

**REPORT**

SUBJECT: Gold prices forecasting:

A comparison of various forecasting models

**LECTURER: Assoc. Prof. Dr. Nguyễn Đình Thuân**

**TA: Nguyễn Minh Nhựt**

**CLASS: STAT3013.N11.CTTT**

**Phạm Thành Đạt – 20521175**

**Thiều Huy Hoàng - 20521350**

**Nguyễn Quang Vy -20522181**

**H**

**Phạm Tiến Sỹ - 18521356**

**Hình Thanh Thiên Trang - 18521523**

**VIETNAM NATIONAL UNIVERSITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

**INFORMATION SYSTEMS FACULTY**

INDEX

[I. Literature review 4](#_Toc123637688)

[II. Dataset 4](#_Toc123637689)

[III.Related Work 5](#_Toc123637690)

[IV. Method 7](#_Toc123637691)

* [**Performance measure 7**](#_Toc123637692)

[**A. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA) 8**](#_Toc123637693)

[**B. PROPHET 20**](#_Toc123637694)

[**C. LSTM CNN Model 32**](#_Toc123637695)

[**D. Bidirectional LSTM 45**](#_Toc123637696)

[**E. Linear Regression 58**](#_Toc123637697)

[**F. Support Vector Regression 67**](#_Toc123637698)

[V. CONCLUSION AND DISCUSSION 80](#_Toc123637699)

[VI. WORK ASSIGNMENT 82](#_Toc123637700)

[Reference: 83](#_Toc123637701)

ACKNOWLEDGMENT

Dear **Assoc. Prof. Dr. Nguyễn Đình Thuân** and **Mr. Nguyễn Minh Nhựt, teaching assistants** at the University of Information Technology!

First and foremost, I would like to express my deepest gratitude to **Assoc. Prof. Dr. Nguyễn Đình Thuân** and **Mr. Nguyễn Minh Nhựt** who always cared for our team during this course. I appreciate your assistance and the sharing of your valuable experience, which will help us accomplish our project. I would sincerely like to thank ***Mr. Nhựt***, who teaches and supports our team throughout our course.

I would like to take this opportunity to thank **Assoc. Prof. Dr. Nguyễn Đình Thuân** for permitting us to carry out our project. Finally, I would like to express how honored when our team was able to learn from and attend your class.

Many thanks to **Assoc. Prof. Dr. Nguyễn Đình Thuân** and **Mr. Nguyễn Minh Nhựt** for your tireless efforts in guiding the team to success and encouraging our team to keep moving forward. My heartfelt gratitude goes to all of my classmates, especially my friends, for devoting their time to assisting and supporting our team in the fabrication of our project.

Hồ Chí Minh City, 31 December 2022

Yours Sincerely,

Team 6

Gold prices forecasting:

A comparison of various forecasting models

# **Literature review**

In any unstable political and economic situation or crisis, gold is one of the most effective financial currencies for maintaining price and avoiding risk. From ancient times until now, gold has been an important currency for the Vietnamese people in general, so every year the Vietnamese people have a god of fortune, and on that day, everyone buys gold with the hope of luck. Most well-to-do families in Vietnam own some gold, and they are particularly interested in fluctuations in gold prices.

As a result, in this project, our group has chosen to predict the gold price using machine learning and deep learning models in the hope of assisting the people of Vietnam.

# **Dataset**

The data set was gathered on Kaggle Web between January 2, 2012, and December 30, 2022 **[1]**. It has 2870 rows and 2 attribute columns in total. One column describes the timeline of data representation, while the others show the gold price in VND over time.



Figure 1. Dataset

# **Related Work**

The list of paper which have been review based on problems under consideration, the problems, algorithms, evaluation.

