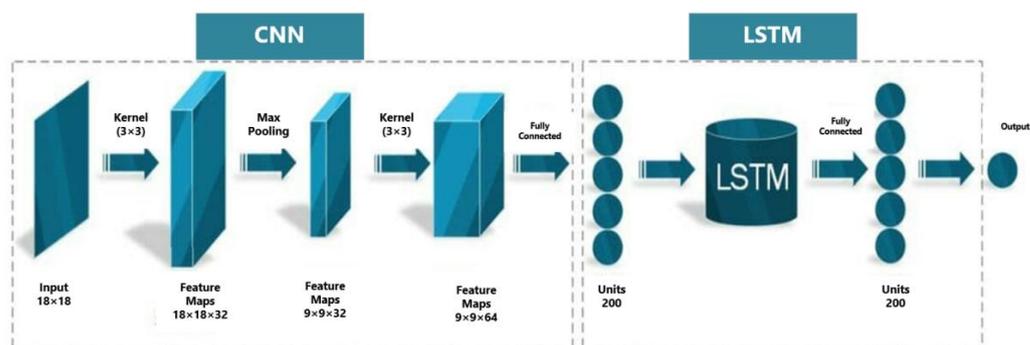


Figure 3. Flowchart of main steps.



**Figure 4.** The Proposed 2D CNN–LSTM Model Architecture.

**Table 2.** Information of Models.

Model	Description
ANN	Five hidden layers in size order of 200, 150, 120, 80, and 50 neurons Activation function RELU Learning rate 0.01 Batch normalization Cost function MSE
PCA-ANN	Five hidden layers in size order of 20, 30, 20, 15, and 10 neurons Activation function RELU Learning rate 0.01 Batch normalization Cost function MSE
CNN-ANN	The convolutional layer including 32 filters size (3 × 3) Max pooling layer (2 × 2) The convolutional layer of 64 filters size (3 × 3) Five hidden layers in size order of 200, 150, 120, 80, 50 neurons Activation function RELU Learning rate 0.01 Batch normalization Cost function MSE
LSTM	100 units
2D CNN-LSTM	The convolutional layer including 32 filters of size (3 × 3), max pooling layer (2 × 2) The convolutional layer including 64 filters of size (3 × 3) Fully connected layer with 200 neurons LSTM layer with 100 units

### 3.3. Characteristics of the Data

In this study, we collected Brent Crude Oil’s daily prices (The data were collected from [www.finance.yahoo.com](http://www.finance.yahoo.com)) from 17 February 2013, to 30 December 2019, before COVID-19, and from 31 December 2019, to 24 September 2020, after the outbreak of the pandemic. We also accumulated Severe Acute Respiratory Syndrome (SARS) data from 18 November 2002, to 19 May 2004. All data were collected from Investing and Yahoo Finance databases. This study uses time-series data to predict Brent Crude Oil return. The input data are normalized according to Equation (2). Table 3 shows the statistical characteristics of the series.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

Table 4 demonstrates the features used in this study before the outbreak of COVID-19. As different markets affect each other, and there is a high correlation between them, we used features such as exchange rates, crude oil, and natural gas prices to forecast Brent

Crude Oil return. It should be noted that the outputs of GARCH and ARIMA are used as inputs of our models. For the selected period, the return changes have been illustrated in Figure 5.

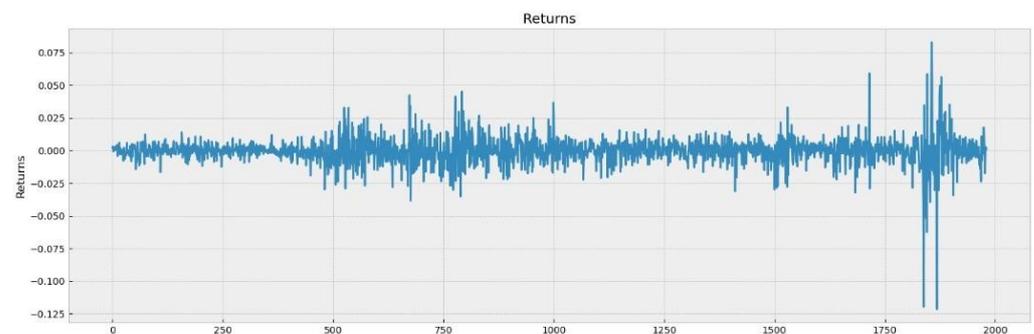
**Table 3.** Data characteristics.

Obs.	1980
Mean	−0.0002
Max	0.0828
Min	−0.1215
Variance	0.0106
Skewness	−1.0340
Kurtosis	21.7231
Q2(10)	6.4905
ARCH test (10)	26.7749

Q2(10): the 10th order of the Ljung–Box Q test for the squared returns. ARCH test (10): the 10th order of Engle’s ARCH test for the squared returns, for a significance level of 5%.

**Table 4.** Selected features as the inputs of the models.

1	Price Normalized	10	8-day lag-return
2	Return	11	9-day lag-return
3	1-day lag-return	12	10-day lag-return
4	2-day lag-return	13	Index EURO/Dollar
5	3-day lag-return	14	Price Natural Gas
6	4-day lag-return	15	US Dollar Index
7	5-day lag-return	16	Price Crude Oil
8	6-day lag-return	17	ARIMA(4,0,3)
9	7-day lag-return	18	GARCH(4,3)



**Figure 5.** The daily return of Brent Crude Oil in the selected period.

### 3.3.1. Performance Evaluation

Two types of mathematical and financial tests, described in detail below, have been used to evaluate the performance of the models in predicting the future return of Brent Oil and the accuracy of the forecasts.

#### Mathematical Tests

Five computational measures, namely mean square error (MSE), root mean square error (RMSE), mean absolute percentage error (MAPE), mean forecast error (MFE), and mean absolute error (MAE), are applied to evaluate the efficiency of the models in forecasting the future return of Brent Crude Oil. Table 5 shows the formula of each of measure, demonstrating the error each time the neural networks are executed. When the neural networks are run for the first time, the model assigns initial values with the random weight

and the bias vectors. Then, the neural network calculates its error at each execution and updates the error values, weights, and biases by minimizing the derivatives.

**Table 5.** The Computational Measures' Formula.

Evaluation Criteria	Formula	Descriptions
MSE	$\frac{\sum_{v=1}^n (A_v - F_v)^2}{n}$	n = number of total iterations each run A <sub>v</sub> = actual value F <sub>v</sub> = forecast value
RMSE	$\left( \frac{\sum_{v=1}^n (A_v - F_v)^2}{n} \right)^{\frac{1}{2}}$	
MAE	$\frac{\sum_{v=1}^n  A_v - F_v }{n}$	
MAPE	$\frac{\sum_{v=1}^n \left  \frac{A_v - F_v}{A_v} \right }{n}$	
MFE	$\frac{\sum_{i=1}^n (e_i)}{n}$	

### Financial Evaluation

The financial evaluation shows the overall performance of the proposed model by implementing the trading strategy in the real world. Brent Crude Oil is bought and sold or held taking into account the predicted return of crude oil according to the trading strategy. To evaluate the financial performance of the proposed model, risk-based performance criteria are used by calculating the spread between the actual return and the hypothetical benchmark return [57]. There are five most widely used methods to calculate risk-adjusted returns, which are discussed in detail Table 6 shows the evaluation criteria and Table 7 presents the formula of each measure.

