University of Essex

Department of Mathematical Sciences

PS981-7-FY: Dissertation

Future gold prices forecasting by LSTM, ARIMA, Prophet

Registration number: 1908015

Supervisor: Andrew P. Harrison

Date of submission(28 August 2020)

Word count: 5026

Contents

Abstract.

Gold is a commodity or currency that most investors and financial institutions give an attention when the economic system is in crisis or paper currencies have less values. Holding gold by Central Bank of each country can imply the financial strength of those countries. The more gold the country hold, the more power of re-payment of foreign debts, and control inflation. Then, knowing the direction of future gold price beforehand can provide number of benefits to investors. Hence, the aim of this paper is to forecast future gold price by using deep learning model like Long short term memory (LSTM) and compare to linear model such as Facebook Prophet model, and Auto-regressive Integrated Moving Average (ARIMA) model. We used a sliding window approach for Long short term memory (LSTM) and the window size is fixed for 3 days for prediction of next day. Various related topic papers and Background of three different algorithms were studied in this research. Root Mean Squared Error (RMSE) and Mean absolute Error (MAE) were deployed to measure the performance of each algorithm. The result in this research showed that Long short term memory (LSTM) algorithm have the highest accuracy in prediction and are capable of capturing the volatility change of daily gold price while other two linear model cannot capture it. Hence, Long short term memory (LSTM) model was used to forecast future gold price.

Keywords: gold price forecast, long short term memory (LSTM), auto-regressive integrated moving average (ARIMA), Facebook Prophet model

1 Introduction

Gold is a kind of precious commodities which most financial institutions and investors have highly attention to it. In the past, most countries including the United Stated had used gold as one kind of currency form [1]. In recent times, gold still plays a significant role in economic and monetary system. And gold values still are rising continuously and never-ending. Gold is totally different from shares or bonds since gold cannot generate any cash flows or dividends like shares and bonds. ”Gold is more currency than commodity since gold values has more to do with its longstanding function as a store of value, especially during crises or when you lose faith in paper currencies” [2]. When there is a crisis in economic system, amount of capital tend to move into gold. Then, gold values increase and other assets decrease. There are various propose in investment of gold for instance hedging against decline of stock market, inflation, profit capital gain, diversifying risk of portfolio, and maintaining the values of money. Since to know the future gold price beforehand can make various advantages for investors and financial institutions.

This study will help investors and financial institutions, who tend to invest in gold, to make a correct decision when to buy and to sell gold and reducing the risk of investment in gold and having the better management of diversifying the risk of the mixed portfolio.

The main aim of this paper are to analyse the capability of Long short term memory (LSTM) model whether it can predict future gold price accurately then comparing to traditional time series analysis model such as Auto-regressive Integrated Moving Average (ARIMA) and Prophet model which established by data scientist team of Facebook company. And to understand deeply how the architecture of three different algorithms (Long short term memory (LSTM), Auto-regressive Integrated Moving Average (ARIMA), Prophet model) does work.

This paper composed of 6 sections, the structure is organised as following; section 2 describes literature review, section 3 describes background of each algorithm, in section 4 we defines methodology of data set, Long short term memory (LSTM), Prophet model, and Auto-regressive Integrated Moving Average (ARIMA) model, section 5 shows the results of experiment, and discussion in section 6.

2 Literature review

In this section, we will talk about past papers of gold price forecasting using different algorithms. To begin with paper [1], the authors employed Artificial Neural Networks (ANN) and Linear Regression (LR) models for predicting the future gold price. Many independent variables were used such as precious commodities (gold prices, silver prices, platinum prices, etc.), stock market index (S&P500 Index, NYSE Index), related companies' stock values (Silver Wheaton Corporation (SLW), Eldorado Gold Corporation, and Compania de Minas Buenaventura), Interest rate of China, the United Kingdom, the United States, and Russia. They used the most recent 25% of data set for testing set and the rest 75% for training set. Furthermore, they split the training set into four versions. For the first version, it split 0 to 75 % for training set. The second version is 15-75% for training set, Next, 30 to 75% for training set, and the last version contains 45-75% for training set. The result of this paper is that smaller data set (45-75% of training set) can lead to better performance for both Artificial Neural Networks (ANN) and Linear regression (LR). Next, paper [3] again used Artificial Neural Networks (ANN) and Auto-regressive Integrated Moving Average (ARIMA) for predicting the future gold price values. The observations is 220 months of gold prices between April 1990 and July 2008. The training set and testing set are 200 and 20 months respectively. There are seven indicators were used (USD Index (measures the performance of the United States Dollar against the Canadian Dollar), inflation rate (the United States inflation rates), oil price (West Texas Intermediate Crude Oil Prices), interest rate (the United States interest rates), stock market index (Dow Jones Industrial Average), silver price, and world gold production). The best ARIMA model is ARIMA(1,1,0) and For ANN is contained of seven indicators, twenty four hidden and one output neurons. Hence, the performance of ANN is better than ARIMA. Both paper [4] and [5] the authors employed Auto-regressive Integrated Moving Average (ARIMA) to predict the future gold price. Both papers used different methods and different period of data set. To begin with paper [5], Augmented Dickey-Fuller test statistic (ADF) was employed to test whether the time series is stationary. Auto-correlation function (ACF) and the partial auto-correlation function (PACF) were applied since it has shown that there is no significant tailing and truncation, AIC and BIC were used instead to evaluate which model is the best. Hence, ARIMA(3,1,2) is the best model in their experiment. On the other hand, paper [4] used James Durbin and Geoffrey Watson test statistics for testing the time series whether the data is longitudinal or cross-sectional. Again, Auto-correlation function (ACF) and the partial auto-correlation function (PACF) were used to the data set. Hence, ARIMA(1,1,1) is the best performance in their experiment. In paper [6] has shown that Extreme Learning Machine (ELM) outperformed over four algorithms which are Feed forward networks without feedback, feed forward back propagation networks, Radial basis Function, and Elman Networks. They used monthly average price of five factors (gold price, crude oil, silver price, United Stated Dollars and S&P500 stock price) for Extreme Learning Machine (ELM) algorithm. The next month gold price is the output. The data set is split into training, testing, and validation for 70%, 15%, and 15% respectively. This experiment showed that the accuracy of training and testing are 97.65% and 93.82% respectively for Extreme Learning Machine (ELM). Next, the Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) was used to compare the performance of predicting the future gold price with Auto-regressive Integrated Moving Average (ARIMA) model in paper [7]. The Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) is very popular among financial modelling professionals since it can provide a more real-world context in forecasting the prices and rates of financial commodities. Data set of daily gold price is from 18 July 2001 until 25 September 2012. Since the Auto-correlation Function (ACF) and the Partial Auto-correlation Function (PACF) were test. The best-fitted model of Auto-regressive Integrated Moving Average (ARIMA) and Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) were ARIMA(1,1,1) and GARCH(1,1). Since, the AIC and Mean Absolute Percentage Error (MAPE) of Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) is lower than ARIMA model. It can be clearly said that Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) outperformed Auto-regressive Integrated Moving Average (ARIMA) in this experiment. In paper [8], the authors employed deep neural network which are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short Term Memory (LSTM). These algorithms can be applied to various field like image processing, natural language processing (NLP), time series analaysis, etc. These deep neural network were compared to linear model such as Auto-regressive Integrated Moving Average (ARIMA) in this experiment. The data set is minute wise stock price for the period of July 2014 to June 2015. The stock price consists of two different IT companies and one Pharma company. A sliding window approach was used for this experiment. The window size was 100 minutes with 90 minute of training set and 10 minute for testing set. The result of this paper is that Convolutional Neural Networks (CNNs) outperformed other two deep neural network algorithms since Convolunation Neural Network (CNNs) used only the current window for prediction which is different from other two algorithms that used previous information to predict the future values. Furthermore, this paper showed that all three deep neural networks (LSTM, RNN, CNN) model outperform Auto-regressive Integrated Moving Average (ARIMA) model.