**Table 1.** Table on Related Works

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SI No** | **Year** | **Title** | **Author(s)** | **Dataset** | **Problems** | **Purpose** | **Algorithm** | **Evaluation** |
| 1 | 2019 | Gold and Diamond Price Prediction Using Enhanced Ensemble Learning | (Pandey et al., 2019) [1] | Previous data of the product. | Variation in price of gold market. | To analyze and examine the patterns of previous close prices. | linear regression on and random forest | Mean, Best and worst is calculated for preciseness. |
| 2 | 2020 | Gold Price Prediction and Modelling using Deep Learning Techniques | (Vidya and Hari, 2020) [2] | Data taken from World Gold Council for year 1987 to44013 | Gold pricing nonlinearity. | To forecast gold price using LSTM | Long Shortterm Memory Network s (LSTM) | Root mean square error (RMSE) |
| 3 | 2017 | Forecasting Gold Price with Auto Regressive Integrated Moving Average Model | (Tripathy, 2017) [3] | Price of gold from July 1990 to February 2015 | Forecasting models forecasting is inaccurate | To gain more accuracy using ARIMA (Auto regressive Integrated Moving Average) | Box- Jenkins’ ARIMA (Auto regressive Integrated Moving Average) | Provides good results for the error measures used. |
| 4 | 2015 | Gold Price Prediction Using Type-2 Neuro- Fuzzy Modeling and ARIMA | (Modeling et al., 2015) | Gold Price historical data | When time factor is included in dataset there will uncertainty in results which might occur in future | To predict accuracy in predicting price of gold | type-2 neurofuzzy modeling and ARIMA (Auto regressive Integrated Moving Average) | Root mean square error (RMSE)Mean Absolute Percentage error (MAPE)Mean Absolute Error (MAE) |
| 5 | 2016 | Gold Price Forecasting Using ARIMA Model | Banhi Guha and Gautam Bandyopadhyay [4] | secondary monthly data for Gold price, collected from Multi Commodity Exchange of India Ltd (MCX) ranging from November 2003 to January 2014. | Forecasting models forecasting is inaccurate | To forecast the price of Gold using time-series ARIMA Model. | ARIMA (Auto regressive Integrated Moving Average) | Provides good results for the error measures used. |
| 6 | 2018 | The Prediction of Gold Price Using ARIMA Model | Xiaohui Yang  [5] | The data are collected from the World Gold Council, consisting of 1305 observations of daily gold price from July 1st 2013 to June 29 2018. | Variation in price of gold market. | investigate and carry out the prediction of the future international gold price | ARIMA (Auto regressive Integrated Moving Average) | Provides  the most accurate and appropriate model for forecasting |
| 7 | 2019 | Gold Price Forecast based on LSTM-CNN Model | Zhanhong He , Junhao Zhou , Hong-Ning Dai , Hao Wang[6] | World Gold Council [2](WGC) contains 10471 daily gold price transaction record from Dec. 29, 1978 to Feb. 15, 2019 (only on trading day) | Forecasting models forecasting is inaccurate | predict the tendency of daily gold price. | LSTM and CNN neural networks with Attention Mechanism | Root mean square error (RMSE)  Mean Absolute Percentage error (MAPE) Root mean absolute error (RMAE) |

# **Method**

Although there are numerous time-series forecasting models available, this project presents an empirical evaluation of ***seven popular time-series forecasting models*** for ***forecasting gold prices***. In particular, six forecasting models are:

1. **Autoregressive integrated moving average (ARIMA)**
2. **Prophet**
3. **LSTM CNN** **model**
4. **Bidirectional LSTM**
5. **Linear regression.**
6. **Support Vector Regression**

Our project will be implementing multiple forecasting models and comparing their performance using error measures namely MAPE and RMSE for each model to determine which one is the most optimal for estimating the price.

### **Performance measure**

To assess the predictive power of our proposed models, we use two performance measures: the root means square error (RMSE) and the MAPE. When we train models, we use RMSE as a loss function, and MAPE is a statistical measure of prediction accuracy. The following are the equations:



Where M is the number of data points,  is a predicted value and  is a real value.

## **AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)**

* **Definition**

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses [time series data](https://www.investopedia.com/terms/t/timeseries.asp) to either better understand the data set or to predict future trends.

A statistical model is autoregressive if it predicts future values based on past values. For example, an ARIMA model might seek to predict a gold future prices based on its past performance or forecast a company's earnings based on past periods. **[4]**

The ARIMA model stands for Auto Regression (AR), Moving Average (MA) and Differential Integration Integrated - I.

**Important point**: The ARIMA model is not a perfect predictive model for any time series data.

The ARIMA model only works best if the data is highly time dependent. Randomized data usually do not work for ARIMA models.

The ARIMA model is only good at predicting time points.