**Table 6.** Evaluation criteria.

Evaluation Criteria	Description
Sharpe Ratio	As a measure of the excess return earned per unit of volatility over the risk-free rate, [58,59] the Sharpe ratio can be determined.
Deflated Sharpe Ratio	The deflated Sharpe ratio is used to determine the probability that a discovered strategy is a false positive [60].
Sortino Ratio	In terms of the Sharpe ratio, the Sortino ratio [61] is a variation on this. Sortino calculates the portfolio's return by dividing it by its downside risk (downside risk refers to the volatility of returns below a certain level, most commonly the average return of the portfolio or returns below zero). The ratio of return generated per unit of the downside risk is represented by Sortino.
Maximum Drawdown	A maximum drawdown (MDD) is defined as the maximum loss that a portfolio has experienced between a peak and trough before a new peak is reached. During a specified period, maximum drawdown is an indicator of downside risk [62,63].
Information Ratio	In contrast to the volatility of returns, the information ratio (IR) measures portfolio returns that exceed the returns of a benchmark, usually an index. The benchmark used is usually an index representing the market or a specific sector [64].

**Table 7.** Risk-adjusted return criteria formula.

Evaluation Criteria	Formula	Description
Sharpe Ratio	$\frac{R_p - R_f}{\sigma_p}$	$R_p$ = Expected Portfolio Return $R_f$ = Risk-Free Rate $\text{Sigma}(p)$ = Standard Deviation of Portfolio
Deflated Sharpe Ratio	$D\hat{S}R = Z \left[ \frac{(\hat{S}R - \hat{S}R_0)\sqrt{T-1}}{\sqrt{1 - \hat{\gamma}_3 \hat{S}R + \frac{\hat{\gamma}_4 - 1}{4} \hat{S}R^2}} \right]$	$\hat{S}R_0$ = The expected maximum Sharpe ratio $\hat{S}R$ = The estimated Sharpe ratio $T$ = The sample length $\hat{\gamma}_3$ = The skewness of the return's distribution $\hat{\gamma}_4$ = The kurtosis of the return's distribution $N$ = The number of independent trials
Sortino Ratio	$\frac{R_p - R_f}{\sigma_d}$	$\text{Sigma}(d)$ = A measure of the negative asset return's standard deviation
Maximum Drawdown	$\frac{LP - PV}{PV}$	$LP$ = Lowest value after peak value $PV$ = Peak value
Information Ratio	$\frac{R_p - R_b}{\text{Tracking error}}$	$R_b$ = Return's benchmark rate Tracking error = Excess return standard deviation compared to return's benchmark rate

## 4. Results

This section reports the results and outputs of the ANN, PCA-ANN, CNN-ANN, LSTM, and 2D CNN-LSTM models. The information of implementing these methods is provided in Table 2 and has been explained in the previous sections. Given that each trader can choose different horizons for their trades, we decided to choose the forecast horizon of our model for the next one day and the next five days to address this issue.

### 4.1. Before the COVID-19 Pandemic

The pre-COVID-19 data are divided into three segments of training, cross-validation, and testing. They included 1795 data observations, of which 1500 were used for training and cross-validation sets and the remainder (1501–1795 observations) for the testing set.

#### 4.1.1. Computational Performance Evaluation

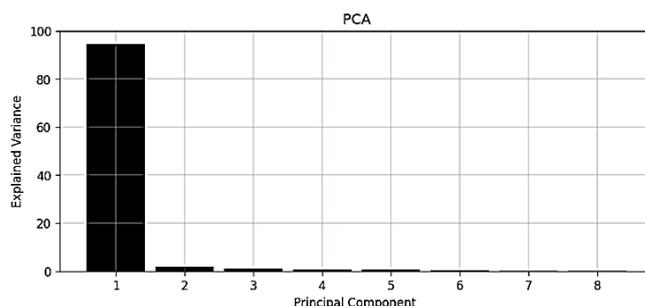
Table 8 displays the results of the one-day-ahead forecast horizon, according to which the 2D CNN-LSTM method has a far better forecasting performance than other models in terms of evaluation measures. The 2D CNN-LSTM has the lowest values on MSE, RMSE, MAE, MAPE, and MFE with values of 0.0001, 0.0122, 0.0039, 0.1301, and 0.0017, respectively. According to Table 8, after the 2D CNN-LSTM model, the LSTM and CNN-ANN models provide better performance than the others. As was mentioned earlier, we used PCA to reduce the dimensions. The principal component is derived from the linear combination of input data and the new space direction matrix. We set the value of the principal components (PCs) at 18 in this study for the PCA-ANN model. Table 9 presents the explained PCs' variance in descending order, the first PC describing the most variability. Figure 6 illustrates that PC 1 and PC 2 could describe approximately 95% of the data's volatility.

**Table 8.** The Amount of Each Measure for pre-COVID-19 with One-Day-Ahead Forecast.

	ANN	PCA-ANN	CNN-ANN	LSTM	2D CNN-LSTM
MSE	0.0004	0.0003	0.0003	0.0002	0.0001
RMSE	0.0190	0.0175	0.0173	0.0149	0.0122
MAE	0.0146	0.0135	0.0132	0.0050	0.0039
MAPE	2.3440	1.4999	2.0530	0.2070	0.1301
MFE	−0.0070	−0.0078	0.0081	0.0086	0.0017

**Table 9.** The Degree of Variability Explained by PCs.

PCs	Explained Variance	PCs	Explained Variance
1	0.9453	10	0.0003
2	0.0185	11	0.0002
3	0.0104	12	0.0001
4	0.0088	13	0.0001
5	0.0074	14	0.0001
6	0.0047	15	0.0001
7	0.0015	16	0.0000
8	0.0013	17	0.0000
9	0.0013	18	0.0000



**Figure 6.** PCs Chart.

It was also noticed that approximately 0.0084% of data was lost when using PC 1 and PC 2 as calculated through Equation (3). The use of PCA in addition to ANN in forecasting the oil return on a one-day-ahead forecasting greatly improved the forecasting accuracy and brought the forecast return closer to the actual value. The observed values for each of the five evaluation measures used in the PCA-ANN model are lower than those in the ANN model. As Table 10 demonstrates, we considered the forecast horizon for five days. According to the obtained results, the proposed model, namely 2D CNN-LSTM, had higher efficiency compared with other models. Based on the observations, 2D CNN-LSTM had the lowest values in MSE RMSE, MAE, and MAPE, which were equal to 0.0001, 0.0119, 0.0033, and 0.1023, respectively. Like the one-day-ahead forecast horizon, in the five-day-ahead forecast horizon, the LSTM model and then CNN-ANN had better performances after the 2D CNN-LSTM model. However, unlike the one-day-ahead forecasting horizon where the use of PCA improved the performance of the ANN model, the use of PCA in the five-day-ahead forecasting did not work well. In other words, the prediction using the ANN model had a better performance without the use of PCA.