3 Background

In this section, there are the background of each algorithm architecture; Long Short Term Memory (LSTM), Facebook Prophet, Auto-regressive Integrated Moving Average (ARIMA) and two most common evaluation metrics which will be used in this paper; Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

3.1 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) [8], [9] is a better version architecture of Recurrent Neural Network (RNN) introduced by Hochreiter and Schmidhuber in 1997 [10]. LSTM and RNN are supervised machine learning models which are the most popular algorithms for use of sequential information such as time-series forecasting. These two algorithms assume that the outputs are dependent on the previous inputs. These algorithms can use their internal state or memory to process sequences of inputs. To begin with, Recurrent Neural Network (RNN) basically has three layers which are input, recurrent hidden state, and output layers as shown in figure 1. The input layer has T input units and the input vector can be written as: $x_t = (x_1, x_2, x_3, ..., x_T)$, the hidden layer has T hidden units which can be written in vector as: $h_t = (h_1, h_2, h_3, ..., h_T)$ and the output layer also has T output units written as: $o_t = (o_1, o_2, o_3, ..., o_T)$. The formula for the hidden layer can be written as:

$$h_t = f_H(W_{hh}h_{t-1} + W_{xh}x_t + b_h), \tag{3.1}$$

$f_H(\cdot)$ is the hidden layer activation functions which is tanh in this formula, W_{hh} is the weight at previous hidden state, h_{t-1} is the previous hidden state, W_{xh} is the weight at current input state and b_h is the bias vector of the hidden units. The hidden units are connected to the output units (o_t) with weighted connections (W_{oh}) can be stated as:

$$o_t = f_O(W_{oh}h_t + b_o) \tag{3.2}$$

$f_O(\cdot)$ is the activation function, W_{oh} is the weight at the output state, b_o is the bias vector of the output units and o_t is the output state. For Recurrent Neural Network (RRN), it uses Backpropogate through time to update the weights by going back in every time stamp. Since RNNs is the model for sequence information, when the length of data dependencies increases, RNNs would be suffer extremely from the gradient vanishing problem [11]. In other words, RNN cannot remember the long previous data. However, ***Long Short Term Memory*** (LSTM) can

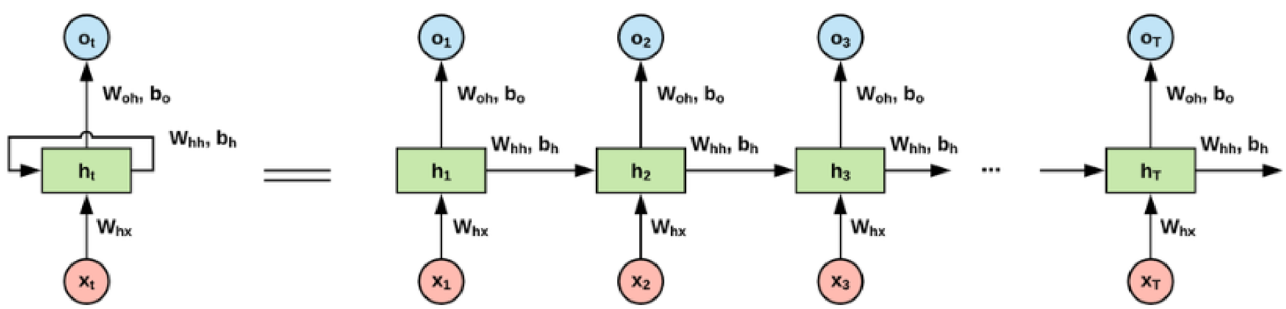

Figure 1: Recurrent Neural Network architecture [9]

overcome the vanishing gradients problem of the simple Recurrent Neural Network (RNN) when facing the long term dependencies. For LSTM, the hidden layers have a more complicated structure. For instance, it has the concepts of gate and memory cell in each hidden layer. It has a three step process named as **Forget gate, Input gate, Output gate** shown in figure 2 . Firstly, the forget gate can help LSTM to decide whether the information should be thrown away from the cell state or not. the sigmoid function (σ) is used to computed the activation of the forget gate as