* **Types of Models ARIMA:**

The ARIMA model is not seasonal

Seasonal ARIMA model (Seasonal ARIMA – SARIMA)

* **Stationary**

A stationary time series is a series of mean, variance, and autocorrelation values ​​that do not change over time and it does not include the trend factor. With most statistical predictive methods, the calculation must be ensured. stationarity of the data series, so checking for stationarity is very important. To test the stationarity of data, we have two popular testing methods: Dickey (DF) test and Improved Dickey Fuller (ADF4). **[5]**

* **ARIMA (*p, d, q*)**

The parameter ***p***is the number of autoregressive terms or the number of “lag observations.” It is also called the “lag order,” and it determines the outcome of the model by providing lagged data points.

The parameter ***d***is known as the degree of differencing. it indicates the number of times the lagged indicators have been subtracted to make the data stationary.

The parameter ***q*** is the number of forecast errors in the model and is also referred to as the size of the moving average window.

The parameters take the value of integers and must be defined for the model to work. They can also take a value of 0, implying that they will not be used in the model. In such a way, the ARIMA model can be turned into:

ARMA model (no stationary data, ***d*= 0**)

AR model (no moving averages or stationary data, just an autoregression on past values, ***d*= 0, *q*= 0**)

MA model (a moving average model with no autoregression or stationary data, ***p*= 0, *d* = 0)**

Therefore, ARIMA models may be defined as:

1. ARIMA(1, 0, 0) – known as the **first-order autoregressive model**
2. ARIMA(0, 1, 0) – known as the **random walk model**
3. ARIMA(1, 1, 0) – known as the **differenced first-order autoregressive model**, and so on.

Once the parameters (*p, d, q*) have been defined, the ARIMA model aims to estimate the coefficients **α** and **θ**, which is the result of using previous data points to forecast values. **[6]**

* **Implementation in Python language**
  1. **Import libraries**

Graphical user interface, text, application

Description automatically generated

Figure 2. Import libraries

* 1. **Import data and get Date column as Index**

Graphical user interface, text, application, email

Description automatically generated

Figure 3. Import data

* 1. **Data visualization**

A picture containing background pattern

Description automatically generated

Chart, scatter chart

Description automatically generated

Figure 4. Visualize data

* 1. **Check if data is stationary series using adf test**

Text

Description automatically generated

Figure 5. Code ADF test

Text

Description automatically generated

Figure 6. Test the stationary of the original data

Since the original series is not stationary, we take the first difference (d = 1) of the series to test for stationarity

Text

Description automatically generated

Figure 7. Test stationary of the first difference (d=1)

Table

Description automatically generated

Figure 8. Data after add d1

* 1. **Split data**

We will split the dataset into train and test sets, we take 90% train and 10% test

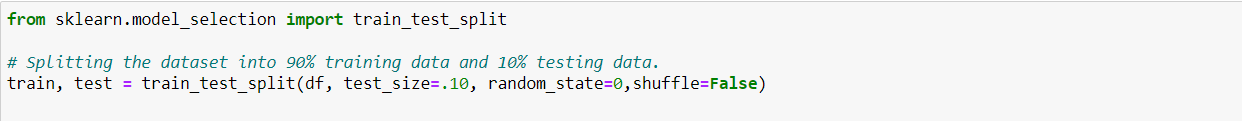


Figure 9. Split data

* 1. **Find coefficients p,q,d**

Build ARIMA model, find coefficients p,q,d using auto\_arima . function

Graphical user interface, table

Description automatically generated with medium confidence

Figure 10. Result of auto\_arima function

After using auto arima, we find 3 coefficients p, d, q respectively 1,1,1

Table

Description automatically generated

Figure 11. Use ARIMA to train model

* 1. **Find predictive data and plot model**

Find predictive data based on train data and compare with test set



Figure 12. Create predict data

Text

Description automatically generated

Figure 13. Compare predict data to test set

Plot predictions against known values

A picture containing text

Description automatically generated

Chart, line chart

Description automatically generated

Figure 14. Model of train, test and predict data

* 1. **Calculate RMSE**

Graphical user interface, text, application, email

Description automatically generated

Figure 15. Calculate RMSE

* 1. **Calculate MAPE**

Graphical user interface, text

Description automatically generated

Figure 16. Calculate MAPE

* 1. **Prediction for the next 30 days**

Graphical user interface, text, application, email

Description automatically generated

Figure 17. Predict the next 30 days

* 1. **Plot predictions against known values**

Text

Description automatically generated with medium confidence

Graphical user interface, chart, line chart, scatter chart

Description automatically generated

Figure 18. Model after predict

Predicted data for the next 30 days

Table

Description automatically generated

Figure 19. Result of the predict 30 days

We perform ARIMA model on 3 cases:

**Table 2. Measuring the ARIMA model according to split data**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train-Test | RMSE | MAPE |
| ARIMA | 7-3 | 5578826.49 | 11.85% |
| 8-2 | 4697358.81 | 10.32% |
| 9-1 | 2074521.9 | 3.94% |

After measuring the ARIMA model, the model with 90% data for training and 10% data for testing is the most optimal in all three cases. We see that the results are very good, with very low RMSE (2074521.9) and MAPE (3.94%), because there has been no strong fluctuation or sudden change over the years due to the characteristics of the gold data set.

## **PROPHET**

Prophet is a free, open-source application developed by Facebook for forecasting time series data, which aids in understanding and potential market forecasts for organizations. It is based on a decomposable additive model, which also accounts for the effects of vacations, and fits non-linear trends with seasonality. **[7]**  
**Trend:**

The trend shows the tendency of the data to increase or decrease over a long period of time and it filters out the seasonal variations.

**Seasonality:**

Seasonality is the variations that occur over a short period of time and is not prominent enough to be called a “trend”.

**Understanding the Prophet Model**

The general idea of the model is similar to a generalized additive model. The “Prophet Equation” fits, as mentioned above, trend, seasonality and holidays. This is given by,

**y(t) = g(t) + s(t) + h(t) + e(t)**

where:

g(t) refers to trend (changes over a long period of time)

s(t) refers to seasonality (periodic or short term changes)

h(t) refers to effects of holidays to the forecast

e(t) refers to the unconditional changes that is specific to a business or a person or a circumstance. It is also called the error term.

y(t) is the forecast.

* **Implementation in Python language**

1. **Import libraries:**

Graphical user interface, text, application

Description automatically generated

Figure 20. Import libraries

1. **Import data**

Graphical user interface, text, application

Description automatically generated

Figure 21. Import data

1. **Split data**

We will split the dataset into train and test sets, we take 70% train and 30% test

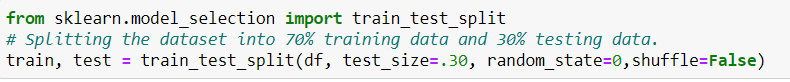


Figure 22. Spit data

1. **Reading data**

Prophet's input is always a dataset with two attributes ‘ds’ and ‘y’. Where ‘ds’ has date format, timestamp. And the ‘y’ column represents the quantitative value, which represents the measurement we predict.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Figure 23. Read data train and test

Graphical user interface, text, application

Description automatically generated

Figure 24. Combine data train and test

1. **Data visualization**

Chart, scatter chart

Description automatically generated

Figure 25. Visualize data

1. **Check the train test set again**

Graphical user interface, text, application

Description automatically generated

Figure 26. Check size

1. **Create data to predict and compare with test set**

Create data future with data test to predict and compare the results with the test set, using prophet and print out the model

Graphical user interface, text

Description automatically generated

Figure 27. Create predict data

Chart, line chart, histogram

Description automatically generated

Figure 28. Plot predict data

Text

Description automatically generated

Chart, scatter chart

Description automatically generated

Figure 29. Plot predict data

1. **Predict 1 year**

Predictions are then made on a dataframe with a column ds containing the dates for which a prediction is to be made. You can get a suitable dataframe that extends into the future a specified number of days using the helper method “Prophet.make\_future\_dataframe”. By default it will also include the dates from the history, so we will see the model fit as well.

Graphical user interface, text, application

Description automatically generated

Figure 30. Create future data

Then we use predict function to predict 1 year later

Text, table

Description automatically generated

Figure 31. Result of future data

1. **Visualize the future model**

After having predictive future data, we visualize the model

Graphical user interface

Description automatically generated with low confidence

Chart, histogram

Description automatically generated

Figure 32. Plot future data

Text

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated

Figure 33. Plot future data

We use the Prophet.plot\_components method to see the forecast components. By default you’ll see the trend, yearly seasonality, and weekly seasonality of the time series.

