8/23/2020

TIME SERIES FORECASTING

A COMPARITIVE STUDY OF LSTM MODELS

Sumit Sharma
CDCSIT, TU

Tribhuvan University

Institute of Science and Technology

Seminar Report

On

Time series forecasting: A comparative study of LSTM models

Submitted to

Central Department of Computer Science & Information Technology

Tribhuvan University, Kirtipur

Kathmandu, Nepal

Submitted by

Sumit Sharma

Roll No.: 05 / 075

In partial fulfillment of the requirement for Master's Degree in Computer Science and Information Technology (M.Sc. CSIT), 2nd Semester

2020

TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this Seminar report is prepared under my supervision by **Mr. Sumit Sharma** entitled "*Time series forecasting: A comparative study of LSTM models*" be accepted as fulfillment in partial requirement for the degree of Masters of Science in Computer Science and Information Technology.

…………………………………

Mr. Balkrishna Subedi

CDCSIT, TU

Tribhuvan University

Institute of Science and Technology

LETTER OF APPROVAL

This is to certify that the seminar report prepared by **Mr. Sumit Sharma** entitled "*Time series forecasting : A comparative study of LSTM models*" in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Information technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Evaluation Committee

……………………………..	……………………........
Mr. Nawaraj Paudel	Mr. Balkrishna Subedi
Head of Department, CDCSIT	Supervisor

..…………………………………

(Internal)

ABSTRACT

Time series forecasting is the process of predicting future values and behavior of data by analyzing the previous historic data gathered over a long period of time. Traditional statistical models like moving average, auto regression, and autoregressive integrated moving averages have been widely used for forecasting time series data. However, due to stochastic dependency of data over time statistical models are not able to perform forecasting tasks reliably and effectively. Recently, the use of deep neural networks in the field of time series forecasting has caught attention of many researchers and have proved to be efficient and effective than traditional statistical models. Among different deep learning models, Long Short-Term Memory (LSTM) is widely used for time series data. This paper explores LSTM model and its variants for time series forecasting task. This paper is based on Brent oil prices dataset made available by U.S Energy Information Administration and gold price dataset made available by World Gold council. The main objective of the paper is to compare different existing LSTM models and evaluate their effectiveness in time series forecasting.

Table of Contents

List of abbreviation

ADAM	= Adaptive Moment Estimation
ANN	= Artificial Neural Networks
AR	= Autoregressive
ARMA	= Autoregressive Moving Average
ARIMA	= Autoregressive Integrated Moving Average
CNN	= Convolution Neural Networks
GRU	= Gated Recurrent Unit
LSTM	= Long Short-Term Memory
MA	= Moving Average
MLP	= Multilayered Perceptron
ReLU	= Rectified Linear Unit
RNN	= Recurrent Neural Networks
SANN	= Seasonal Artificial Neural Networks
SARIMA	= Seasonal Autoregressive Integrated Moving Average
TDNN	= Time Delayed Neural Networks
TL-ANN	= Time Lagged Artificial Neural Networks

List of figures

List of Tables

1. Introduction

Time series is as an ordered sequence of values measured at equal spaced time intervals [1, 2]. It is used to analyze and predict future values and trends by use of historic records of values over long period of time. It has wide range of applications in the field of economics, finance, marketing, climate, astronomy etc. [2]. Although, the study of time series forecasting has been going on for a long period of time a perfect solution to the problem is not available yet. Many different models have been purposed by researchers with some effectiveness. The research has shifted from traditional statistical models to machine learning models and deep leaning models over the course of time [2, 3]. The shift is due to the volatility and stochastic dependence of data, which makes the problem more and more complex that can be addressed effectively by artificial neural networks models, machine learning and deep learning models [3, 4].

Many Artificial Neural Networks (ANN) structures like Multilayer Perceptron's (MLP's), Time Delayed Neural Networks (TDNN's) have been proposed in the literature to deal with time varying patterns [3, 4]. However, recent studies show that Recurrent Neural Networks (RNN) are more efficient than these models for time series forecasting [3, 7]. RNN is a class of ANN with block like structure with loops, which allow the network to store information about the data [5, 6, 7, 8]. Each block in RNN uses its own stored information, its own input and output from preceding block to provide an output [5, 6, 7, 8]. The RNN models however are incapable of learning long-term dependencies [9].

Long Short-Term Memory (LSTM) is the class of RNN particularly designed for sequential data [10, 11]. LSTM can effectively avoid the problems of gradient disappearance and gradient explosion in the process of RNN training [10, 11]. LSTM has achieved promising results for time series prediction [3, 11, 12, 13]. Its units consist input gate, forget gate, and output gate, which are responsible for capturing information from data, store them and produce an output [10]. It is popular due to the ability of learning hidden long-term sequential dependencies, which actually helps in learning the underlying representations of time series [3, 11].

Due to popularity, many variants of LSTM have emerged and have had success with time series forecasting [3]. Peephole connection LSTM, gated recurrent units (GRU), stacked LSTM, Convolution LSTM (CNN LSTM) are few examples of LSTM variants used for time series forecasting. The main objective of the paper is to compare the different LSTM variants based on their effectiveness of prediction of time series data. The paper is based on Brent oil prices dataset made available by U.S Energy Information Administration and gold price dataset available in world gold council website [13, 14]. The study involves use of historical data to predict unknown data (separated from original data) and evaluation of different models by use of error and accuracy metrics.

2. Literature Review

The development of time series forecasting started from linear statistical models [2, 3]. These models use concept of regression, autocorrelation and moving averages for predicting future values [1]. Autoregressive (AR) models use the concept that the current output is dependent upon number of previous output values [2]. The number of previous values that an AR model takes as input to predict current output is termed as lags [2]. Moving Average (MA) model uses average values of past few data to predict current values [2]. The number of values that a MA model uses to predict current output is called window. Exponential smoothing is modified version of MA models which assign weights to past values on the basis of lag time period (i.e. the farthest value has lowest weight) while evaluating the current output [2, 15]. Exponential smoothing is effective and reliable compared to AR and MA models [2, 15].

In course of time, robust statistical models were introduced by combination of AR and MA models [2, 3, 16]. The ARMA model was first hybrid model introduced by P. Whittle in 1951, which uses the combination for time series forecasting [2, 16]. However, the model is limited, as it required the condition that the time series must be stationary [2, 16]. Box and Jenkins introduced Autoregressive Integrated Moving Average (ARIMA) and Seasonal ARIMA (SARIMA) models in 1970 [1, 2]. These models deal with non-stationary time series data by providing mechanism for creating stationary data [1, 2]. These are widely used statistical models for time series analysis and forecasting [3].