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.3}$$

the outputs of the forget gate (f_t) are a value between zero and one corresponding to the last cell state ($C_t - 1$). The value zero means omitting the last state thoroughly, while the value one is keeping the last state. Next, LSTM need to know what new information is going to be kept in the new cell state so the input/update gate (i_t) works on this process by using a sigmoid layer (σ);

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.4}$$

where W_i is the weight of input/update gate. At the same time, the previous hidden state (h_{t-1}) and current input (x_t) are computed into tanh function to make value between -1 and 1 for regulating the network where

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3.5}$$

Then, the output of input gate (i_t) and the candidate cell state ($\widetilde{C}_t$) are multiplied each other then the result will show what information is important to keep. Now, the information tend to be enough to calculate the cell state. Then, the previous cell state (C_{t-1}) go to multiply with the forget gate (f_t). This process have a possibility to drop the values in the cell state if the result value of multiplication near zero. After that, the output of the input gate (i_t) will sum up which updates the cell state (C_t) to new values so the neural network can find the relevant information, which gives

$$C_t = \widetilde{C}_t i_t + C_{t-1} f_t \tag{3.6}$$

Hence, we have the new cell state (C_t). At last, the output gate (o_t) will determine what the next hidden state should be. So, the output gate (o_t) is the previous hidden state (h_{t-1}) and the current input (x_t) are calculated into sigmoid function (σ) as

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.7}$$

From the previous gate (input gate), we have the recently new cell state (C_t) so we will pass it to the $\tanh$ function and then multiplied by the output (o_t) to get what the value of the hidden state should be, which give

$$h_t = o_t \cdot \tanh(C_t) \tag{3.8}$$

Hence, The new cell state (C_t) and the new hidden state (h_t) will be used for the next time step of Long Short Term Memory (LSTM) model.

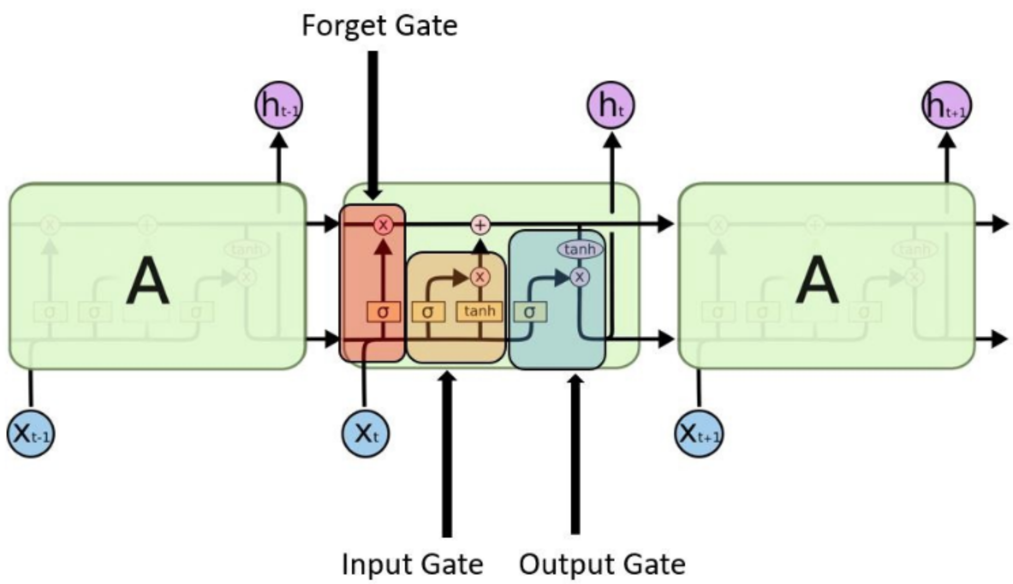

Figure 2: Long Short Term Memory architecture [9]

3.2 Prophet

Prophet is an open source software produced by Facebook's Core Data Science team. It is only available for R and Python. Prophet is specially an algorithm for use in propose of forecasting time series. The prominent points of Prophet is that Prophet is extremely optimized for the business prediction tasks since Prophet can encounter the character of the data as following; Firstly, the observations can be hourly, daily, weekly with at least a few months of history but preferably to a year. Secondly, it has very strong various seasonalities which can be set up by human-scale such as day of week, and time of year. Thirdly, the important holidays that can irregularly affect the intervals which are known beforehand for example, the Super Bowl. Next, a reasonable number of missing data, amount of outliers, and historical trend changes. Lastly, non-linear growth trends where a trend is at a natural limit. From the figure 3, it can be seen that there are analyst-in-the-loop and automated zones. Firstly, we begin by modeling the time series which each of the parameter can be easily interpretable for the analysts in the analyst-in-the-loop series. Then, we forecast the target values and set a baselines with the simulated forecast dates, and then evaluate forecast performance. If the performance of model is poor, we note these problems to the analysts so that the analysts can inspect the forecast and adjust the model to be more proper from this feedback.

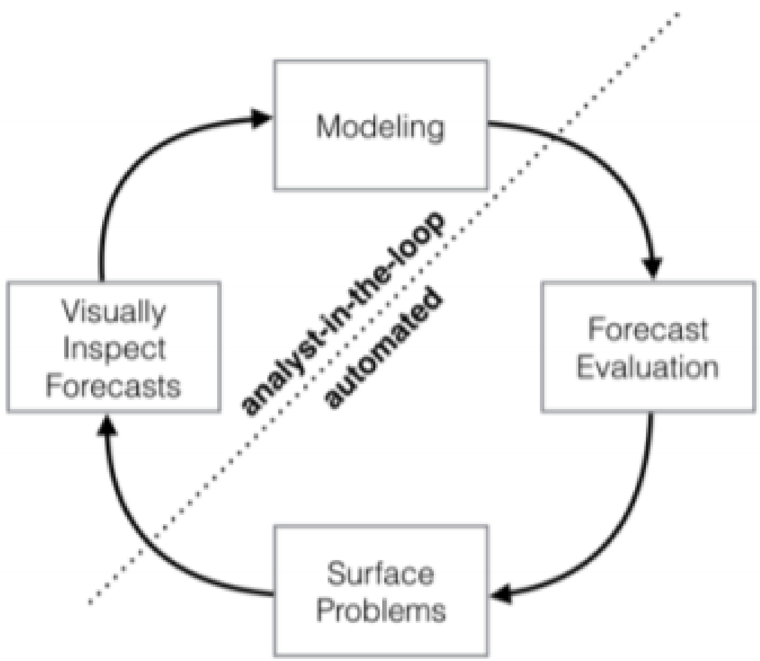

Figure 3: The picture illustrates the analyst-in-the-loop approach and automated tasks of Prophet [12]