Shape

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated

Graphical user interface, application, table

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated

Figure 34. Components of data

1. **Calculate MAPE, RMSE**

Text, letter

Description automatically generated

Figure 35. Result of MAPE, RMSE

We perform PROPHET model on 3 cases:

**Table 3.** Measuring the PROPHET model according to split data

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train-Test | RMSE | MAPE |
| PROPHET | 7-3 | 1573796.94 | 2.99% |
| 8-2 | 1664335.79 | 3.28% |
| 9-1 | 1753647.79 | 3.39% |

After measuring the PROPHET model, the model with 70% data for training and 30% data for testing is the most optimal in all three cases. We see that the results are very good, with very low RMSE (1573796.94) and MAPE (2.99%), because there has been no strong fluctuation or sudden change over the years due to the characteristics of the gold data set.

## **LSTM CNN Model**

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network that can learn order dependence in sequence prediction problems. This is a necessary characteristic in complex problem domains such as machine translation, speech recognition, and others. **[7]**

An LSTM layer is made up of a collection of recurrently connected memory blocks. Each block contains one or more recurrently connected memory cells through three multiplicative units - the input, output, and forget gates. These provide continuous analogs of the cells' write, read, and reset operations.

The advent of LSTM networks minimizes the drawback of gradient vanishing in part by allowing information to propagate more directly through the cell state.

LSTM cell:

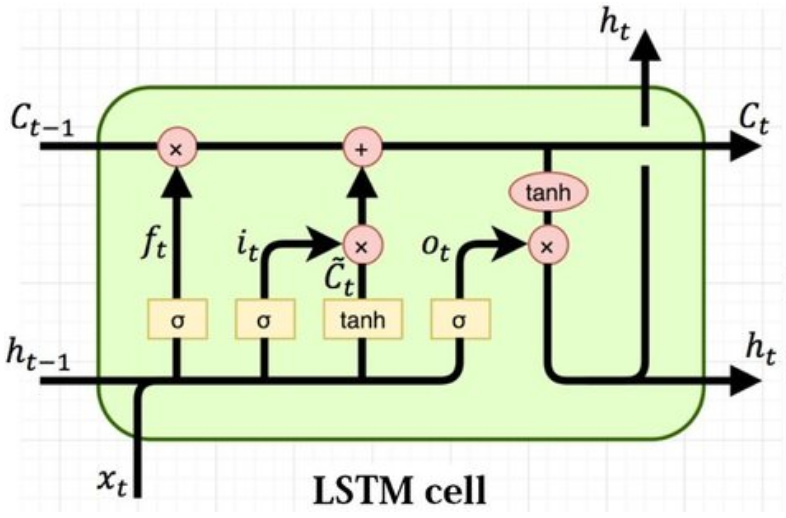


Figure 36. LSTM cell **[8]**

Calculate in LSTM cell:

 **[9]**

* **Implementation in Python language**

1. **Import libraries**

Text

Description automatically generated

Figure 37. Import libraries

1. **Import data**

A screenshot of a computer

Description automatically generated with medium confidence

Figure 38. Import data

1. **Visualization Gold Price in the past**

Text

Description automatically generated

Chart, histogram

Description automatically generated

Figure 39. Visualize data

1. **Train Split**

Text

Description automatically generated

Figure 40. Split data

1. **Scale data**

Text

Description automatically generated

Figure 41. Scale data

1. **Create Training Data**

Text

Description automatically generated

Figure 42. Create Training Data

1. **Build Model**

Text

Description automatically generated

Figure 43. Build model

1. **Save Model**

Text

Description automatically generated

Figure 44. Save model

1. **Create Testing Data**

Text

Description automatically generated

Figure 45. Create Test data

1. **Evaluate On Test Data**

Graphical user interface, website

Description automatically generated

Figure 46. Evaluate

1. **Plot and visualize data**

Text

Description automatically generated

Chart

Description automatically generated

Figure 47. Visualize data

1. **Compare the actual and prediction values**

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 48. Compare

1. **Measure model by using MAPE and RMSE**

Text

Description automatically generated

Figure 49. Calculate MAPE, RMSE

1. **Predict the next 30 days**
   1. ***Take records and create list***

Graphical user interface, text, application

Description automatically generated

Figure 50. Create List

* 1. ***Predict next 30 days price using the current data***

Text

Description automatically generated

Figure 51. Predict

* 1. ***Inverse transformations and print out***

Graphical user interface, text

Description automatically generated

Figure 52. Inverse and print out

* 1. ***Visualization the next 30 days price of gold***

Text

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

Figure 53. Visualize predict data

Text

Description automatically generated

Chart

Description automatically generated

Figure 54. Visualize predict data 30 days

* **Conclusion:** After train and test split for three cases such as 90% train with 10% test, 80% train with 20% test, and 70% train with 30% test. I had the table of results below:

**Table 4.** Measuring the LSTM model according to split data

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Train-Test** | **RMSE** | **MAPE** |
| **LSTM** | 7-3 | 465100 | 0.81% |
| 8-2 | 512538 | 1.01% |
| 9-1 | 404883 | 0.74% |

After measuring the LSTM model, the model with 90% data for training and 10% data for testing is the most optimal in all three cases. We see that the results are very good, with very low RMSE (404883) and MAPE (0.74%), because there has been no strong fluctuation or sudden change over the years due to the characteristics of the gold data set.

## **Bidirectional LSTM**

Bidirectional long-short term memory (BiLSTM) is a technique that allows any neural network to store sequence information both forward and backward. BiLSTM allows input flow in both directions, whereas normal LSTM only allows input flow in one direction. **[10]**

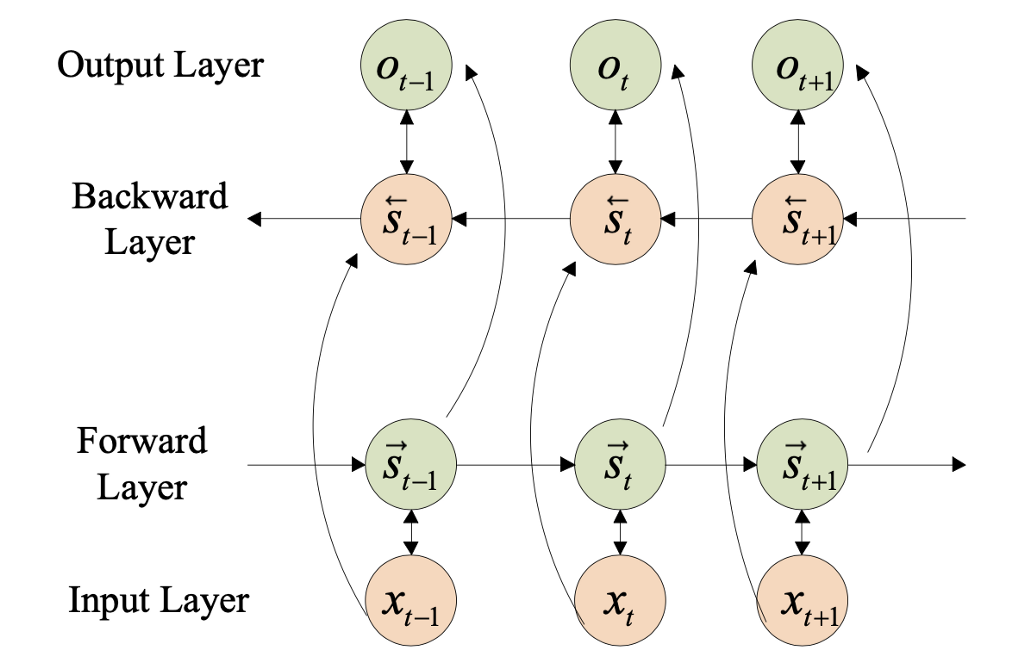


Figure 55. The basic structure of bidirectional LSTM **[11]**

* **Implementation in Python language**
  1. **Import libraries**

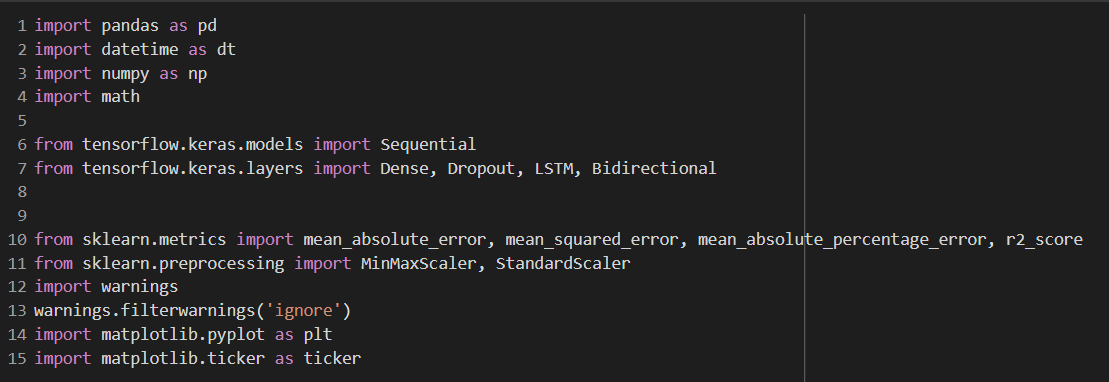


Figure 56. Import libraries

* 1. **Import data**

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 57. Import data

* 1. **Visualization Gold Price in the past**



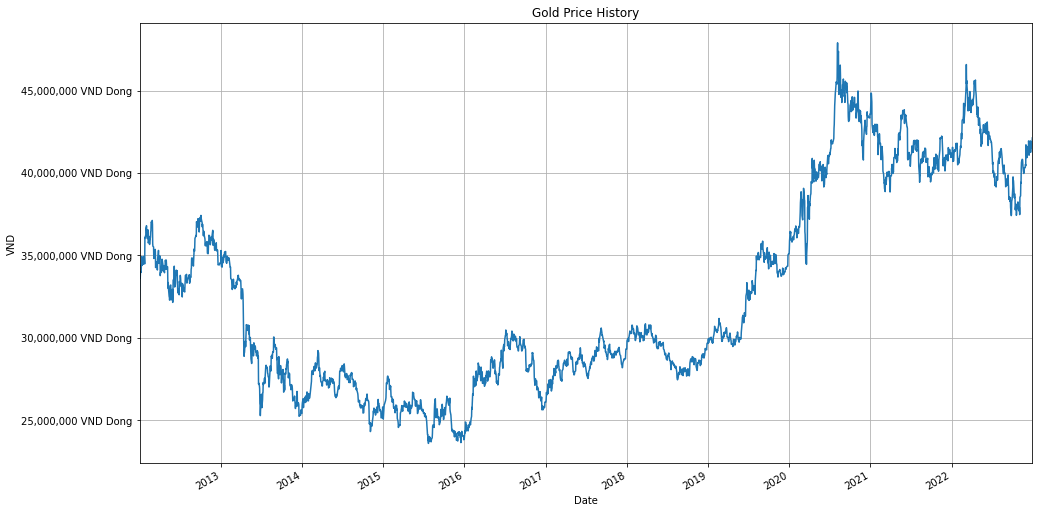
****

Figure 58. Visualize data

* 1. **Train Split**

Text

Description automatically generated

Figure 59. Split data

* 1. **Scale data**

Text

Description automatically generated

Figure 60. Scale data

* 1. **Create training data**

Text

Description automatically generated

Figure 61. Create Train data

* 1. **Build Model**

Text

Description automatically generated

Figure 62. Build model

* 1. **Save Model**



Figure 63. Save model

* 1. **Create testing data**

Text

Description automatically generated

Figure 64. Create Test data

* 1. **Evaluate on test data**

Text

Description automatically generated

Figure 65. Evaluate

* 1. **Plot and visualize data**