The development of machine learning and deep learning lead the researchers to research on possible use of these techniques for time series forecasting [2, 3]. Support Vector Machine (SVM) are widely used for time series forecasting [17, 3]. SVM models have good accuracy over traditional statistical models and some basic artificial neural networks models for time series prediction [17]. Multi-Layered Perceptron (MLP-ANN) is most widely used ANN model for time series forecasting. It uses single hidden feed forward layer, input layer and output layer [18, 19]. The perceptron ANN uses backpropagation algorithm for error minimization [18, 19]. Time Lagged ANN (TL-ANN) is modified version of MLP-ANN that uses time lagged values as input for prediction of output values [19, 20A TL-ANN model has a number of input neurons equal to time lagged values of the present input, on which it depends [19, 20]. Seasonal ANN (SANN) a model proposed by C. Hamzacebi to improve the forecasting performance of ANNs for seasonal time series data [20]. The proposed SANN model does not require any preprocessing of raw data [20]. In addition, SANN can learn the seasonal pattern in the series, without removing them, contrary to some other traditional approaches, such as SARIMA [20].

Recurrent Neural Networks are based on David Rumelhart's work in 1986 [5, 6, 7]. RNNs can use their internal state (memory) to process variable length sequences of inputs [5, 6, 7]. The ability of RNN to retain information (dependency) played a vital role for its use in time series forecasting [7, 8]. Hochreiter and Schmidhuber invented Long Short-Term Memory (LSTM) networks in 1997 [10]. The LSTM model solve the problem of vanishing and exploding gradient while training RNN [10, 11]. The LSTM models are capable of learning long-term dependency, which made the useful for forecasting data based on previous values [10, 11, 12].

Peephole connection a popular LSTM variant was proposed by Schmidhuber and Gers in 2000 [21]. The model is capable of learning the fine distinctions of spikes separated by 49 or 50 time steps without the help of short training examples [21]. Gated Recurrent Units (GRUs) was introduced in 2014 which merged the forget gate and input gate of LSTM into a single update gate [22]. GRU has less parameters than LSTM, which makes it simpler and easier for training [23, 24, 25]. A traditional LSTM only has one hidden layer, which limits the performance of the LSTM on feature extraction so the use of more than one LSTM layers improve the performance [26, 27, 28]. The Stacked LSTMs use more than one LSTM layers for better feature extraction and performance [27, 28]. Hybrid models like Convolution LSTM are also used for time series predictions [29]. The model combines features of convolution neural networks and LSTM for a better and robust model [29]. LSTM and their variants are used in different areas of time series forecasting like weather forecast, financial time series forecasting, industrial production forecasting, product price forecasting flow prediction etc. [12, 25, 29, 30, 31, 32, 33, 34].

3. The LSTM model

Long Short-Term Memory networks are special types of recurrent neural networks introduced by Hochreiter and Schmidhuber in 1997 as a solution to problems that limited simple RNN [10, 11]. LSTMs became popular as it achieved high accuracy and precision in tasks such as speech recognition, handwriting recognition and time series forecasting [35]. The success of LSTM networks is due to the ability of retaining information (dependency) for a long period of time [10]. This is possible due to the cell state feature of LSTM that decides what information to store and forget.

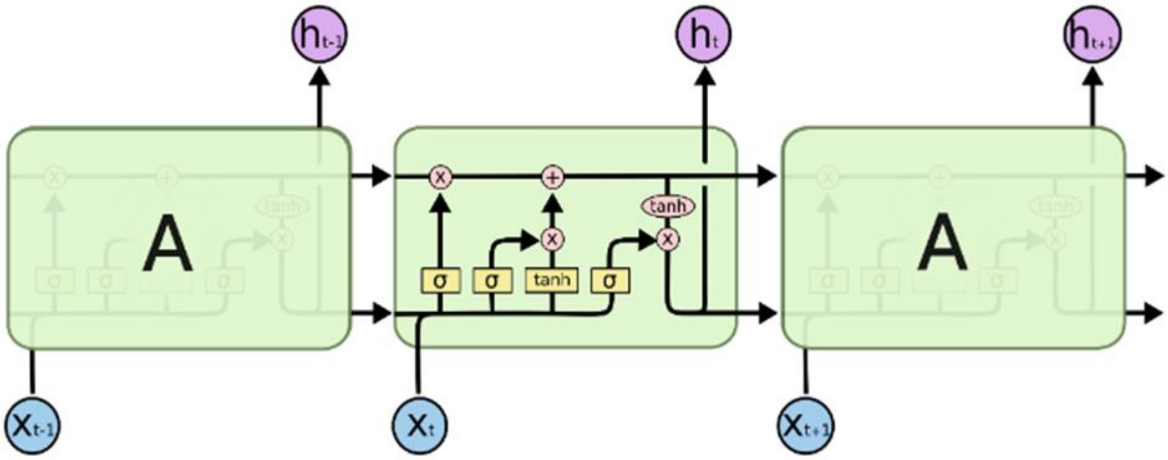


Figure 3. 1 Structure of LSTM model (source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

The LSTM network is made up of recurrent blocks having a chain like structure sharing information among them. Each LSTM block receives cell state of previous block, output of previous block and the input data as inputs. The working of the LSTM cell revolves around updating cell state, C_t. Each cell consists of three pointwise linear multiplication operation termed as gates. The forget gate decides how much information from previous cell should be retained. The input gate decides what part of input should be stored and the output gate processes cell information and input data to give output. Each block holds information in its own cell state that can interpret dependency of previous cell with current cell. The output from output gate is transferred to next LSTM block and other networks lying beneath LSTM layer to produce a final output.

3.1. Components of LSTM cell

i. The cell state

The cell state is conveyor belt like structure, which runs through entire cells of LSTM network undergoing some minor linear operations in each cell. The cell state is used to pass information between cells and store information by each cell.

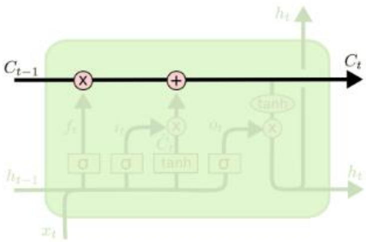


Figure 3.1. 1 the cell state of LSTM (source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

ii. Gates

Gates are the structure in LSTM unit that allows the update on the cell state. They are composed of sigmoid neural net layer and pointwise linear multiplication operation. The output of sigmoid layer is in between zero and one, describing how much of each component should be let through. A value of zero denies the entry of component completely and a value of one lets the component to pass through completely.