Prophet model [12] can be decomposable in three main model components which are trend, seasonality, and holidays. These are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{3.9}$$

where $g(t)$ represents as the trend function which models non-periodic changes in the time series value, $s(t)$ represents as periodic changes for instance weekly and yearly seasonality, and $h(t)$

represents as the effects of holidays which occur on irregular schedules one or more than one day. The error term (ϵ_t) represents as any idiosyncratic changes which are not accommodated by the model; however, the parametric assumption that ϵ_t will be normal distributed. Now, we will look into how they calculate in each three main model component. In Trend Model $g(t)$, there are two trend models: **a saturating growth model**, and **a piecewise linear model**. For saturating growth (Nonlinear), nonlinear growth can saturates at a carrying capacity. It looks similar to population growth in natural ecosystems. This growth is using the logistic growth model and can be computed in its most basic form as:

$$g(t) = \frac{C}{1 + \exp\left(-k(t-m)\right),} \tag{3.10}$$

where C refers the carrying capacity, k refers the growth rate, and m refers an offset parameter. However, there are two more important point of growth that is not in the equation 3.10. Firstly, the carrying capacity C is not constant so it will be replaced the fixed capacity C with a time-varying capacity $C(t)$. Secondly, the growth rate is not constant. So, changepoints where the growth rate is allowed to change are applied to the growth model. For example, S Changepoints at times $s_j, j = 1, ..., S$. We define a vector of rate changes $\delta \in \mathbb{R}^S$, where δ_j is the change in rate which occurs at time s_j then the base rate k will be plus all of the adjustments rate; $k + \Sigma_{j:t>s_j}\delta_j$. To be more easily understand, defining a vector $a(t) \in 0, 1^S$ if t $\geq s_j$ $a_j(t)$ will be 1, otherwise 0. Then the rate at time t is $k + a(t)^\tau \delta$. Since the rate k is adjusted, the offset parameter m need to be adjusted also. Hence, it can be computed as

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l\right)\left(1 - \frac{k + \sum_{l<j}\delta_l}{k + \sum_{l\leq j}\delta_l}\right). \tag{3.11}$$

The logistic growth model can be calculated as

$$g(t) = \frac{C(t)}{1 + \exp\left(-(k + a(t)^\tau \delta)(t - (m + a(t)^\tau \gamma))\right)} \tag{3.12}$$

The parameter $C(t)$ is very important in the model. The analysts could often have the insight into market sizes. For a piecewise linear model, it is for forecasting problems which are not saturaiting growth. It can be computed as

$$g(t) = (k + a(t)^{\tau}\delta)t + (m + a(t)^{\tau}\gamma), \tag{3.13}$$

where k is the growth rate, δ is the rate adjustments, m is the offset parameter, and γ_j is set to $-s_j\delta_j$ to make the function continuous. For **seasonality** $s(t)$, multi-period seasonality is very essential in field of business forecasting. The seasonality function is based on Fourier series so the model of periodic effects is flexible. The seasonality can be computed as

$$s(t) = \sum_{n=1}^{N}\left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right)\right) \tag{3.14}$$

P is the regular period that we expect the time series to have. For instance, P = 365.25 for yearly or 7 for weekly data. Estimation of the 2N parameter $\beta = [a_1, b_1, ..., a_N, b_N]^{\tau}$ will be required for fitting the model. It can be computed as, for example with yearly seasonality and N = 10,

$$X(t) = \left[\cos\left(\frac{2\pi(1)t}{365.25}\right), ..., \sin\left(\frac{2\pi(10)t}{365.25}\right)\right] \tag{3.15}$$

So the seasonal component is

$$s(t) = X(t)\beta \tag{3.16}$$

From the equation 3.16, we are making the seasonality more smoothly by taking β $\sim Normal(0,\sigma^2)$. Lastly, **Holidays and Events** $h(t)$ can provide a large impact on many business time series. It can make a very high fluctuation at the specific dates or periods. The Holidays and Events $h(t)$ can be computed as

$$h(t) = Z(t)\kappa \tag{3.17}$$

where

$$Z(t) = [1(t \in D_1), ..., 1(t \in D_L)] \tag{3.18}$$

D_i are the set of past and future dates for each holiday i. Indicator function was added for representing whether time t is during holiday i. Then assigning each holiday i with parameter κ_1

which is the adjustment in the prediction. Prior $\kappa \sim \text{Normal}(0, v^2)$

3.3 AutoRegressive Integrated Moving Average (ARIMA)

Auto-regressive Integrated Moving Average or ARIMA [4],[5],[13] is a model for use in forecasting any non-seasonal time series based on own historical values which are its own lags and the lagged forecast errors. There are three parameters characterized p, d, and q which are required to determine for fitting the ARIMA model. For the first parameter, p is the order of the AutoRegressive (AR) term, Next, q is the order of the Moving Average (MA) term, and lastly d is the number of differencing for making time series stationary. Auto-regressive (AR) part can be computed as following equation;

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + \epsilon_1 \tag{3.19}$$

where Y_{t-1} represents the lag 1 of the time series, β_1 represents the coefficient of lag1 which the model estimates, and α represents the intercept term, also the model will estimate. Moving Average (MA) part can be computed as;

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + ... + \phi_q \epsilon_{t-q} \tag{3.20}$$