Text

Description automatically generated

**Chart

Description automatically generated**

Figure 66. Visualze data

* 1. **Compare the actual and prediction values**

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 67. Compare

* 1. **Measure model by using MAPE and RMSE**

Text

Description automatically generated

Figure 68. Calculate MAPE,RMSE

* 1. **Predict the next 30 days**

**14.1 Take records and create list**

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 69. Create List

**14.2 Predict next 30 days price using the current data**

Text

Description automatically generated

Figure 70. Predict 30 days

**14.3 Inverse tranformations and print out**

A picture containing graphical user interface

Description automatically generated

Figure 71. Result predict data

**14.4 visualization the next 30 days price of gold**

Text

Description automatically generated

**Graphical user interface, chart, line chart

Description automatically generated**

Figure 72. Visualize the next 30 days

Text

Description automatically generated

Figure 73. Visualize the next 30 days

**Chart

Description automatically generated**

Figure 74. Visualize the next 30 days

* **Conclusion:** After train and test split for three cases such as 90% train with 10% test, 80% train with 20% test, and 70% train with 30% test. I had the table of results below:

**Table 5.** Measuring the Bi-LSTM model according to split data

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train-Test | RMSE | MAPE |
| Bi-LSTM | 7-3 | 493253 | 0.96% |
| 8-2 | 511867 | 0.97% |
| 9-1 | 397070 | 0.73% |

After measuring the Bi-LSTM model, the model with 90% data for training and 10% data for testing is the most optimal in all three cases. We see that the results are very good, with very low RMSE (397070) and MAPE (0.73%), because there has been no strong fluctuation or sudden change over the years due to the characteristics of the gold data set.