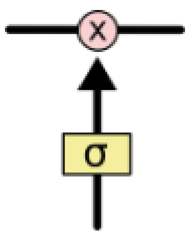


Figure 3.1. 2 gate structure of LSTM (source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

There are three types of gates in LSTM cell. They are:

a. Forget gate

The forget gate decides what information should be thrown away from the previous cell state. It combines the input x_t and h_{t-1} using the sigmoid function.

A value of zero erases the information on cell state completely and a value of one keeps the cell state unchanged.

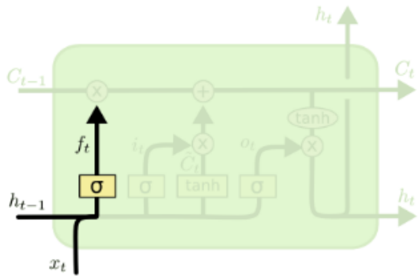


Figure 3.1. 3 Forget gate (source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

The cell information retained depends on f_t, which is given by the formula

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

Where, σ is the sigmoid function, W_f is the weight of forget gate, h_{t-1} is the output of previous cell.

b. Input gate

The input gate adds new information to the cell state C_t. It involves two major steps. First, the sigmoid layer decides what values we will update based on x_t and h_{t-1}. The Tanh layer creates a new candidate value of C_t that will be added to the cell state.

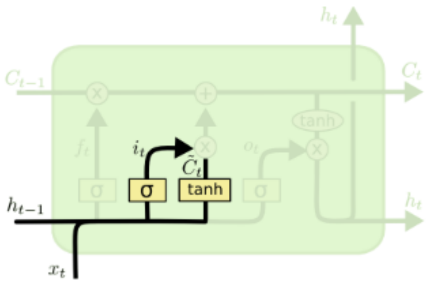


Figure 3.1. 4 Input gate (source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Here,

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

$$\hat{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c)$$

Where, W_i and b_i are weight and bias of input layer and W_c and b_c are weights and bias of cell state.

The next step is to combine the output of forget gate and input gate to create a new cell state C_t.

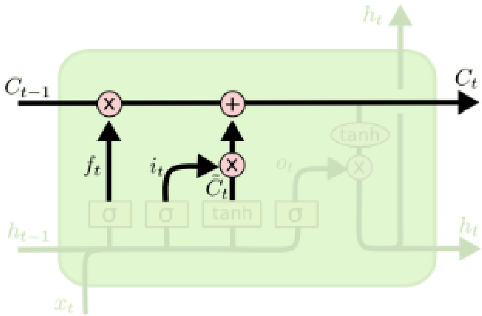


Figure 3.1. 5 the update on cell state (source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

The final cell state is given by the formula

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

c. Output gate

The output gate gives the output of a LSTM cell. The output is determined by the values of cell state, input data and output of previous cell. First, a sigmoid layer decides what part of input is to be passed as output. A Tanh layers truncates cell state C_t between -1 and 1 and the value is multiplied with output of sigmoid layer to produce a final output h_t.

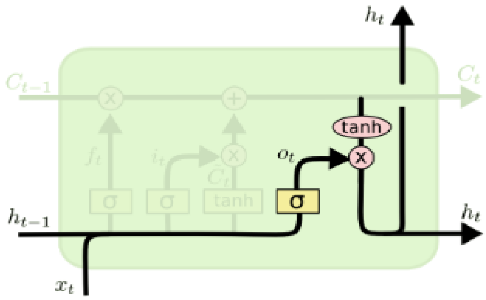


Figure 3.1. 6 output gate (source:https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

The output is given by the equation

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

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

W_0 and b_0 are weights and bias of output gate.

3.2. LSTM variants

Due to success and popularity of LSTM in different applications, many variants of it have been proposed and studied. The paper discusses only four of its variants, which are widely used for time series forecasting. They are:

a) Vanilla LSTM

A Vanilla LSTM or simple LSTM consists of a single hidden layer of LSTM units and an output layer used to make a prediction. It is simplest variant of LSTM and is widely used for learning activities with few features. Vanilla LSTM is used as a benchmark for comparison with other LSTM models [36]. The vanilla LSTM contains a sigmoid output layer. The structure of Vanilla LSTM model is shown below.

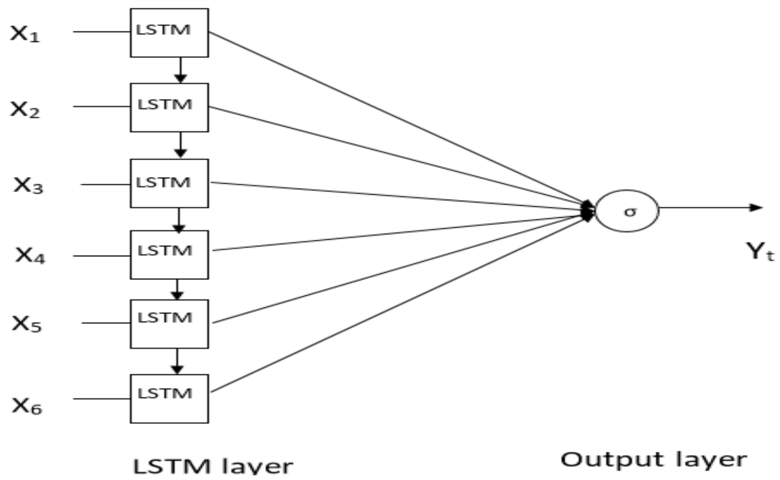


Figure 3.2. 1 Vanilla LSTM

b) Stacked LSTM

Stacked LSTM uses stacking of LSTM layers to improves feature extraction and learn long-term dependencies efficiently [27, 28]. However, the stacking of LSTM layers comes with the price of computational complexity as the number of parameters increase significantly with increase of LSTM layers [28]. The model used in this paper has two LSTM layers followed by fully connected layers and a single sigmoid output layer.