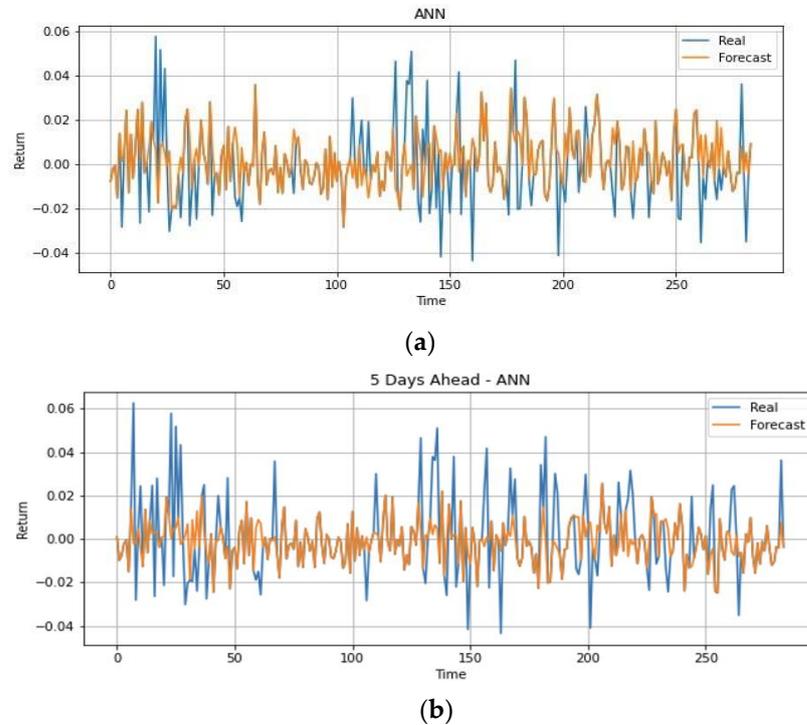
$$\frac{\frac{1}{m} \sum_{i=1}^m \| x^{(i)} - x_{approx}^{(i)} \|^2}{\frac{1}{m} \sum_{i=1}^m \| x^{(i)} \|^2} \leq 0.01 \tag{3}$$

Regarding the observed values of the models implemented in this study to predict the return of Brent Crude Oil before the outbreak of the COVID-19 pandemic, it can be stated that the models outperformed in the five-day-ahead forecast horizon. The values for MSE were 0.0001 and 0.0001 for the one-day- and five-day-ahead forecasts, respectively, 0.0122 and 0.0119 for the RMSE, 0.0039 and 0.0033 for the MAE, 0.1301 and 0.1023 for the MAPE, and finally 0.0017 and 0.0010 for the MFE.

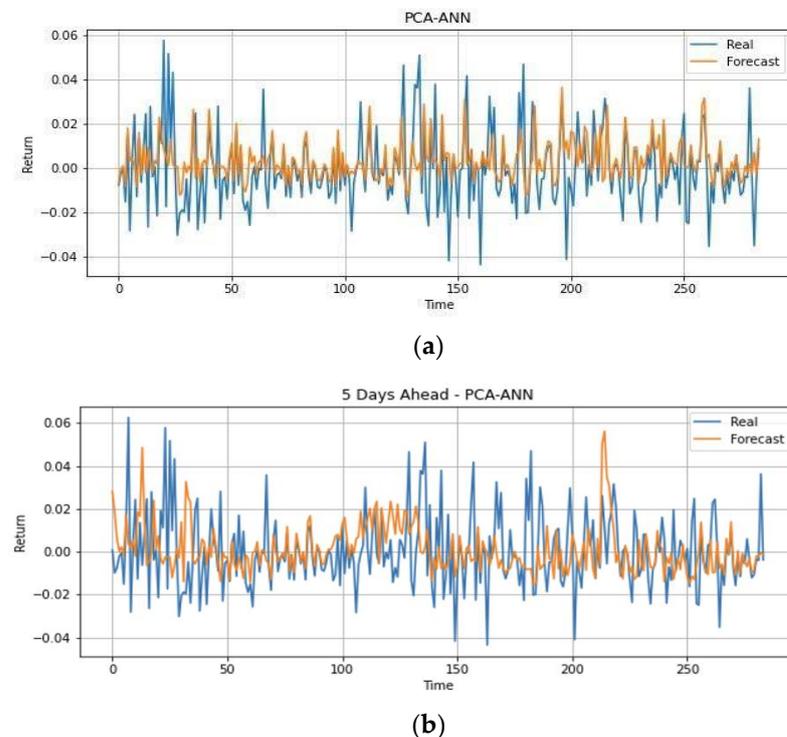
**Table 10.** The Amount of Each Measure for pre-COVID-19 with Five-Day-Ahead Forecast.

	ANN	PCA-ANN	CNN-ANN	LSTM	2D CNN-LSTM
MSE	0.0003	0.0003	0.0003	0.0002	0.0001
RMSE	0.0177	0.0177	0.0165	0.0141	0.0119
MAE	0.0135	0.0145	0.0066	0.0048	0.0033
MAPE	1.5511	1.926	1.0537	0.1495	0.1023
MFE	0.0002	0.0022	0.0001	0.0017	0.0010

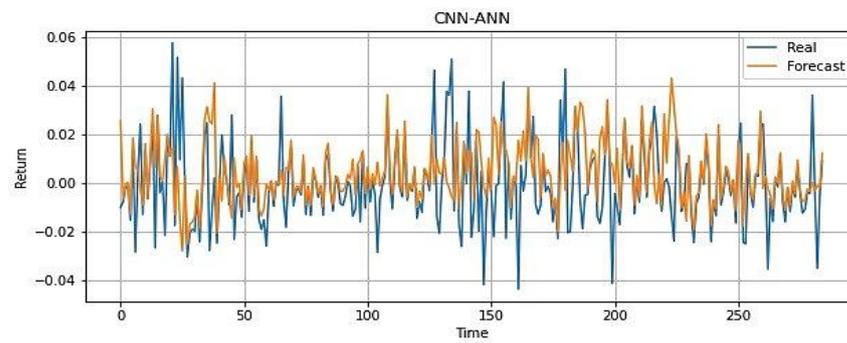
In Figure 7, The Brent Crude Oil return forecast can be observed using the ANN method with the one- and five-day-ahead forecast horizons. Figure 8 illustrates Brent Crude Oil return forecast using PCA-ANN. Figure 9 also demonstrates the output of the CNN-ANN method. Finally, Figure 10 displays LSTM, and Figure 11 illustrates 2D CNN-LSTM predicting the future return of Brent Crude Oil in the one-day- and five-day-ahead forecasting horizons.



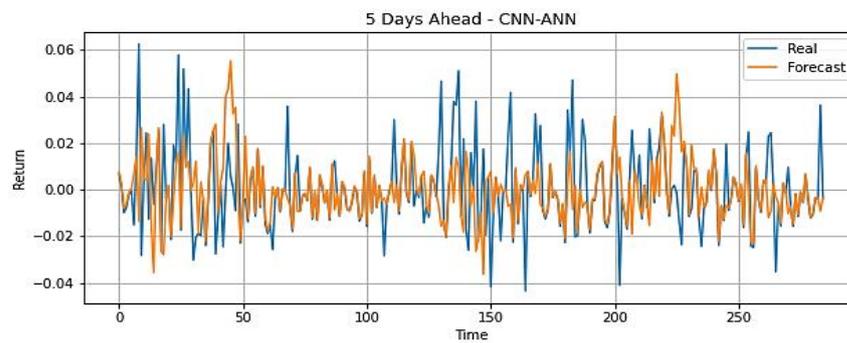
**Figure 7.** ANN Forecasting Result for One-Day- (a) and Five-Day-Ahead (b).