ϵ represents the errors of the auto-regressive part of the specific lags. Next, the propose of the order of differencing (d) is to make the time series stationary by substracting the previous value from the present value. If the time series already is stationary so d is zero. The Augmented Dickey Fuller test [14] are used to check if the time series is stationary. The null hypothesis of the Augmented Dickey Fuller test is the time series is non-stationary. Then, if the p-value of the test is less than 0.05 or significance level then the null hypothesis will be rejected. Hence, the time series is stationary. After we found the number of d, we will find the p and q for next step. The Partial Auto-correlation function (PACF) and the Auto-correlation function (ACF) are used to determine the number of Auto-regressive (AR) term and the Moving Average (MA) term respectively. Then we will select the number of lags which are above the significance line for both p and q parameters from the ACF and PACF plots. Once we have all the order number of three (p,d,q) parameters, the full equation of ARIMA model can be written as:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_p Y_{t-p}\epsilon_t + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + ... + \phi_q\epsilon_{t-q} \tag{3.21}$$

where Y_t is the predicted values.

3.4 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is usually used for measure the error of the absolute differences between predicted value and actual value in average. It can be computed as;

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{3.22}$$

3.5 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is also very common metric for measure of how far the predicted value from the actual value by taking square before taking root. So, it can be computed as;

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{3.23}$$

4 Methodology

In this section, there is where the data set we gathered and how we evaluated the performance of the experiment. Moreover, there are methods of each algorithm in the following subsections.

4.1 Dataset and Evaluation

The historical gold price data set consists of daily closed price for the period of 30^{th} August 2000 to 19^{th} June 2020 from finance.yahoo website. It includes High, Low, Open, Close, Volume and Adjusted close price in each day. For this paper we have used only Close price for an experiment. The gold price currency is the United State Dollar (USD). The amount of observations is 4957 which were split into training and testing set; 90% for training and 10% is for testing, 4461 observations and 496 observations respectively. It is shown in the figure 4. All the model were evaluated the accuracy by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The data set processing depends on the requirement of three different algorithms which shown in each following subsection below.

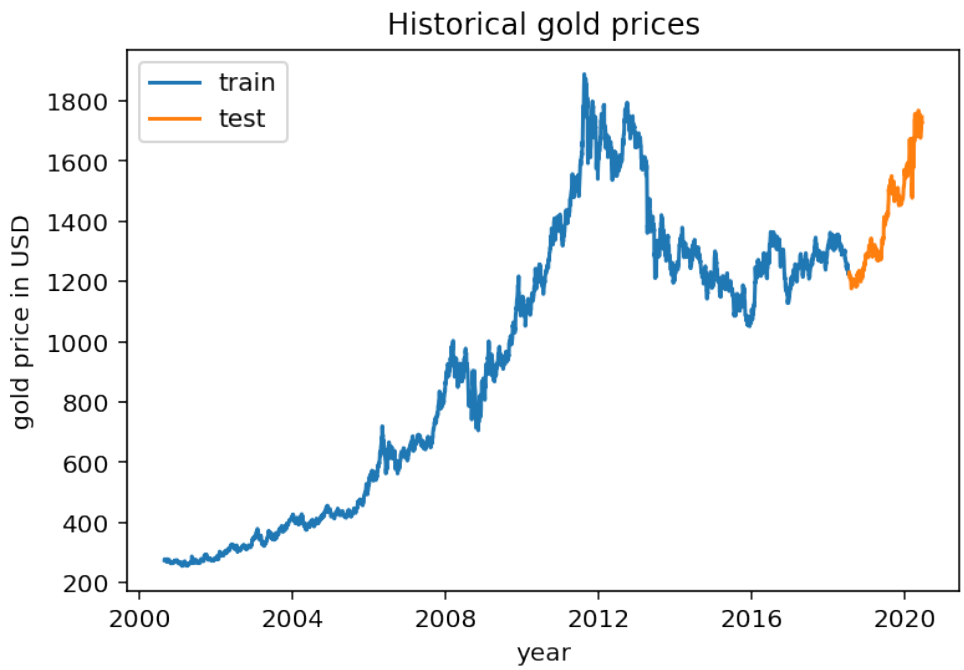

Figure 4: Gold price graph by training and testing set

4.2 Method of Long Short Term Memory (LSTM)

As we can see the gold price is very varies by time. Normalising data can help the algorithm for training so we normalised the data set by using maximum and minimum of training set because we assume that testing set is unseen data. Then, the data set is in range between 0 and 1. Next, we created the sliding window of last 3 days as an input for predicting the next day as an output. Then, the second input will be last 2 day of the previous input included the previous output for predicting the next output and so on for training the model. LSTM model were trained for 300 epochs with only single layer of 100 units, 0.5 of dropout rate layer, batch size of 64, and two dense layers with 20 units of Rectified Linear Unit (ReLu) methods and 1 unit with sigmoid activation function respectively.

4.3 Method of Prophet Model

For Facebook Prophet model, as it requires only two columns of data frame which are named ‘ds’ and ‘y’ refer to date time column and historical gold price column respectively. After that we can take training set as input into the prophet model. we set manually seasonality as following; for period of daily is 1 with 10 fourier order, weekly period is six days with 15 fourier order, montly period is 24 days with 10 fourier order, quarterly period is 72 days with 8 fourier order, yearly period is 265 days with 10 fourier order, and three year seasonality is 795 days with 20 fourier order since the gold market is open six days per week. And we set the growth parameter as linear since there is no saturation of gold price market, seasonality mode is multiplicative, seasonality prior scale is 25, change point scale is 30, and number of change point is 265 or a year.

4.4 Method of Auto-regressive Integrated Moving Average (ARIMA)

In ARIMA model, making time series stationary is required so Augmented Dickey-Fuller test was used to test the null hypothesis if the p-value is less than 0.05, the null hypothesis will be rejected hence time series is stationary. From the table 1, we can see that differencing the time series only one time is enough so d=1 is used in the model.