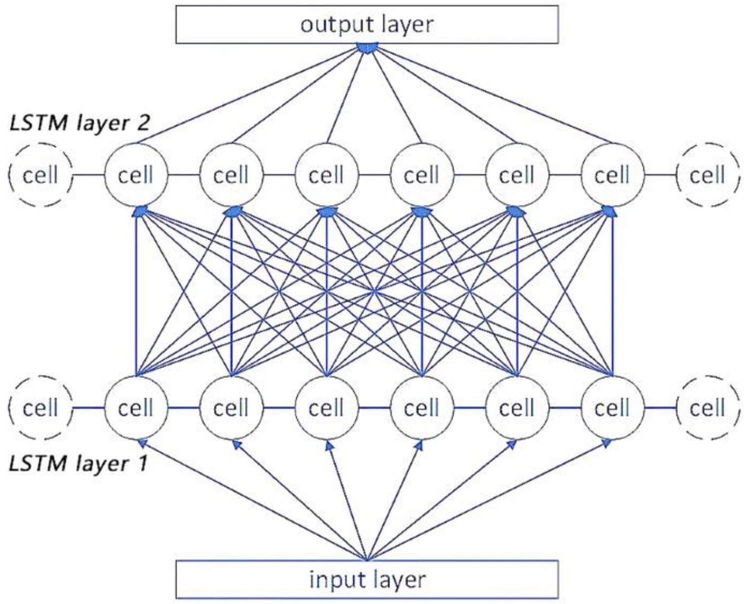
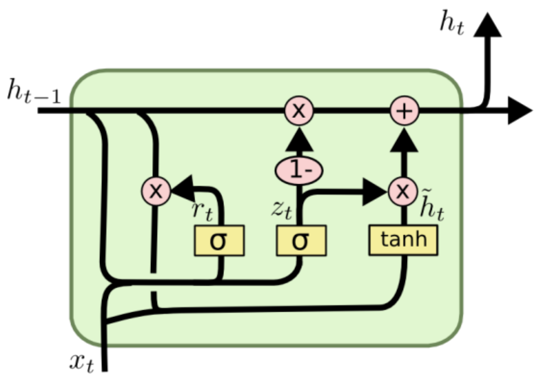


Figure 3.2. 2 Stacked LSTM structure [27]

c) Gated Recurrent Units

Gated Recurrent Units was introduced in 2014 [21]. The model combines forget and input gates into a single update gate and merges the cell state and hidden state [23]. Hence, there is a single connection between adjacent units in GRU layer. The resulting model is simpler than standard LSTM models, and has been growing increasingly popular [24, 25]. Due to less training parameters, the training of GRU is faster than LSTM [25].



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 3.2. 3 GRU structure (source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

d) Convolution Neural Network-LSTM (CNN LSTM)

Convolutional layers are characterized by their ability to extract useful knowledge and learn the internal representation of time-series data, while LSTM networks are effective for identifying short-term and long-term dependencies [29]. The principle idea of this model is to combine the advantages of these deep leaning techniques [29]. CNN LSTM consists of two main components: The first component consists of convolutional and pooling layers in which complicated mathematical operations are performed to develop features of the input data, while the second component exploits the generated features by LSTM and dense layers [29].

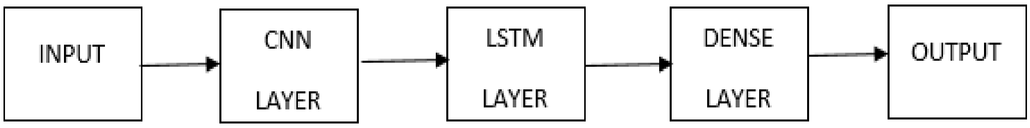


Figure 3.2. 4 Architecture of CNN LSTM model

3.3. Training LSTM models

The training of LSTM model involves minimization of cost function by use of backpropagation algorithm with optimization technique for updating weights parameters in the network [10]. The loss metrics used in this paper is mean squared error (MSE). MSE is the sum of squared distances between our target variable and predicted values.

$$MSE = \frac{\sum_{i=1}^{N}(y_i - y_i^p)^2}{N}$$

Where, y_i is the actual value and y_i^p is the predicted value. The optimization technique used in this paper is Adaptive Moment Estimation (Adam). Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [37]. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters [37]. Adam is widely used as optimization algorithm for LSTM training. The learning rate used in the models is 0.001. The activation function used for the models is Rectified Linear Unit (ReLU).

4. Experiments and results

The paper explores two different forecasting problems to determine effectives and compare the LSTM variants. The first experiment uses gold price dataset provided by World Gold Council in their website [13]. The data contains price of gold per Troy ounce (approx. 31.10 grams) in different major currencies used around the world. The price of gold used in this paper is U.S dollar. The dataset contains gold price from 1 January 1979 to 31 July 2020. The second dataset is the Brent oil price dataset made available by U.S Energy Information Administration [14]. The data contains daily Brent oil prices of crude oil per barrel from 17th of May 1987 until July 31, 2020. Experiment 2 is based on this dataset. The snapshot of the data is shown below in the figure. The experiments were performed in Jupyter notebook using Python 3.6 in windows operating system. The models were developed using Keras API with Tensorflow as back-end.