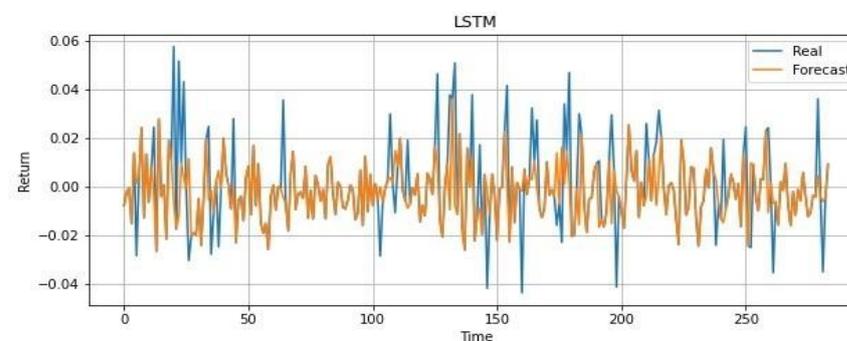
**Figure 8.** PCA-ANN Forecasting Result for One-Day- (a) and Five-Day-Ahead (b).



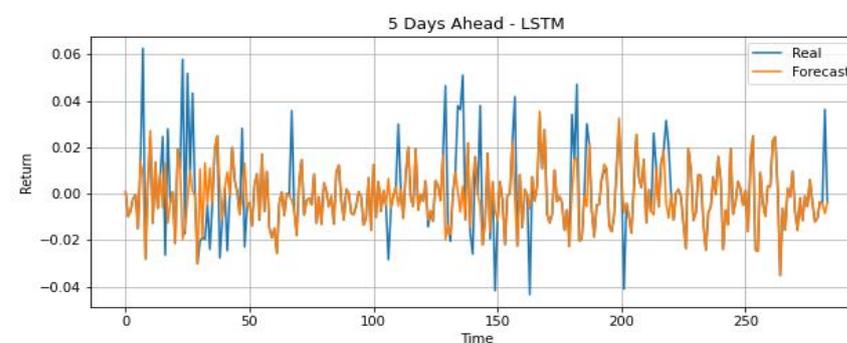
(a)



(b)

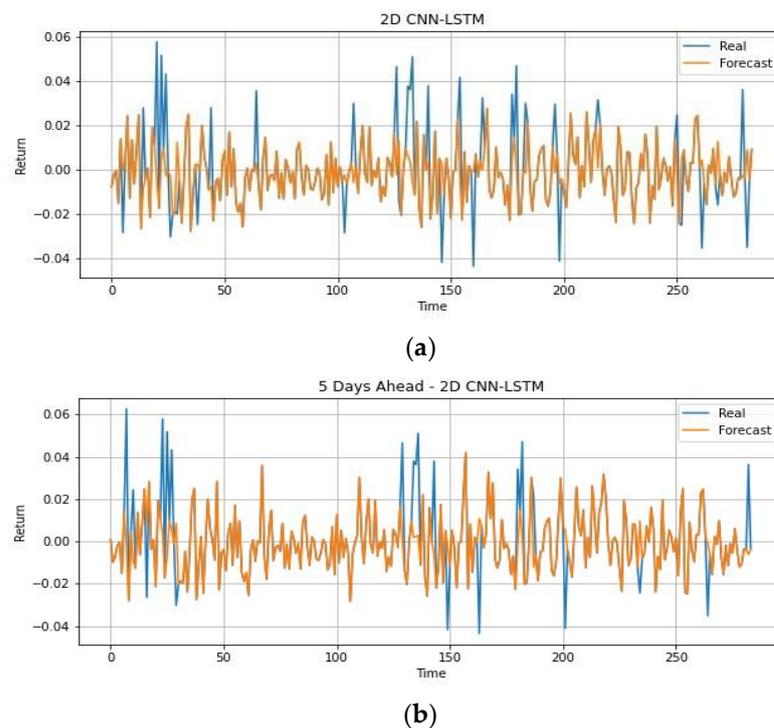
**Figure 9.** CNN-ANN Forecasting Result for One-Day- (a) and Five-Day-Ahead (b).

(a)



(b)

**Figure 10.** LSTM Forecasting Result for One-Day- (a) and Five-Day-Ahead (b).



**Figure 11.** 2D CNN-LSTM Forecasting Result for One-Day- (a) and Five-Day-Ahead (b).

#### 4.1.2. Financial Performance Evaluation

After predicting the future return of Brent Crude Oil using the presented models, this section adopts buying and selling decisions according to the trading strategy presented in Section 3.3, after which financial trades are implemented with related sequential buying and selling pairs. The results of the financial evaluation with the one-day- and five-day-ahead forecast horizons are given below Table 11.

**Table 11.** Descriptive Statistics of Daily Brent Crude Oil Return's with One-Day-Ahead.

Model	Mean	Standard Deviation	SE Mean	95% CI for $\mu$
ANN	0.0007	0.0155	0.0006	(−0.0004, 0.0019)
PCA-ANN	0.0020	0.0139	0.0006	(0.0008, 0.0031)
CNN-ANN	0.0006	0.0055	0.0006	(−0.0005, 0.0018)
LSTM	0.0021	0.0063	0.0006	(0.0009, 0.0032)
2D CNN-LSTM	0.0026	0.0130	0.0006	(0.0015, 0.0038)

Table 12 provides the descriptive statistics of daily Brent Crude Oil return prediction using the proposed method with the one-day- and five-day-ahead forecast horizons, respectively.

**Table 12.** Descriptive Statistics of Daily Brent Crude Oil Return's with Five-Day-Ahead.

Model	Mean	Standard Deviation	SE Mean	95% CI for $\mu$
ANN	0.0016	0.0141	0.0006	(0.0005, 0.0028)
PCA-ANN	0.0015	0.0115	0.0006	(0.0004, 0.0027)
CNN-ANN	0.0021	0.0114	0.0006	(0.0010, 0.0033)
LSTM	0.0022	0.0150	0.0006	(0.0010, 0.0033)
2D CNN-LSTM	0.0033	0.0111	0.0006	(0.0022, 0.0045)

In Table 13, we considered the forecast horizon for 1 day, confirming the higher performance of the proposed model, namely 2D CNN-LSTM, compared with other models. According to observations, 2D CNN-LSTM had the highest values in Sharpe ratio, Sortino ratio, and information ratio, which were equal to 0.2582, 0.5388, and 0.2491, respectively,

and the lowest value in the criterion of maximum drawdown. After the 2D CNN-LSTM model, the CNN-ANN model and then LSTM performed better than the other models. According to the results of deflated Sharpe ratio and considering the values of 0.15, 0.2 and 0.25 for the expected Sharpe ratio, 88%, 56%, and 99% of the proposed strategy had a positive Sharpe ratio, respectively. In other words, there were 12%, 44%, and 1% chances that this strategy would not make earnings.

**Table 13.** The Amount of Each Measure with One-Day-Ahead Forecast.

	ANN	PCA-ANN	CNN-ANN	LSTM	2D CNN-LSTM
Sharpe Ratio	0.0808	0.0900	0.1441	0.1110	0.2582
Deflated Sharpe Ratio (0.15)	0.0163	0.0340	0.1401	0.0600	0.8801
Deflated Sharpe Ratio (0.2)	0.0012	0.0040	0.0204	0.0076	0.5658
Deflated Sharpe Ratio (0.25)	0.8620	0.8774	0.9917	0.9439	0.9999
Maximum Drawdown	0.1059	0.1038	0.0952	0.1349	0.0764
Sortino Ratio	0.1268	0.1036	0.2762	0.1637	0.5388
Information Ratio (IR)	0.0737	0.0813	0.1353	0.1043	0.2491