Table 1: Augmented Dickey-Fuller test

Augmented Dickey-Fuller test			
Order differencing	d=0	**d=1**	d=2
ADF Statistic	-1.31	**-20.85**	-19.22
P-value	0.624	**0.00**	0.00

After that we determine p and q parameters by plotting auto correlation Function and the partial auto correlation function with time series of d = 1. From the figure 5 and 6, it can be seen that both 1 lag are significant. In this work, we trained both ARIMA(p=0,d=1,q=0) and ARIMA(p=1,d=1,q=1) to see that which model can give least of Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values. The result is ARIMA(p=0,d=1,q=0) can give better performance which Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are 34644.037 and 34656.843 than ARIMA(p=1,d=1,q=1) of Akaike Information

Criterion (AIC) and Bayesian Information Criterion (BIC) are 34644.348 and 34669.960 respectively. Therefore, ARIMA(p=0,d=1,q=0) was deployed to forecast the future gold price.

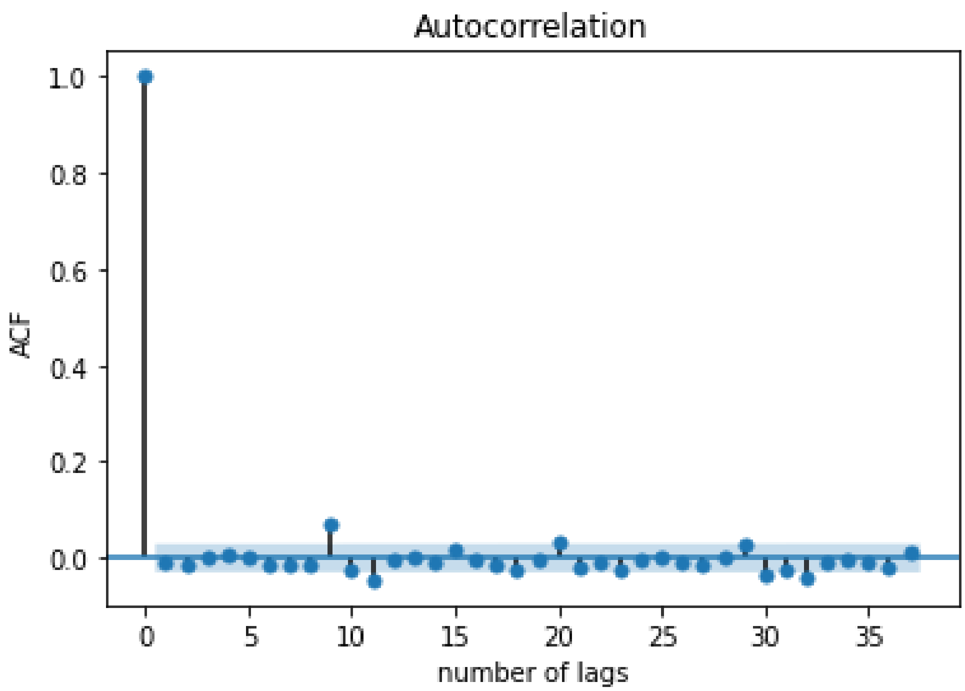

Figure 5: the Auto-correlation Function

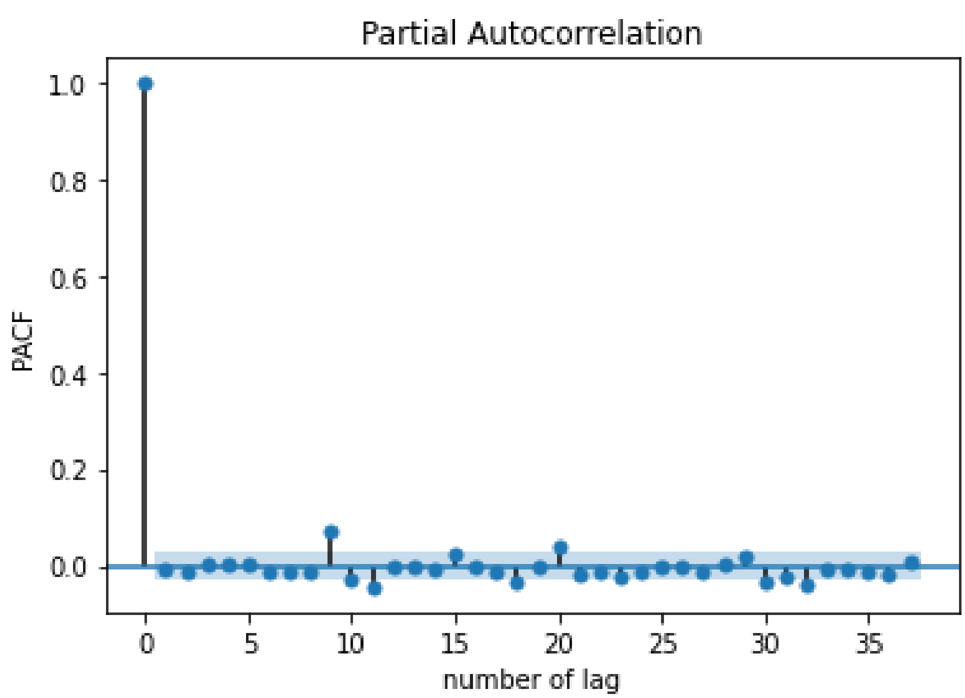

Figure 6: the Partial Auto-correlation Function

5 Results

The experiment of three different models was done. The evaluation of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) obtained for each model is given in the table 2. From the table, it can be clearly seen that Long Short Term Memory (LSTM) can provide more accurate outcome of gold price forecasting than Facebook prophet and Auto-regressive Integrated Moving Average (ARIMA) models. This is due to the reason that Long Short Term Memory (LSTM) is one type of non-linear model which can truly find the deep patterns of dynamical changes in gold price volatility in the individual current window whereas both Prophet and Auto-regressive Integrated Moving Average (ARIMA) models are type of linear models. They failed to capture the latent volatility occurring in the data. Since gold commodity has high dynamic and volatility, the range value of patterns and trends will always be different. Hence, prophet and Auto-regressive Integrated Moving Average (ARIMA) models are not possible to predict the future gold price accurately.

