Long Short-Term Model for Brent Oil price forecasting

[1]Harsh Salvi, [2]Avdhi Shah, [3]Manthan Mehta, [*]Prof. Stevina Correia
[1, 2, 3]Student, [*]Faculty
Department of Information Technology,
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
[1]harshsalvi24@gmail.com, [2]avdhi1998@gmail.com, [3]manthu.mehta@gmail.com,
[*]stevina.dias@djsce.ac.in

Abstract

The oil market has its effect straightforwardly or in a roundabout way on the income distribution of countries influencing the stock market, average cost for basic items, education, essential commodities and many more. Moreover, in response, oil costs are influenced by various elements. In this manner, there is an unmistakable need to figure the oil value patterns. This challenge has been undertaken by numerous studies. Machine Learning has been the crucial crux for a lot of them. LSTM models have been used time and again for time series forecasting. This article studies the LSTM neural network and its use to predict future trends of Brent oil prices based on the previous price of Brent oil.

Keywords: *Brent oil, price forecasting, time series prediction, LSTM, RNN.*

1. Introduction

Recurrent neural network (RNN) is a model for predicting sequential data, for example, sound, time arrangement information or natural language. One of RNNs' strengths is the concept that they can relate previous information to the current task, such as using previous video frames to notify the present frame's understanding. You understand every word when you read this paper based on your interpretation of previous words. You do not neglect it all and start thinking again without any planning. LSTM is a recurrent neural network (RNN) architecture that remembers values over arbitrary intervals that have persistence. Traditional neural networks do not have any previous understanding, and it is a major shortcoming. Recurrent neural networks are used to address this issue. They are networks with loops in them, which allows information to persist. A few models of RNNs like LSTM are utilized for time series forecasting. LSTMs have a footing over standard feed-forward neural networks and RNN in some ways. This occurs due to the property of selectively remembering patterns for a long time. The expectation of Brent oil cost has stood out from industry to the scholarly community. Different AI calculations, for example, neural networks, genetic algorithms, support vector machines, and others are utilized to anticipate Brent oil costs. This paper shows the use of LSTM for the forecasting of Brent oil prices.

2. Related Works

In [1] LSTM-based predictive model was used for fault detection in a simulated cyber-attack with multivariate time series forecasting. In this approach real object data sets were used for normal and anomalous behavior. A mathematical model was created to get realistic data with anomalies. An LSTM neural network was trained and tested based on a self-consistent mathematical model and model variables. LSTM architecture parameters were adopted after investigation of the results. The end result was that LTSM provided better results and was advantageous than the classic fault-detection methods.

In order to effectively capture dynamic non-linear traffic, an LSTM network was utilized in [2]. In the following study, the LSTM network comprised of 3 layers, wherein the hidden layer was made up of blocks of memory, and by proper training method it automatically determines the optimal time lags. [3] was an improvement of [2]. In [3] An LSTM network was connected with multiple layers on memory-based units by constructing a cascade. An ODC matrix was also integrated into the LSTM network through vector generators and fully connected layers. To assist the LSTM network to capture the traffic flow features, the matrix included the spatial-temporal correlations of links inside the road network. This model had better accuracy than existing traffic forecast methods.

In [4], in order to foretell phone prices on European markets, a robust forecasting model using Support Vector Regression (SVR) and Long Short-Term Memory neural network (LSTM) was proposed. A comparison study of the two techniques was proposed. Various univariate models were studied and compared. Based on the study LSTM and SVM neural networks appear as the most accurate ones. In addition, comparison of multivariate models for both techniques was done by introducing more variables in the multivariate approach, better performance was obtained for the prediction. Though by using the univariate model the SVR model predicted the following day price with an RMSE of 38.88 Euros, but by using multivariate models, recurrent neural network LSTM gave the most precise prediction for the following day's price with an RMSE of 28.68 Euros.

For stock price prediction, in order to foretell price trends, the LSTM networks have displayed good performances in NLP by mostly using news text data as input. Work by making use of the price data to predict price movement has also been implemented, [6] to employ stock indexes along with historic price data to predict whether a stock price falls, rises or remains the same on a particular day. [5] differentiates the performance of LSTM and MLP against their own suggested method based on an amalgamation of Convolutional Neural Networks and wavelets, which performs better than both LSTM and MLP but has close results to the LSTM network.

In [7] a deep learning model was put forward for building a crude oil price predicting model. This model was utilized to capture the unknown complex nonlinear characteristics of the crude oil price movements. For the proposed model, the performance was evaluated using the price data in the WTI crude oil markets. In 2005, after several tests [8] concluded that time series for future prices is non-linear and stochastic. GARCH and ARMA techniques were compared to ANN and found that the latter produced better performance and results for crude oil price forecasting.