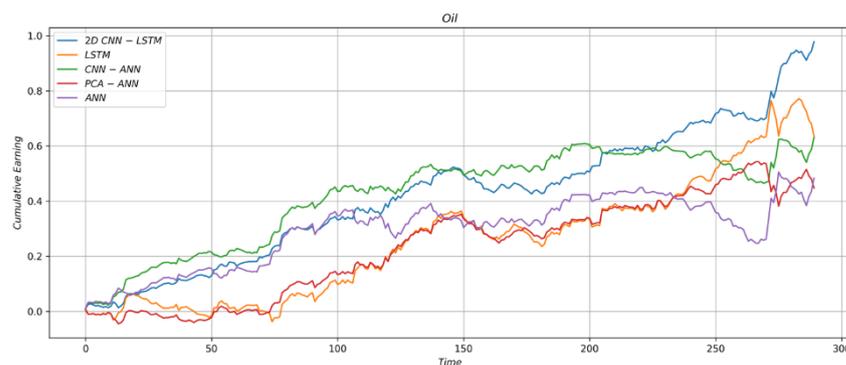
Table 14 provides the five-day forecast horizon, based on which the 2D CNN-LSTM model had the highest values in Sharpe ratio, Sortino ratio, and information ratio equal to 0.2501, 0.4005, and 0.2341, respectively, and the lowest amount in the criterion of maximum drawdown with value of 0.0790. After the 2D CNN-LSTM model, the LSTM model performed better than the other models. According to the deflated Sharpe ratio results and considering the values of 0.15, 0.2, and 0.25 for the expected Sharpe ratio, 1.21%, 13%, and 50% of the proposed strategy had a positive Sharpe ratio, respectively.

**Table 14.** The Amount of Each Measure with Five-Day-Ahead Forecast.

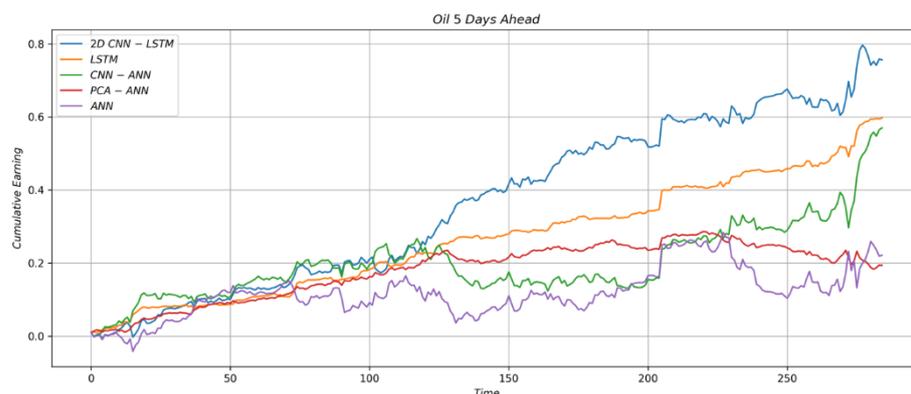
	ANN	PCA-ANN	CNN-ANN	LSTM	2D CNN-LSTM
Sharpe Ratio	0.0158	0.1062	0.0266	0.1638	0.2501
Deflated Sharpe Ratio (0.15)	0.0119	0.2202	0.0183	0.5999	0.9879
Deflated Sharpe Ratio (0.2)	0.0010	0.0492	0.0017	0.2546	0.8704
Deflated Sharpe Ratio (0.25)	0.0000	0.0056	0.0001	0.0581	0.5012
Maximum Drawdown	0.1724	0.1424	0.0524	0.1220	0.0790
Sortino Ratio	0.0127	0.1554	0.0128	0.2513	0.4005
Information Ratio (IR)	0.0093	0.0990	0.0085	0.1561	0.2341

#### 4.1.3. Trading Strategy Results

As shown in the diagram Figures 12 and 13, based on the presented trading strategy, the cumulative profit obtained by considering the forecasted return of crude oil with the one-day- and five-day-ahead forecasting horizons, respectively, was higher using the 2D CNN-LSTM model compared with other models. According to the results, the LSTM model and then CNN-ANN had better performances after the 2D CNN-LSTM model.



**Figure 12.** Comparison of Cumulative Trading Profit with One-Day-Ahead.



**Figure 13.** Comparison Of Cumulative Trading Profit with Five-Day-Ahead.

#### 4.2. During COVID-19 Outbreak

First, the data during the COVID-19 outbreak, with 188 observations, were divided into training and testing sets. Using the sliding window technique, the data were converted into a series of dependent and continuous sequences. Because this study considered the window size of 20, the number of sets of independent series became 168.

Because of the lack of data during the outbreak of the COVID-19 pandemic, this study used SARS data. It is demonstrated that COVID-19 behavior would be similar to the SARS pandemic. Therefore, this study tried to select a portion of the SARS data that behaved more like COVID-19. First, the oil returns during the SARS period were separated into sets in which the length of each set was 188 because the number of observations for COVID-19 was 188. Therefore, 300 SARS data were separated into 188 observation-length data series, after which the correlation between each of the SARS-derived sets and the 188 COVID-19 series was examined. The Pearson Correlation Coefficient test was used to investigate the relationship between SARS and COVID-19 data series. This research considered no correlation between the series as the null hypothesis of the test. The alternative hypothesis was a nonzero correlation value at a confidence level of 95%. After calculating the correlation between all SARS and COVID-19 data series, the SARS data series that could not reject the null hypothesis at the significance level were kept, and the rest were left out. Then, this research kept the series showing the highest correlation with the COVID-19 data series from the remaining data series. The results obtained from this test showed that the highest correlation coefficient was the 60th data series, leading to the selection of set 60 according to the description. Then, this study separated the SARS data from 60 to 300 and used it as an aid for network training. Moreover, the sliding window was applied to convert the data into different segments.

The window size used in this study was 20 for the 188 COVID-19 data records used in network training.  $X_i$  to  $X_{i+19}$  data with  $i = 1, 2, \dots, 168$  was utilized to predict the next day's return as the response variable  $Y_{i+20}$ . For instance, when we consider  $i$  equal to 1, data  $X_1$  to  $X_{20}$  is selected to predict  $Y_{21}$  (the return of day 21). Figure 14 shows the sliding window process with a window size of 20.

The effect of COVID-19 on oil return is similar to that of SARS regarding the above descriptions. Because the spread of COVID-19 is faster than that of SARS, in this paper, we decided to use the SARS data in addition to those of COVID-19 in the network training. Therefore, the 220 SARS data sets and 130 COVID-19 data observations were employed for the network training and the rest for network testing.

We evaluated the performance of the model according to the previously introduced measures. Considering the above explanations, we compared the COVID-19 and SARS data with the forecast horizons of one and five days in network training. Given the observations shown in Figure 15, related to the one-day-ahead forecast horizon, the results demonstrate that the LSTM model (SARS + COVID-19) performs better than the LSTM (COVID-19) model. The LSTM (SARS + COVID-19) and the LSTM (COVID-19) models produced the

values of 0.0002 and 0.0005 for the MSE, 0.0153 and 0.0224 for RMSE, 0.0110 and 0.0158 for MAE, 2.2044 and 2.7714 for MAPE, and 0.0067 and  $-0.0056$  for MFE. Table 15 Given these values, it can be deduced that the performance of the LSTM network (SARS + COVID-19) in the prediction of Brent Crude Oil return has increased significantly and improved the network performance.