Date	US dollar	Euro	Japanese yen	Pound sterling	Canadian dollar	Swiss franc	Indian rupee	Chinese renmimbi	US dollar	Turkish lira	Saudi riyal	Indonesian rupiah
5/26/2020	1,720.3	1,566.1	185,090.3	1,392.0	2,374.6	1,664.7	130,154.1	12,274.8	1,720.3	11,556.2	6,463.1	25,382,288.8
5/27/2020	1,694.6	1,544.3	182,737.2	1,387.5	2,341.9	1,643.7	128,304.5	12,123.5	1,694.6	11,484.3	6,365.8	24,927,567.4
5/28/2020	1,717.4	1,553.4	184,829.8	1,394.0	2,363.2	1,658.4	130,094.7	12,294.9	1,717.4	11,708.0	6,447.8	25,308,588.4
5/29/2020	1,728.7	1,554.1	186,241.5	1,398.3	2,389.4	1,660.8	130,717.8	12,352.6	1,728.7	11,792.3	6,495.6	25,256,306.3
6/1/2020	1,730.6	1,553.9	186,134.7	1,386.4	2,357.1	1,662.2	130,738.2	12,347.7	1,730.6	11,788.0	6,495.8	25,284,067.4
6/2/2020	1,742.2	1,557.9	189,171.4	1,387.4	2,349.7	1,673.5	131,292.8	12,387.6	1,742.2	11,729.0	6,539.2	25,143,578.5
6/3/2020	1,705.4	1,520.0	185,584.7	1,353.8	2,300.3	1,640.6	128,702.8	12,124.9	1,705.4	11,476.6	6,403.5	24,062,489.9
6/4/2020	1,700.1	1,498.6	185,314.0	1,348.4	2,292.7	1,627.0	128,485.5	12,108.3	1,700.1	11,474.9	6,383.7	23,979,205.9
6/5/2020	1,683.5	1,488.0	184,750.2	1,322.9	2,257.4	1,620.2	127,241.5	11,929.9	1,683.5	11,418.4	6,321.4	23,374,702.6
6/8/2020	1,690.4	1,495.8	183,546.7	1,331.2	2,263.0	1,616.0	127,678.5	11,959.6	1,690.4	11,480.9	6,343.9	23,495,866.4
6/9/2020	1,713.5	1,509.1	184,569.7	1,346.8	2,300.4	1,626.3	129,566.3	12,144.3	1,713.5	11,650.9	6,429.1	23,800,515.0
6/10/2020	1,722.1	1,517.0	184,681.3	1,350.1	2,312.7	1,628.3	130,182.7	12,169.7	1,722.1	11,702.2	6,462.0	24,074,259.7
6/11/2020	1,738.3	1,527.3	185,375.7	1,374.9	2,355.0	1,632.5	131,744.1	12,289.1	1,738.3	11,865.3	6,521.9	24,361,573.8
6/12/2020	1,733.5	1,541.4	186,056.6	1,382.4	2,359.1	1,649.0	131,490.3	12,263.1	1,733.5	11,853.8	6,506.7	24,555,027.5
6/15/2020	1,710.5	1,517.1	183,710.9	1,361.1	2,330.1	1,624.8	130,042.4	12,134.8	1,710.5	11,713.2	6,417.6	24,177,210.1
6/16/2020	1,719.9	1,530.5	184,651.7	1,368.5	2,341.0	1,636.0	131,061.2	12,179.6	1,719.9	11,777.1	6,452.0	24,258,485.6
6/17/2020	1,724.4	1,536.5	184,927.9	1,375.1	2,337.3	1,642.4	131,330.8	12,217.2	1,724.4	11,824.7	6,468.4	24,283,160.3

Figure 4. 1 Snapshot of gold price dataset

Back to Contents	Data 1: Europe Brent Spot Price FOB (Dollars per Barrel)
Sourcekey	RBRTE
Date	Europe Brent Spot Price FOB (Dollars per Barrel)
Apr 09, 2020	20.23
Apr 14, 2020	21.74
Apr 15, 2020	19.8
Apr 16, 2020	18.69
Apr 17, 2020	19.75
Apr 20, 2020	17.36
Apr 21, 2020	9.12
Apr 22, 2020	13.77
Apr 23, 2020	15.06
Apr 24, 2020	15.87
Apr 27, 2020	15.17
Apr 28, 2020	15.6
Apr 29, 2020	17.86
Apr 30, 2020	18.11
May 01, 2020	18.49
May 04, 2020	20.4
May 05, 2020	25.46
May 06, 2020	24.2
May 07, 2020	24.23
May 11, 2020	25.53
May 12, 2020	26.67
May 13, 2020	27.89

Figure 4. 2 Snapshot of Brent oil price dataset

4.1.Model settings

The different parameters used in the experiment is shown below in the table 4.1.1. The rectified linear unit was used as activation function for LSTM units and fully connected layers. The output layer is sigmoid as the data is normalized between 0 and 1. A dropout value of 0.2 was used to avoid overfitting of the models during training. Dropout is a regularization technique that allows network to avoid certain portion of its neurons during training [38].

The settings were obtained by iterative approach i.e. various settings for model parameters were tested to obtain good results. However, the model settings of GRU and Vanilla LSTM were kept same to compare the effectiveness of these two models. The stacking of LSTM was limited to two because the over stacking of LSTM layers increased the parameters significantly which caused difficulty in training. Similarly, in CNN LSTM model one of each layers was used to develop the hybrid model. The fully connected layers were adjusted by iterative approach to achieve best possible results. Regularization mechanism and train-test loss graphs were used to prevent the overfitting of the models.

Table 4.1. 1 parameters setting for different models

Parameters	Vanilla LSTM	Stacked LSTM	GRU	CNN LSTM
LSTM layers	1	2	-	1
Activation function	RELU	RELU	RELU	RELU
Number of fully connected layers	1	2	1	2
GRU layers	-	-	1	-
CNN layers	-	-	-	1
Output layer	Sigmoid	Sigmoid	Sigmoid	Sigmoid
Batch size	16	16	16	16
Dropout	0.2	0.2	0.2	0.2
Lookback	8	8	8	8
Loss function	MSE	MSE	MSE	MSE
Optimizer	Adam	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001	0.001
Epochs	10	10	10	10

4.2.Experiment 1: Forecasting Gold price in US dollar

In this experiment, the gold price dataset was used to train different models to obtain two forecast values, one day ahead forecast values and month ahead forecast values. The visualization of the dataset is shown in figure 4.2.1. The graph shows daily data, one-year moving average and five-year moving average plots of gold. The one-year and five year moving average plot show a steady rise of gold price in 2005-2013 and 2019-2020. There is significant decrease of price between 2013 and 2016. The current trend in the data can be seen as increasing as the price of gold sets new records unattained before. The plot shows that the decrease in gold price is very rare as there are only few significant drops in price of gold over the course of time. The gold price is well described by one-year moving average plot.

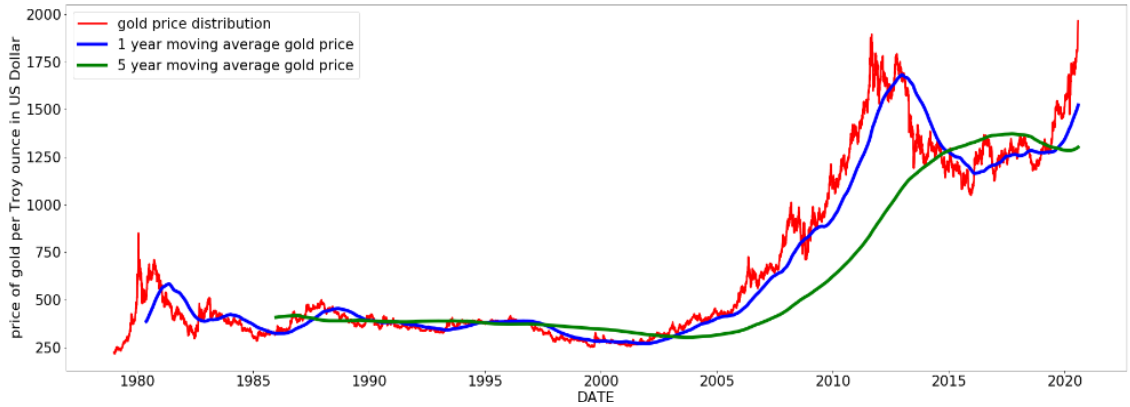


Figure 4.2. 1 visualization of gold price (U.S dollar) dataset

The result of prediction of one-month ahead plots are shown in figure 4.2.2. In this test, the gold price for entire month of July was forecasted using model trained and validated in historic data. The plot shows that the predictions for all models are quite good. The RMSE values for Vanilla LSTM, Stacked LSTM, GRU and CNN LSTM are observed as 34.61, 30.07, 37.11 and 28.97 respectively.