Table 2: Measurement of each model with RMSE and MAE

Model	RMSE	MAE
Long Short Term Memory (LSTM)	**22.39**	**14.93**
Facebook Prophet	187.61	156.41
Auto-regressive Integrated Moving Average (ARIMA)	205.05	156.14

From the figure 7, it can be clearly observed that Long Short Term Memory (LSTM) model is capable of predicting the movement of actual gold price between July 2018 and July 2020 but there is a bit gap between predicted gold price and actual gold price each day. This is due to the reason that Long Short Term Memory (LSTM) uses the previous current window for prediction of next day so that it can learn deeply the patterns and the volatility changes in the current window.

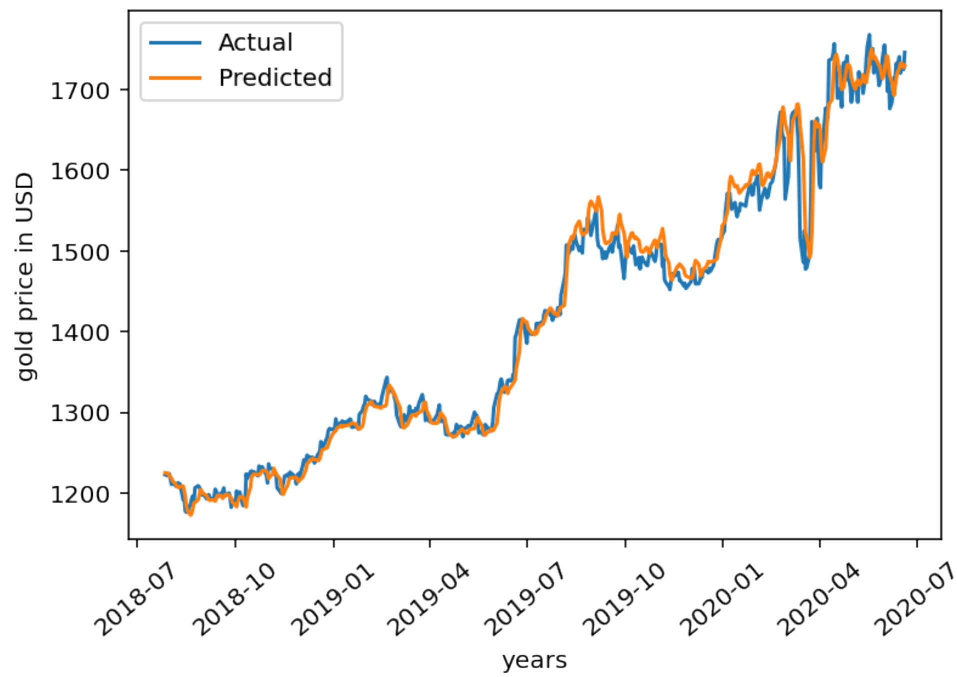

Figure 7: Comparison of Predicted gold price and Validation gold price of LSTM

The figure 8 illustrates the predicted gold price versus the actual gold price from July 2018 to July 2020. It can be observed that Facebook Prophet model is not capable of forecasting the future gold values correctly. The trend of predicted gold price is going down whereas the actual gold price is going up.

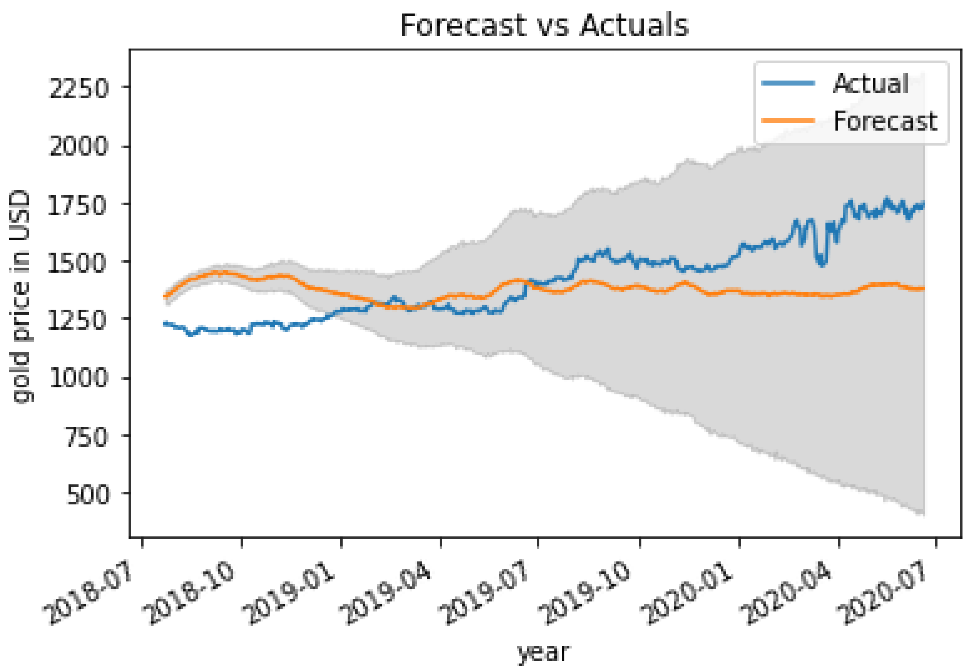

Figure 8: Comparison of Predicted gold price and Validation gold price of Prophet model

Last but not least, the figure 9 shows that the Auto-regressive Integrated Moving Average (ARIMA)(p=0,d=1,q=0) predicted a upward straightforward line of gold price aligned with actual gold price. Moreover, it truly can be observed that is not able to capture any volatility of actual daily gold price between July 2018 and July 2020. However, the 95% confidence interval can cover all the actual daily gold price.