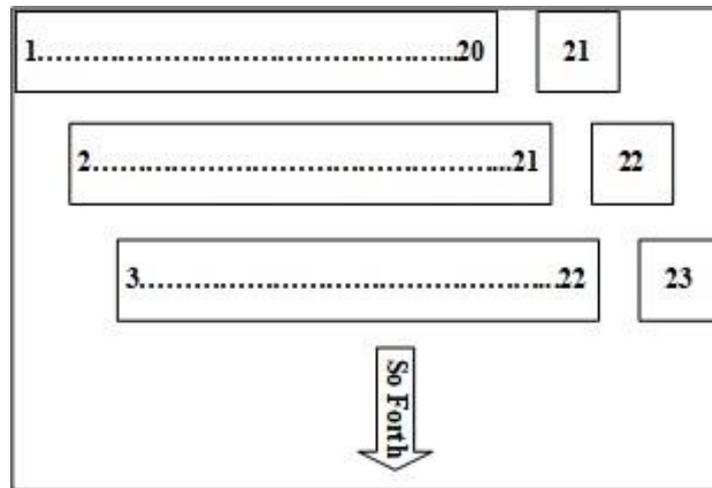


Figure 14. Process of Sliding Window.

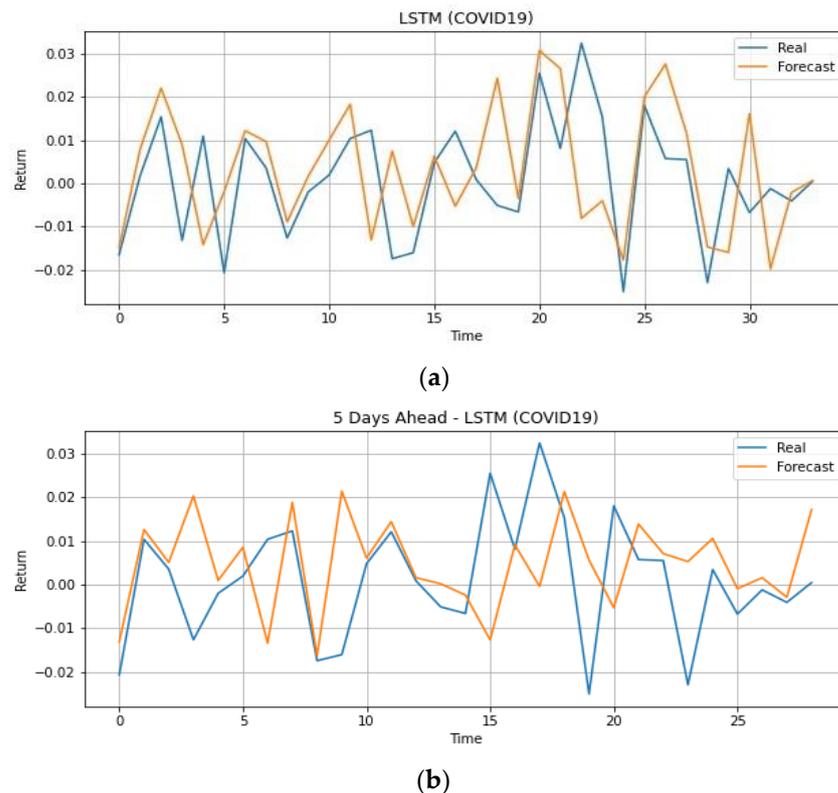


Figure 15. LSTM COVID-19 Forecasting Result After COVID-19 Pandemic with One-Day- (a) and Five-Day-Ahead (b) Forecast.

Table 16 shows the five-day-ahead forecast horizon. The values of the LSTM (SARS + COVID-19) and LSTM (COVID-19) models are 0.0003, 0.0003 for MSE, 0.0177 and 0.0184 for RMSE, 0.0145 and 0.0143 for MAE, 1.9260 and 2.8300 for MAPE, and 0.0022 and  $-0.0001$  for MFE. Given these values, the LSTM network (SARS COVID-19) has a better performance than the other models in predicting Brent Crude Oil return concerning the three criteria

of MSE, RMSE, and MAPE. Based on the observed values, the performance of LSTM (SARS + COVID-19) is better in the one-day- than five-day-ahead forecast horizon.

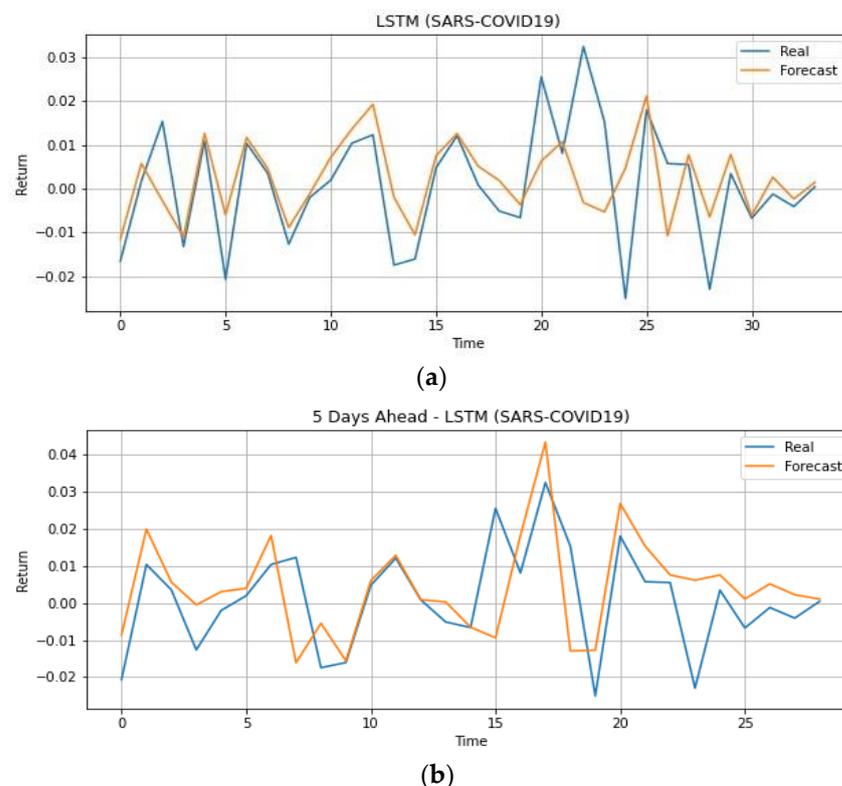
**Table 15.** The Value of Each Measure for COVID-19 Period with One-Day-Ahead Forecast.

	LSTM (SARS + COVID-19)	LSTM (COVID-19)
MSE	0.0002	0.0005
RMSE	0.0153	0.0224
MAE	0.0110	0.0158
MAPE	2.2044	2.7714
MFE	0.0067	−0.0056

**Table 16.** The Value of Each Measure for the COVID-19 Period with Five-Day-Ahead Forecast.

	LSTM (SARS + COVID-19)	LSTM (COVID-19)
MSE	0.0003	0.0003
RMSE	0.0177	0.0184
MAE	0.0145	0.0143
MAPE	1.9260	2.8300
MFE	0.0022	−0.0001

Figure 15 shows the prediction of Brent Crude Oil return using the LSTM method with one-day- and five-day-ahead forecast horizons during the outbreak of the COVID-19 pandemic. Figure 16 demonstrates the diagram of the return for LSTM (SARS + COVID-19). The difference of this method with simple LSTM is the consideration of oil price return data during the SARS outbreak, which is considered as an input in the implementation of the network and has significantly improved the network performance of the model.



**Figure 16.** LSTM (SARS + COVID-19) Forecasting Result After COVID-19 Pandemic with One-Day- (a) and Five-Day-Ahead (b) Forecast.