Currently, LSTM based RNN has been widely used in sequence-based problems such as text classification [10], stock market prediction [12], power demand forecasting [11] music notes recognition [13], machine translation [9], sentiment analysis [14].

3. Dataset Description

The dataset used was published on Kaggle. The Brent oil price movements are subject to various and dissimilar influencing factors. The dataset was derived from the U.S. Energy Information Administration. The dataset aims to predict future trends of the Brent Oil Prices based on available historical data. The dataset contains Brent oil prices for every day starting from the 17-May-1987 until the 30-September-2019. The dataset contains two columns, Date and Price. The price column is the daily Brent oil price in US

Dollars. The date column is presented in a standard datetime format. The price has been mentioned only for weekdays.

4. Methodology

The first step of the implementation is to use the date columns as an index. Then a min-max scaler is used to scale the data. The min-max scaler is a built-in function in scikit-learn. It is used to scale a range of data into a smaller scale. In this case, the range of Brent oil prices is scaled to a range of 0-1. This scaler has been proved to gives better results when the standard scaler does not work as desired. The min-max scaler is a much better option to utilize when the distribution does not follow a Gaussian pattern or when the distribution has a low standard deviation. Since the Brent oil price does not represent a Gaussian distribution, the min-max scaler was found to provide a more accurate representation of the prices. The next step is creating the dataset for the neural network to train on. The lookback value is the number of previous values which will be used as the "X" value for any particular "y" value. The lookback value set is 60. This means the features used for determining the Brent oil price on the 61st date would be the price on day 1-60. This method is used to create the dataset for training and testing. This means the new dataset has 61 values for every row. The first 60 values form the "X" matrix, while the last value is the "y" value. Once this dataset is created, it is split into a 70:30 ratio to form the training data and testing data.

The next step is designing the neural network model. As noted in previous time series forecasting results, Long Short-Term Memory (LSTM) networks give good accuracy. LSTM is a recurrent neural network (RNN) design that does not forget values over impulsive intervals. LSTM is appropriate to classify, process and predict time series given time slacks of unknown period. Relative lack of care toward gap length gives a favorable position to LSTM over alternative RNNs, hidden Markov models and other pattern learning techniques.

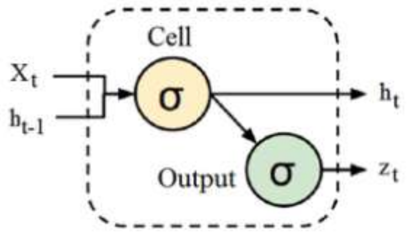

Fig 1. Standard RNN Cell

RNN cell, as shown in Fig 1, takes in two inputs which are output from the last hidden state and observation at time = t. Besides the hidden state, there's no info concerning the past to recollect. After a series of RNN cells, this leads to the problem of exploding and imploding cells. To counter this an LSTM cell is used.

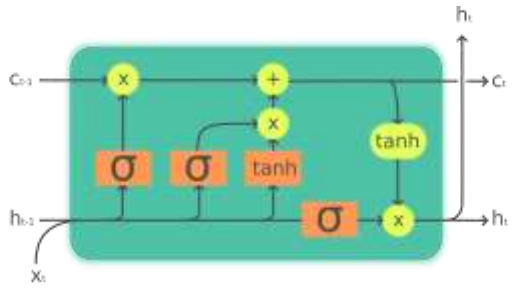

Fig 2. LSTM Cell

LSTMs are unequivocally designed to beat the long-term dependency problem. LSTMs are designed keeping in mind to retain information for a long time. All recurrent neural networks have a chain of repeating cells. In standard RNNs, this repeating cell has a simple structure, like a single tanh layer. LSTMs as shown in Fig 2 also have a similar chain-like structure. The difference, however, is that the repeating cell has a very different structure. The LSTM model has can retain and forget information as required using structures known as gates. These gates are used to selectively let information through. They consist of a sigmoid neural net layer followed by a pointwise multiplication operation. The sigmoid function outputs numbers in the range of 0 and 1. This number shows how much of each component is let through. A value of 0 means no information is let through, while a value of 1 means all information is let through. An LSTM has three of these gates, to control the cell state.

The model implemented, as shown in Fig 3, consists of 4 LSTM layers with 60 units each. Each LSTM unit is followed by a Dropout layer. The dropout level is set to 0.2. Dropout refers to ignoring neurons during the training phase. These neurons are chosen at random. By “ignoring”, it is meant that units are not considered during a particular forward or backward pass. At each training stage, individual nodes are dropped out of the network with probability p, so that a reduced network is left. Incoming and outgoing edges to a dropped-out node are removed. In our case, the probability of a neuron being dropped out is 0.2 i.e. 20%. The primary reason for adding a dropout layer is to prevent overfitting. Overfitting means when the model has adjusted too much for the training data. This leads to poor performance on the test data. Overfitting is a common problem in time series forecasting. In order to tackle this problem, dropout layers are implemented. For uniformity, dropout layers are added after every LSTM layer.