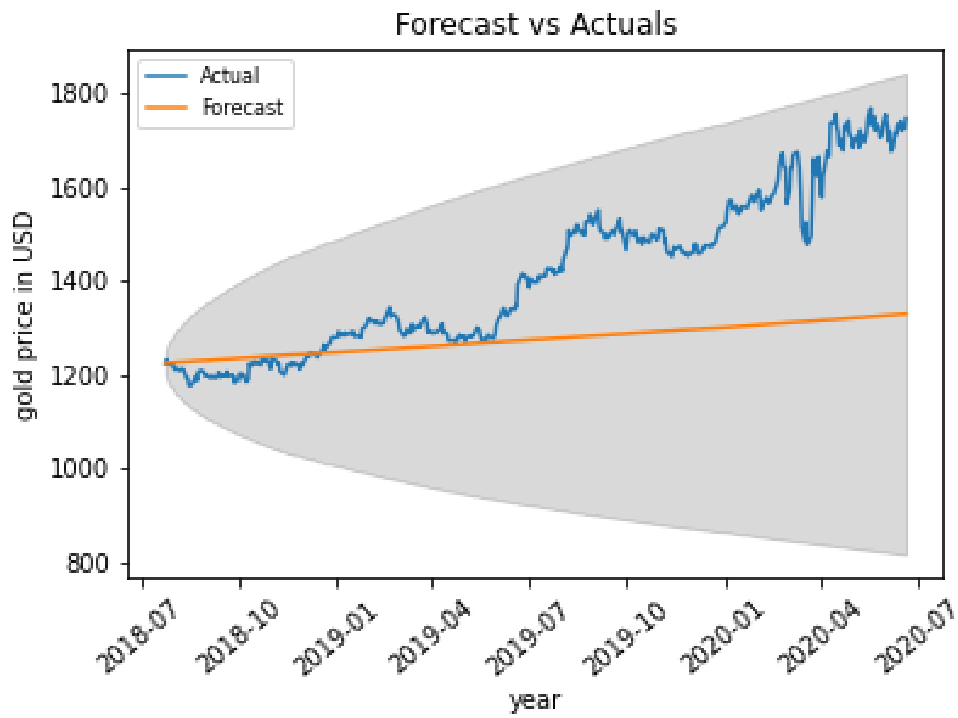

Figure 9: Predicted gold price vs actual gold price of ARIMA model

6 Discussion

The main findings of this research showed that Long short term memory (LSTM) model have higher capability of capturing the latent volatility of daily gold price movement and most accurate prediction than prophet and auto-regressive integrated moving average (ARIMA) models since long short term memory (LSTM) uses the recent information for prediction. However, auto-regressive integrated moving average (ARIMA) model showed that it can capture the upward trend of future gold price while prophet model cannot. Moreover, we deeply studied and understood how the architecture of three different algorithms does work. The data set of daily gold price from 30^{th} August 2000 to 19^{th} June 2020 were used in this paper. Two evaluation metrics, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), were used to analyse the performance of three different algorithms. Hence, it can clearly conclude that Long Short Term Memory (LSTM) is the best performance in this experiment. Additionally, there are other algorithms which can work well with time series data such as Convolutional Neural Network (CNN), and Gated Recurrent Unit (GRU). Therefore, in the future we will experiment on comparing the performance of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) algorithms to Long Short Term Memory (LSTM).

References

[1] ur Sami I and Junejo KN. Predicting future gold rates using machine learning approach. *International Journal of Advanced Computer Science and Applications*, 8(12):92–99, 2017. doi: https://doi.org/10.14569/IJACSA.2017.081213.

[2] Sieron A. What is gold: Asset, commodity, currency or collectible? [Online]. Available from: https://www.kitco.com/commentaries/2018-04-25/What-Is-Gold-Asset-Commodity-Currency-or-Collectible.html: :text=Gold [Accessed 15 August 2020].

[3] Mombeini H and Yazdani-Chamzini A. Modeling gold price via artificial neural network. *Journal of Economics, business and Management*, 3(7):699–703, 2015. doi: https://doi.org/10.7763/JOEBM.2015.

[4] Guha B and Bandyopadhyay G. Gold price forecasting using arima model. *Journal of Advanced Management Science*, 4(2), 2016. doi: https://doi.org/10.12720/joams.4.2.117-121.

[5] Yang X. The prediction of gold price using arima model. In *2nd International Conference on Social Science, Public Health and Education (SSPHE 2018)*. Atlantis Press, 2019. doi: https://doi.org/10.2991/ssphe-18.2019.66.

[6] Sivalingam KC., Mahendran S, and Natarajan S. Forecasting gold prices based on extreme learning machine. *International Journal of Computers Communications & Control*, 11(3): 372–380, 2016. doi: https://doi.org/10.1002/fut.3990130605.

[7] Ping PY, Miswan NH, and Ahmad MH. Forecasting malaysian gold using garch model. *Applied Mathematical Sciences*, 7(58):2879–2884, 2013. doi: https://doi.org/10.12988/ams.2013.13255.

[8] Selvin S, Vinayakumar R, Gopalakrishnan EA, Menon VK, and Soman KP. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017. doi: https://doi.org/10.1109/ICACCI.2017.8126078.

[9] Gudikandula P. Recurrent neural networks and lstm explained. [Online]. Available from: https://medium.com/@purnasaigudikandula/recurrent-neural-networks-and-lstm-ex plained-7f51c7f6bbb9/. [Accessed 6 August 2020].

[10] Hochreiter S and Schmidhuber J. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997. doi: https://doi.org/10.1162/neco.1997.9.8.1735.

[11] Bengio Y, Simard P, and Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. doi: https://doi.org/10.1109/72.279181.

[12] Taylor SJ and Letham B. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018. doi: https://doi.org/10.1080/00031305.2017.1380080.

[13] Prabhakaran S. Arima model - complete guide to time series forecasting in python. [Online]. Available from: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/. [Accessed 1 August 2020].

[14] Mushtaq R. Augmented dickey fuller test. 2011. doi: https://dx.doi.org/10.2139/ssrn.1911068.