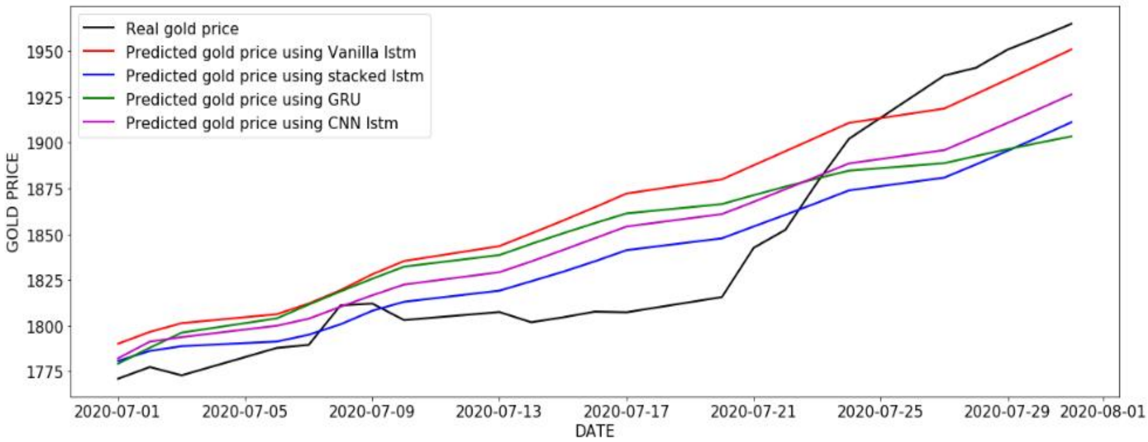


Figure 4.2. 2 visualization of one-month ahead prediction of gold price for month of July 2020

Figure 4.2.3 shows the one-day ahead prediction plots of the four different models. In this prediction real values of gold price a day before were used to predict the price of current day. These predictions are more accurate than one-month ahead predictions because the predictions term is very short. Similar to the one-month ahead predictions all four models were able to make good predictions. However, GRU model outperformed other models as the predicted values and trend matches quite well with the actual values. The RMSE values or Vanilla LSTM, Stacked LSTM, GRU and CNN LSTM are observed as 27.52, 30.29, 13.85 and 20.46 respectively.

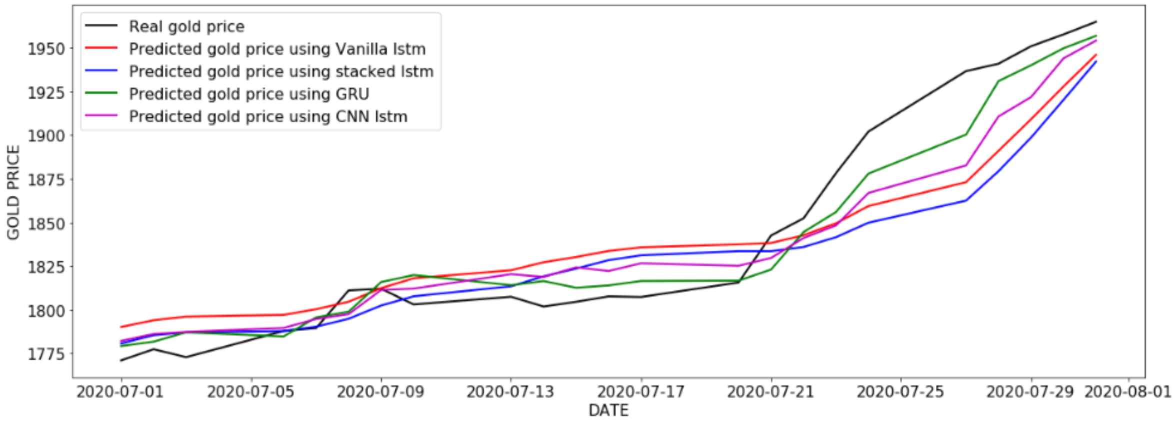


Figure 4.2. 3 visualization of one-day ahead predictions of gold price for July 2020.

4.3.Experiment 2: Forecasting Brent oil price per barrel in US dollar

This experiment is based on the gold price dataset, which was used to train different models to obtain two forecast values, one day ahead forecast values and one month ahead forecast values. The visualization of the dataset is shown in figure 4.3.1. The graph shows daily data, one-year moving average and five-year moving average plots of gold. The one-year and five year moving average plots unlike in gold dataset are not able to represent the actual data. The graph shows numerous heavy increase and decrease of prices. The current trend in the data can be seen as increasing as the price of oil recovers from fall of global pandemic Covid-19.