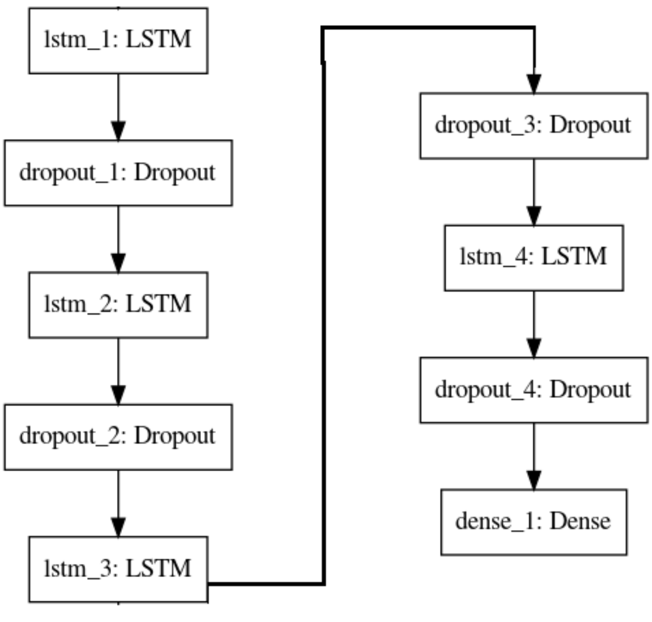

Fig 3. Keras Model

This model is then trained. The loss function used is the mean squared error function. This loss function does not differentiate the model predicting a higher value form a model predicting a lower value. Just the difference in the actual and predicted value is important. This loss function is a very common one used in cases of regression. The optimizer used is the Adam optimizer. Adam is an optimization algorithm that can be utilized as an alternative to the old-style stochastic gradient descent procedure to update neural network weights in an iterative manner based on training data. Stochastic gradient descent maintains one learning rate (termed alpha) for all weight updates and also the learning rate doesn't modify throughout learning. When using the Adam optimization, the learning rate alpha is a variable quantity. Each network weight has a different learning rate which adjusts itself at the time of learning. Adam optimizer combines the advantages of two of the extensions of stochastic gradient descent: AdaGrad and RMSProp. The model is trained for 25 epochs with a batch size of 64.

5. Results

The value of Brent oil price was initially scaled using the min-max scaler. For the data to be intelligible to the user, the values need to be scaled back to original prices. The inverse_transform function is used to scale the values back to the original range of Brent oil prices. Four types of errors have been calculated to check the accuracy of model. The errors calculated are the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). These calculations have been performed on the train as well as the test dataset. MAE is calculated by taking average of all the absolute error values of the predictions. RMSE is calculated by taking the square root of average of squared differences between predicted and observed values. Since MAE and RMSE are negatively-oriented values, lower values are better. The four values can be seen below in Table 1.

Train Mean Absolute Error	1.1962655198913466
Train Root Mean Squared Error	1.9164871479339118
Test Mean Absolute Error	2.215981231537119
Test Root Mean Squared Error	2.823476892225809

Table 1. Results

The loss function for the train and test dataset was calculated for every epoch. The loss describes the error in predictions. The loss function is the mean squared error function. The loss function is calculated on the basis of scaled values and not real values. This leads to difficulty in understanding the actual loss during each epoch. However, plotting a graph gives us a good idea as to what is the benefit of each epoch. The loss function calculated on un-scaled data would be in the same ratio. The graph for the same is shown in Fig 4.