## 5. Conclusions

The current study has investigated the problem of forecasting future returns of Brent Crude Oil by proposing a new hybrid CNN-LSTM to take advantage of each. Moreover, to enhance the performance of the proposed models, the outputs of calibrated ARIMA and GARCH models, along with other variables such as return lags, price, and macroeconomic variables, have been considered as explanatory features for model training. According to the obtained results and performance measures, our proposed model performs better than others illustrated in the literature, including ANN, PCA-ANN, LSTM, and CNN-ANN. The tests performed on the proposed model using the trading strategy showed a higher cumulative profit from trading with the prediction results of the proposed two-dimensional model compared with other models presented in this research. Furthermore, we used the SARS pandemic data that had a high correlation with COVID-19 data to tune the proposed models to increase their efficiency during the COVID-19 pandemic. The results demonstrate that the proposed LSTM model, tuned by the SARS data, can better predict the Brent Crude Oil return during the COVID-19 pandemic. Using accurate forecasts of crude oil return for various economic decision-making problems such as real options valuation, portfolio optimization, and designing investment strategies is a promising further research area. Our proposed ML-based method can also be combined with state-of-the-art papers in the realm of energy delivery to predict the amount of oil product cargo and set up a sustainable process considering [65].

**Author Contributions:** Conceptualization, S.M.A.S. and E.H.; methodology, E.H. and P.K.; software, S.M.A.S. and P.K.; validation, S.M.A.S. and P.K.; formal analysis, E.H. and B.D.; investigation, E.H. and S.F.; resources, S.M.A.S.; data curation, E.H.; writing—original draft preparation, P.K. and S.F.; writing—review and editing, E.H., B.D. and S.D.; visualization, S.M.A.S., P.K., S.F. and S.D.; supervision, E.H. and B.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

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

## References

1. Deng, S.; Xiang, Y.; Nan, B.; Tian, H.; Sun, Z. A hybrid model of dynamic time wrapping and hidden Markov model for forecasting and trading in crude oil market. *Soft Comput.* **2020**, *24*, 6655–6672. [[CrossRef](#)]
2. Quayyoun, S.; Khan, M.H.; Shah, S.Z.A.; Simonetti, B.; Matarazzo, M. Seasonality in crude oil returns. *Soft Comput.* **2020**, *24*, 13547–13556. [[CrossRef](#)]
3. Wang, B.; Wang, J. Forecasting hybrid neural network with variational learning rate and q-DSCID synchronization evaluation for energy market. *Soft Comput.* **2020**, *24*, 16811–16828. [[CrossRef](#)]
4. Zhang, L.; Wang, J. Forecasting global crude oil price fluctuation by novel hybrid E-STERNN model and EMCCS assessment. *Soft Comput.* **2021**, *25*, 2647–2663. [[CrossRef](#)]
5. Engle, R.F. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econom. J. Econom. Soc.* **1982**, *50*, 987–1007. [[CrossRef](#)]
6. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [[CrossRef](#)]
7. Kristjanpoller, W.; Minutolo, M.C. Gold price volatility: A forecasting approach using the Artificial Neural Network–GARCH model. *Expert Syst. Appl.* **2015**, *42*, 7245–7251. [[CrossRef](#)]
8. Hajizadeh, E.; Mahootchi, M.; Esfahanipour, A. A new NN-PSO hybrid model for forecasting Euro/Dollar exchange rate volatility. *Neural Comput. Appl.* **2019**, *31*, 2063–2071. [[CrossRef](#)]
9. Yu, L.; Wang, S.; Lai, K.K. A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Appl. Soft Comput.* **2009**, *9*, 563–574. [[CrossRef](#)]
10. Lu, Y.K.; Perron, P. Modeling and forecasting stock return volatility using a random level shift model. *J. Empir. Finance* **2010**, *17*, 138–156. [[CrossRef](#)]
11. Racine, J. On the nonlinear predictability of stock returns using financial and economic variables. *J. Bus. Econ. Stat.* **2001**, *19*, 380–382. [[CrossRef](#)]
12. Hajizadeh, E.; Seifi, A.; Zarandi, M.F.; Turksen, I. A hybrid modeling approach for forecasting the volatility of S&P 500 index return. *Expert Syst. Appl.* **2012**, *39*, 431–436. [[CrossRef](#)]

13. Di Persio, L.; Honchar, O. Artificial neural networks architectures for stock price prediction: Comparisons and applications. *Int. J. Circuits Syst. Signal Process.* **2016**, *10*, 403–413.
14. Wu, C.-H.; Lu, C.-C.; Ma, Y.-F.; Lu, R.-S. A New Forecasting Framework for Bitcoin Price with LSTM. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018.
15. Karakoyun, E.; Cibikdiken, A. Comparison of ARIMA Time Series Model and LSTM deep Learning Algorithm for Bitcoin Price Forecasting. In Proceedings of the 13th Multidisciplinary Academic Conference in Prague, Prague, Czech Republic, 12–13 October 2018.
16. Bildirici, M.; Bayazit, N.G.; Ucan, Y. Analyzing Crude Oil Prices under the Impact of COVID-19 by Using LSTARGARCHLSTM. *Energies* **2020**, *13*, 2980. [[CrossRef](#)]
17. Dey, P.; Saurabh, K.; Kumar, C.; Pandit, D.; Chaulya, S.K.; Ray, S.K.; Prasad, G.M.; Mandal, S.K. t-SNE and variational auto-encoder with a bi-LSTM neural network-based model for prediction of gas concentration in a sealed-off area of underground coal mines. *Soft Comput.* **2021**, *25*, 14183–14207. [[CrossRef](#)]
18. Bollerslev, T.; Chou, R.Y.; Kroner, K.F. ARCH modeling in finance: A review of the theory and empirical evidence. *J. Econ.* **1992**, *52*, 5–59. [[CrossRef](#)]
19. Bauwens, L.; Laurent, S.; Rombouts, J.V.K. Multivariate GARCH models: A survey. *J. Appl. Econ.* **2006**, *21*, 79–109. [[CrossRef](#)]
20. Lam, K.S.; Tam, L.H. Liquidity and asset pricing: Evidence from the Hong Kong stock market. *J. Bank. Finance* **2011**, *35*, 2217–2230. [[CrossRef](#)]
21. Wang, L.; Feng, C.; Song, Q.; Yang, L. Efficient semiparametric garch modeling of financial volatility. *Stat. Sin.* **2012**, *22*, 249–270. [[CrossRef](#)]
22. Maciel, L.; Gomide, F.; Ballini, R. Evolving Fuzzy-GARCH Approach for Financial Volatility Modeling and Forecasting. *Comput. Econ.* **2016**, *48*, 379–398. [[CrossRef](#)]
23. Sadik, Z.A.; Date, P.M.; Mitra, G. News augmented GARCH(1,1) model for volatility prediction. *IMA J. Manag. Math.* **2018**, *30*, 165–185. [[CrossRef](#)]
24. Naimy, V.; Haddad, O.; Fernández-Avilés, G.; El Khoury, R. The predictive capacity of GARCH-type models in measuring the volatility of crypto and world currencies. *PLoS ONE* **2021**, *16*, e0245904. [[CrossRef](#)]
25. Hamid, S.A.; Iqbal, Z. Using neural networks for forecasting volatility of S&P 500 Index futures prices. *J. Bus. Res.* **2004**, *57*, 1116–1125. [[CrossRef](#)]
26. Pérez-Rodríguez, J.V.; Torra, S.; Andrada-Félix, J. STAR and ANN models: Forecasting performance on the Spanish “Ibex-35” stock index. *J. Empir. Financ.* **2005**, *12*, 490–509. [[CrossRef](#)]
27. Wang, Y.-H. Nonlinear neural network forecasting model for stock index option price: Hybrid GJR-GARCH approach. *Expert Syst. Appl.* **2009**, *36*, 564–570. [[CrossRef](#)]
28. Bildirici, M.; Ersin, Ö.Ö. Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange. *Expert Syst. Appl.* **2009**, *36*, 7355–7362. [[CrossRef](#)]
29. Roh, T.H. Forecasting the volatility of stock price index. *Expert Syst. Appl.* **2007**, *33*, 916–922. [[CrossRef](#)]
30. Adhikari, R.; Agrawal, R.K. A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Comput. Appl.* **2014**, *24*, 1441–1449. [[CrossRef](#)]
31. Mohammed, G.T.; Aduda, J.A.; Kube, A.O. Improving Forecasts of the EGARCH Model Using Artificial Neural Network and Fuzzy Inference System. *J. Math.* **2020**, *2020*, 6871396. [[CrossRef](#)]
32. Tay, F.E.; Cao, L. Application of support vector machines in financial time series forecasting. *Omega* **2001**, *29*, 309–317. [[CrossRef](#)]
33. Tang, L.-B.; Tang, L.-X.; Sheng, H.-Y. Forecasting volatility based on wavelet support vector machine. *Expert Syst. Appl.* **2009**, *36*, 2901–2909. [[CrossRef](#)]
34. Chen, C.-H.; Yu, W.-C.; Zivot, E. Predicting stock volatility using after-hours information: Evidence from the Nasdaq actively traded stocks. *Int. J. Forecast.* **2012**, *28*, 366–383. [[CrossRef](#)]
35. Zhiqiang, G.; Huaiqing, W.; Quan, L. Financial time series forecasting using LPP and SVM optimized by PSO. *Soft Comput.* **2013**, *17*, 805–818. [[CrossRef](#)]
36. Geng, L.; Liang, Y.; Zhang, Z.; Shi, X. Forecasting Range Volatility Using Support Vector Machines with Improved PSO Algorithms. *Telkomnika* **2016**, *14*, 208. [[CrossRef](#)]
37. Zhao, J.; Mao, X.; Chen, L. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed. Signal Process. Control* **2019**, *47*, 312–323. [[CrossRef](#)]
38. Lu, C.-J.; Lee, T.-S.; Chiu, C.-C. Financial time series forecasting using independent component analysis and support vector regression. *Decis. Support Syst.* **2009**, *47*, 115–125. [[CrossRef](#)]
39. Sun, H.; Yu, B. Forecasting Financial Returns Volatility: A GARCH-SVR Model. *Comput. Econ.* **2020**, *55*, 451–471. [[CrossRef](#)]
40. Lawrence, S.; Giles, C.L.; Tsoi, A.C.; Back, A.D. Face recognition: A convolutional neural-network approach. *IEEE Trans. Neural Netw.* **1997**, *8*, 98–113. [[CrossRef](#)]
41. Yang, J.; Nguyen, M.N.; San, P.P.; Li, X.; Krishnaswamy, S. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In Proceedings of the International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
42. Gunduz, H.; Yaslan, Y.; Cataltepe, Z. Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowl.-Based Syst.* **2017**, *137*, 138–148. [[CrossRef](#)]