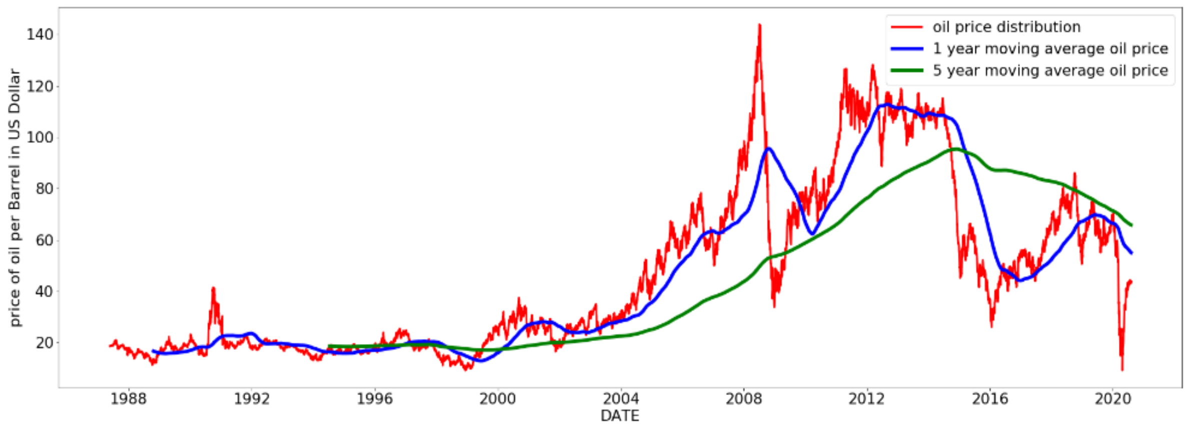


Figure 4.3 1 Visualization of Brent oil price per barrel in US dollar

The result of prediction of one-month ahead plots of different models are shown below in figure 4.3.2. The price of oil for entire month of July was forecasted using model trained and validated in historic data. The complex models (Stacked LSTM and CNN LSTM) predicted the price more accurately than simpler models. The RMSE values for Vanilla LSTM, Stacked LSTM, GRU and CNN-LSTM are observed as 4.76, 0.556, 8.54 and 0.522 respectively.

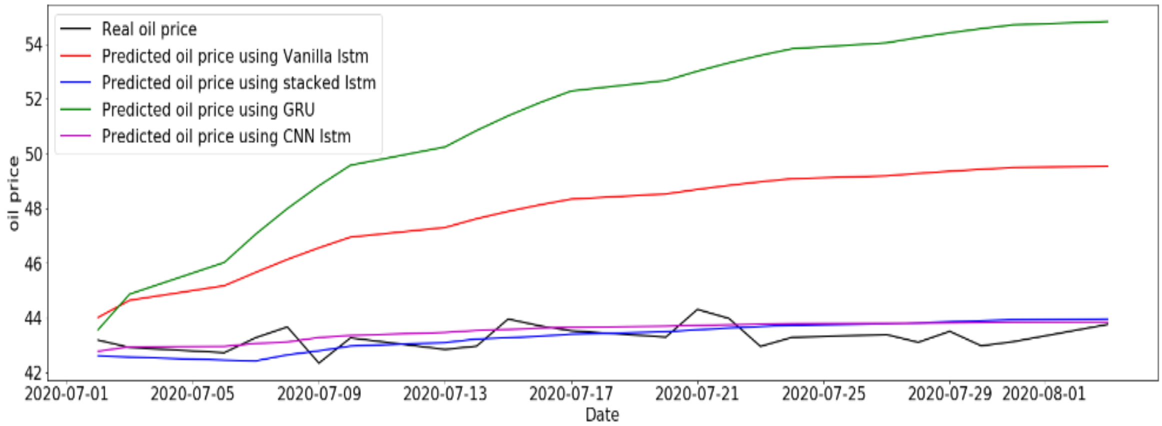


Figure 4.3 2 visualization of one-month ahead forecasted oil price values for month of July 2020

Similar to experiment 1, the models developed were used to make one-day ahead predictions. The values obtained from different models are plotted in figure 4.3.3. It is clear from the graph that CNN LSTM and Stacked LSTM models predictions are better than GRU and Vanilla LSTM models. The predictions of complex models are similar with the one-month ahead predictions, which shows that these models capture the stochastic dependencies of historic data effectively. The RMSE values for Vanilla LSTM, Stacked LSTM, GRU and CNN LSTM are observed as 1.92, 0.551, 1.755 and 0.539 respectively.

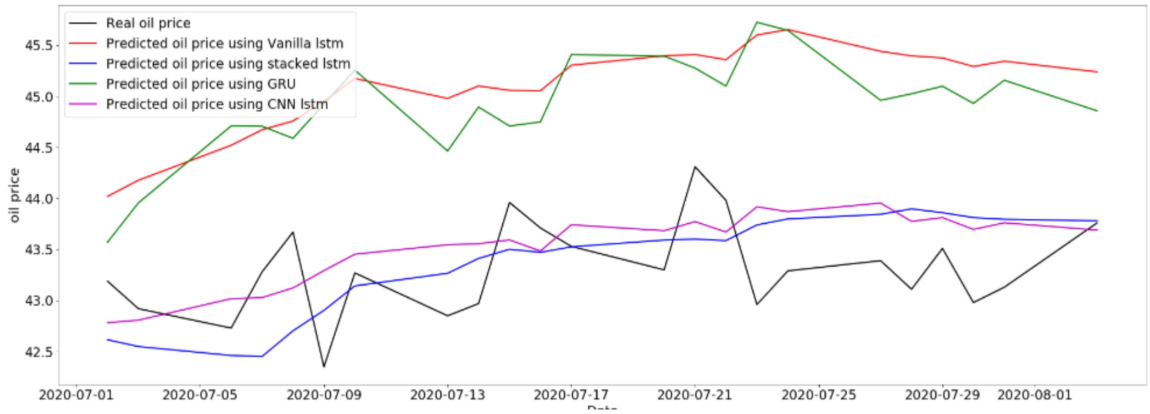


Figure 4.3 3 visualization of one-day ahead forecast oil price values for month of July 2020

5. Analysis

The graphs obtained from the two experiments performed above shows that all the LSTM variants are suitable for time series forecasting tasks. In the first experiment all models predictions were good as shown in figure and figure. The reason for this result is due to the nature of dataset as the price of gold is in increasing trend in most of the parts of dataset as shown in figure, which makes the prediction problem less complex so the obtained results are better. However, the second dataset contains numerous rise and fall trends, which makes the predictions problem more complex and learning difficult. The RMSE values for different models for two experiments are shown below in table.

Table 5. 1 RMSE values for models obtained in experiments

MODELS	Gold price forecast		Oil price forecast	
	One-month ahead predictions	One-day ahead predictions	One-month ahead predictions	One-day ahead predictions
Vanilla LSTM	34.61	27.52	4.76	1.92
Stacked LSTM	30.07	30.29	0.556	0.551
GRU	37.11	13.85	8.54	1.755
CNN-LSTM	28.97	20.46	0.522	0.539

The predictions made by GRU are better than the predictions made by Vanilla LSTM model for both tasks as shown in table. Therefore, it can be reasoned that the gated recurrent units are better than vanilla LSTM models for time series forecasting. In addition, the training time for GRU was less as compared to the Vanilla LSTM model. The RMSE values for CNN LSTM are better for all predictions than Stacked LSTM models but are similar. The graphs shown above also show close predictions of these models. Hence, the comparison between these two models is vague based on only these experiments. However, the result show that the stacked LSTM and CNN LSTM models perform better than Vanilla LSTM and GRU models and are suitable for time series forecasting. It can be inferred from results that the LSTM models are better suited for short term predictions as one day ahead forecast values are more accurate than one month ahead forecast values. The obtained results suggests that the stacking of LSTM models and use of hybrid models like CNN LSTM helps to improve accuracy of long term forecast.