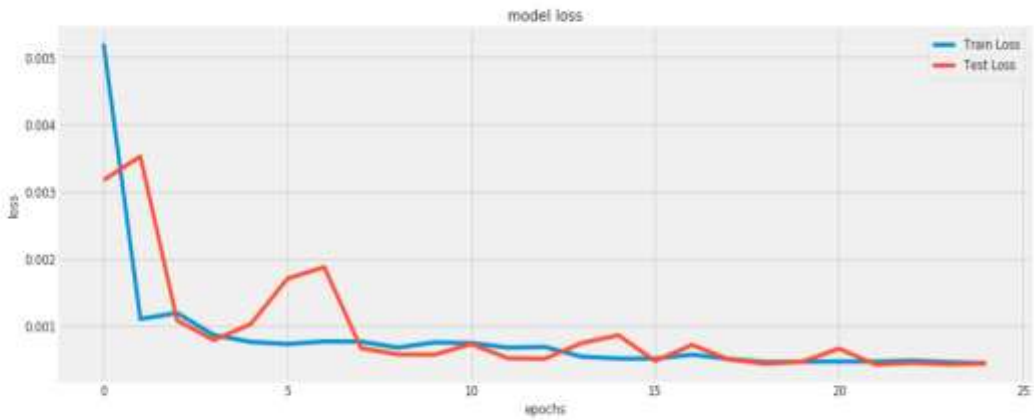

Fig 4. Loss vs Epochs graph

The graph, as shown in Fig 5, shows the plotting of actual and predicted values. It can be inferred that the LSTM model adapts to the trend of the actual value but is still away from the value. The model cannot predict sudden changes to the price value. This problem is faced during any of the time series forecasting issues. These changes likely happen due to the occurrence of some event in the real world.

Fig 5. Actual vs Predicted value graph

6. Conclusion and Future Work

Precisely predicting oil prices is important because it drives inflation and economic activity. Most studies focus on other oil pricing benchmarks rather than Brent oil. However, in recent times, Brent oil pricing serves as a global benchmark for oil pricing and hence it is important to accurately predict the pricing of Brent oil. In this paper, the Brent oil price dataset is divided into two sets, train and test dataset. LSTM is used to predict the price of Brent oil on the test dataset based on the model generated by the training dataset.

Pricing of oil is very unpredictable as it is affected by many factors that cannot be quantified, as they are qualitative in nature. Adding more factors to predict the value of Brent oil will help in improving the accuracy of the prediction. In addition, combining two different models could help in improving the accuracy of the predicted prices of Brent oil.

References

[1] Filonov, P., Lavrentyev, A. and Vorontsov, A., 2016. Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. *arXiv preprint arXiv:1612.06676*.

[2] Ma, X., Tao, Z., Wang, Y., *et al.*: 'Long short-term memory neural network for traffic speed prediction using remote microwave sensor data', *Transport. Res. C Emerging Technol.*, 2015, **54**, pp. 187–197

[3] Zhao, Z., Chen, W., Wu, X., Chen, P.C. and Liu, J., 2017. LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, *11*(2), pp.68-75.

[4] Chniti, G., Bakir, H. and Zaher, H., 2017, December. E-commerce time series forecasting using LSTM neural network and support vector regression. In *Proceedings of the International Conference on Big Data and Internet of Thing*(pp. 80-84). ACM.

[5] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *Big Data (Big Data), 2015 IEEE International Conference on*, Oct 2015, pp. 2823–2824.

[6] Xie, W., Yu, L., Xu, S. and Wang, S., 2006, May. A new method for crude oil price forecasting based on support vector machines. In *International Conference on Computational Science* (pp. 444-451). Springer, Berlin, Heidelberg.

[7] Yanhui Chen, Kaijian Heb, and Geoffrey K.F. Tso, "Forecasting Crude Oil Prices: a Deep Learning based Model", Procedia Computer Science, vol. 122, pp. 300-307, 2017.

[8] S. Moshiri, and F. Foroutan, "Forecasting nonlinear crude oil futures prices," The Energy Journal vol. 27, pp. 81-95, 2005.

[9] Ilya Sutskever, Oriol Vinyals, and Quok V Le. "Sequence to sequence learning with neural networks. Advances in neural information processing systems, pp. 3104–3112, 2014.

[10] Ji Young Lee and Franck Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. arXiv preprint arXiv:1603.03827 (2016).

[11] Y. Cheng, C. Xu, D. Mashima, V.L.L. Thing, and Y. Wu, "Power LSTM: Power Demand Forecasting Using Long Short-Term Memory Neural Network.", In: Cong G., Peng WC., Zhang W., Li C., Sun A. (eds) Advanced Data Mining and Applications. ADMA 2017. Lecture Notes in Computer Science, vol 10604. Springer.

[12] K. Chen, Y. Zhou, and F. Dai, "An LSTM-based method for stock returns prediction: A case study of China stock market," in Big Data (Big Data), 2015 IEEE International Conference on, Oct 2015, pp. 2823–2824.

[13] Yu Kang Jia, Zhicheng Wu, Yanyan Xu, Dengfeng Ke, and Kaile Su, "Long Short-Term Memory Projection Recurrent Neural Network Architectures for Piano's Continuous Note Recognition," Journal of Robotics, vol. 2017, Article ID 2061827, 7 pages, 2017.

[14] Minlie Huang, Yujie Cao, and Chao Dong, "Modeling rich contexts for sentiment classification with LSTM". arXiv preprint arXiv:1605.01478. 2016.