43. Hoseinzade, E.; Haratizadeh, S. CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Syst. Appl.* **2019**, *129*, 273–285. [[CrossRef](#)]
44. Siامي-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.
45. Kim, H.Y.; Won, C.H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Syst. Appl.* **2018**, *103*, 25–37. [[CrossRef](#)]
46. Cao, J.; Li, Z.; Li, J. Financial time series forecasting model based on CEEMDAN and LSTM. *Phys. A Stat. Mech. Appl.* **2019**, *519*, 127–139. [[CrossRef](#)]
47. Tomar, A.; Gupta, N. Prediction for the spread of COVID-19 in India and effectiveness of preventive measures. *Sci. Total Environ.* **2020**, *728*, 138762. [[CrossRef](#)]
48. Livieris, I.E.; Pintelas, E.; Pintelas, P. A CNN–LSTM model for gold price time-series forecasting. *Neural Comput. Appl.* **2020**, *32*, 17351–17360. [[CrossRef](#)]
49. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
50. Li, W.; Zhu, L.; Shi, Y.; Guo, K.; Cambria, E. User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models. *Appl. Soft Comput.* **2020**, *94*, 106435. [[CrossRef](#)]
51. Sagheer, A.; Kotb, M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* **2019**, *323*, 203–213. [[CrossRef](#)]
52. Baek, Y.; Kim, H.Y. ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Syst. Appl.* **2018**, *113*, 457–480. [[CrossRef](#)]
53. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459. [[CrossRef](#)]
54. Chen, Y.; Hao, Y. Integrating principle component analysis and weighted support vector machine for stock trading signals prediction. *Neurocomputing* **2018**, *321*, 381–402. [[CrossRef](#)]
55. Zhong, X.; Enke, D. Forecasting daily stock market return using dimensionality reduction. *Expert Syst. Appl.* **2017**, *67*, 126–139. [[CrossRef](#)]
56. Haggag, M.; Abdelhay, S.; Mecheter, A.; Gowid, S.; Musharavati, F.; Ghani, S. An Intelligent Hybrid Experimental-Based Deep Learning Algorithm for Tomato-Sorting Controllers. *IEEE Access* **2019**, *7*, 106890–106898. [[CrossRef](#)]
57. Hussain, O.K.; Dillon, T.S.; Hussain, F.K.; Chang, E.J. Risk Assessment Phase: Financial Risk Assessment in Business Activities. In *Risk Assessment and Management in the Networked Economy*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 151–185. [[CrossRef](#)]
58. Sharpe, W.F. Mutual fund performance. *J. Bus.* **1966**, *39*, 119–138. [[CrossRef](#)]
59. Khodaei, P.; Esfahanipour, A.; Taheri, H.M. Forecasting turning points in stock price by applying a novel hybrid CNN-LSTM-ResNet model fed by 2D segmented images. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105464. [[CrossRef](#)]
60. Bailey, D.H.; de Prado, M.L. Practical Applications of The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting, and Non-Normality. *J. Portf. Manag.* **2014**, *40*, 94–107. [[CrossRef](#)]
61. Mohan, V.; Singh, J.G.; Ongsakul, W. Sortino Ratio Based Portfolio Optimization Considering EVs and Renewable Energy in Microgrid Power Market. *IEEE Trans. Sustain. Energy* **2016**, *8*, 219–229. [[CrossRef](#)]
62. de Melo Mendes, B.V.; Lavrado, R.C. Implementing and testing the Maximum Drawdown at Risk. *Finance Res. Lett.* **2017**, *22*, 95–100. [[CrossRef](#)]
63. Esfahanipour, A.; Khodaei, P. A.; Khodaei, P. A Constrained Portfolio Selection Model Solved by Particle Swarm Optimization Under Different Risk Measures. In *Applying Particle Swarm Optimization*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 133–153. [[CrossRef](#)]
64. Goodwin, T.H. The information ratio. *Financ. Anal. J.* **1998**, *54*, 34–43. [[CrossRef](#)]
65. Szaruga, E.; Kłos-Adamkiewicz, Z.; Gozdek, A.; Załoga, E. Linkages between Energy Delivery and Economic Growth from the Point of View of Sustainable Development and Seaports. *Energies* **2021**, *14*, 4255. [[CrossRef](#)]