6. CONCLUSION AND RECOMMENDATIONS

In this paper, four different popular variants of Long Short-Term Memory models are evaluated in terms of their ability to perform time series forecasting based on two different datasets containing historical records. The models developed for these experiments were based on iterative approach involving parameters tuning, hyper parameters tuning and regularization technique. All four model Vanilla LSTM, Stacked LSTM, GRU and CNN LSTM were able to perform the prediction task successfully. However, the stacked LSTM and convolution LSTM are better than other two models for time series forecasting. The two type of forecast (one-day ahead forecast and one-month ahead forecast) results show that these models perform better if they are used for short-term predictions.

The paper is based on only four LSTM variants out of many different models used for time series forecasting. It is recommended to study and use those models for comparison. The Stacked LSTM and CNN LSTM use only few layers of LSTM layers so, the effectiveness of heavy stacking remains unexplored in this paper. The results on this paper are based on only two experiments and are preliminary thus more experiments should be done to obtain concrete results and conclusions.

7. REFERENCES

[1] G. E. P. Box, *Time series analysis: forecasting and control*. Hoboken N.J.: Wiley, 2016.

[2] D. C. Montgomery, M. Kulahci, and C. L. Jennings, *Introduction to time series analysis and forecasting*. Hoboken, NJ: Wiley, 2016.

[3] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[4] T. Y. Kim, K. J. Oh, C. Kim, and J. D. Do, "Artificial neural networks for non-stationary time series," *Neurocomputing*, vol. 61, pp. 439–447, 2004.

[5] A. C. Tsoi and A. Back, "Discrete time recurrent neural network architectures: A unifying review," *Neurocomputing*, vol. 15, no. 3-4, pp. 183–223, 1997.

[6] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[7] J. Connor, R. Martin, and L. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.

[8] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," Physica D: Nonlinear Phenomena, vol. 404, pp. 132306–132349, Mar. 2020.

[9] Informatik, Fakultit & Bengio, Y. & Frasconi, Paolo & Schmidhuber, "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies", *A Field Guide to Dynamical Recurrent Neural Networks*, 2003.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[12] Elsworth, Steven and Stefan Güttel. "Time Series Forecasting Using LSTM Networks: A Symbolic Approach." *ArXiv*, vol. abs/2003.05672, 2020.

[13] World Gold Council, “Gold Prices,” Retrieved from https://www.gold.org/goldhub/data/gold-prices.

[14] U.S. Department of Energy, Energy Information Administration, Independent Statistics & Analysis, “Europe Brent Spot Price FOB” Retrieved from https://www.eia.gov/dnav/pet/.

[15] R. G. Brown and R. F. Meyer, “The Fundamental Theorem of Exponential Smoothing,” *Operations Research*, vol. 9, no. 5, pp. 673–685, 1961.

[16] A. A. Weiss, “ARMA Models With Arch Errors,” *Journal of Time Series Analysis*, vol. 5, no. 2, pp. 129–143, 1984.

[17] K.-J. Kim, “Financial time series forecasting using support vector machines,” *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[18] O. Rudenko, O. Bezsonov, and O. Romanyk, “Neural network time series prediction based on multilayer perceptron,” *Development Management*, vol. 17, no. 1, pp. 23–34, 2019.

[19] Adhikari, Ratnadip & Agrawal, R., “Forecasting Strong Seasonal Time Series with Artificial Neural Networks”. Journal of scientific and industrial research, vol. 71, 2012.

[20] J. P. Donate, P. Cortez, A. S. D. Miguel, and G. G. Sanchez, “Evolving time-lagged feedforward neural networks for time series forecasting,” *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation - GECCO 11*, 2011.

[21] F. Gers and J. Schmidhuber, “Recurrent nets that time and count,” Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, 2000.

[22] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[23] “Understanding LSTM Networks,” *Understanding LSTM Networks -- Colah’s blog*. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 18-Aug-2020].

[24] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y., "Empirical evaluation of gated recurrent neural networks on sequence modeling". NIPS 2014 Workshop on Deep Learning, December 2014, 2014.

[25] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2016.

[26] Ben Krause, Liang Lu, Iain Murray, Steve Renals, "Multiplicative LSTM for sequence modelling," *CoRR*, abs/1609.07959, 2016

[27] L. Sun, Y. Wang, J. He, H. Li, D. Peng, and Y. Wang, "A stacked LSTM for atrial fibrillation prediction based on multivariate ECGs," *Health Information Science and Systems*, vol. 8, no. 1, 2020.

[28] C. Dyer, M. Ballesteros, Wang Ling, A. Matthews and Noah A. Smith, "Transition-Based Dependency Parsing with Stack Long Short-Term Memory", *CoRR*, abs/1505.08075, 2015.

[29] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Y25eung, Wai-kin Wong, and Wang-chun Woo, "Convolutional LSTM Network: a machine learning approach for precipitation nowcasting". In Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 1, pp. 802–810, 2015.

[30] B. B. Sahoo, R. Jha, A. Singh, and D. Kumar, "Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting," *Acta Geophysica*, vol. 67, no. 5, pp. 1471–1481, 2019.

[31] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, vol. 148, pp. 461–468, 2018.

[32] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[33] Y.-X. Wu, Q.-B. Wu, and J.-Q. Zhu, "Improved EEMD-based crude oil price forecasting using LSTM networks," Physica A: Statistical Mechanics and its Applications, vol. 516, pp. 114–124, 2019.

[34] sJ. Cao, Z. Li, and J. Li, “Financial time series forecasting model based on CEEMDAN and LSTM,” *Physica A: Statistical Mechanics and its Applications*, vol. 519, pp. 127–139, 2019.

[35] “Long short-term memory,” Wikipedia, 01-Jul-2020. [Online]. Available: https://en.wikipedia.org/wiki/Long_short-term_memory. [Accessed: 02-Aug-2020].

[36] G. V. Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artificial Intelligence Review*, 2020.

[37] Kingma, D.P., & Ba, J., “Adam: A Method for Stochastic Optimization,” *CoRR*, abs/1412.6980, 2015.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn*, Res. 15, pp. 1929–1958, January 2014.