## **Linear Regression**

* ***Definition***

In the most simple words, Linear Regression is the supervised Machine Learning model in which the model finds the best fit linear line between the independent and dependent variable i.e it finds the linear relationship between the dependent and independent variable. [12]

Linear Regression is of two types: Simple and Multiple. Simple Linear Regression is where only one independent variable is present and the model has to find the linear relationship of it with the dependent variable. Whereas, In Multiple Linear Regression there are more than one independent variables for the model to find the relationship.

Equation of Simple Linear Regression, where b0 is the intercept, b1 is coefficient or slope, x is the independent variable and y is the dependent variable.

Equation of Multiple Linear Regression, where b0 is the intercept, b1, b2, b3, b4…,bn are coefficients or slopes of the independent variables x1, x2, x3, x4…, xn and y is the dependent variable.

A Linear Regression model’s main aim is to find the best fit linear line and the optimal values of intercept and coefficients such that the error is minimized.

Error is the difference between the actual value and Predicted value and the goal is to reduce this difference.

Mathematical Approach:

Residual/Error = Actual values – Predicted Values

Sum of Residuals/Errors = Sum(Actual- Predicted Values)

Square of Sum of Residuals/Errors = (Sum(Actual- Predicted Values))­2

* **Implementation in Python language**

1. **First step we need to add the necessary libraries**

**Graphical user interface, text, application

Description automatically generated**

Figure 75. Import libraries

1. **Next, we need to get the data for analysis**

****

Figure 76. Import data

1. **To be able to predict by time series we need to create an additional column "Timestamp"**

**Table

Description automatically generated with medium confidence**

Figure 77. Create 'Timestamp'

1. **Draw graphs to visualize input data**

**Chart, line chart

Description automatically generated**

Figure 78. Visualize data

1. **Then we start normalizing / scaling the input data with the StandardScaler() method.**

**Text

Description automatically generated**

Figure 79. Scale data

1. **To have data for train and test, we need to divide the normalized data set into 2 fractions with a ratio of 9/1 corresponding to train and test.**

**Text

Description automatically generated**

Figure 80. Split data

**Chart

Description automatically generated**

Figure 81. Visualize data

1. **Train and run the prediction results**

**Text

Description automatically generated with medium confidence**

Figure 82. Build model

1. **After training and predicting the value, we return the data to its original form.**

**Text

Description automatically generated**

Figure 83. Inverse data

1. **Plot graphs to see predicted results against test data.**

Graphical user interface, chart, line chart

Description automatically generated

Figure 84. Visualize predictive data

1. **Finally, we evaluate the model through MAE, MAPE, MSE, RMSE and R2 indexes.**

**Text

Description automatically generated**

Figure 85. Calculate MAPE,RMSE

1. **Predicting next 30 days price**

Graphical user interface, text, application, email

Description automatically generated

Figure 86. Predict next 30 days

## **Support Vector Regression**

* ***Definition***

Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. The main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated. In SVR, the best fit line is the hyperplane that has the maximum number of points. **[13]**

Chart, scatter chart

Description automatically generated

Figure 87.

Solution:

Constraints:

Chart, scatter chart

Description automatically generated

Figure 88

Minimize:

Constraints:

Linear SVR:

Non-linear SVR:

The kernel functions transform the data into a higher dimensional feature space to make it possible to perform the linear separation.

Types of Kernel functions: **[14]**

**Gaussian RBF Kernel:** The RBF kernel is also called the Gaussian kernel. There is an infinite number of dimensions in the feature space because it can be expanded by the Taylor Series. In the format below, The γ parameter defines how much influence a single training example has. The larger it is, the closer other examples must be to be affected. It is a general-purpose kernel; used when there is no prior knowledge about the data.

**Polynomial kernel**: “Intuitively, the polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these”. With n original features and d degrees of polynomial, the polynomial kernel yields n^d expanded features where d is the degree of polynomial. It is popular in image processing.

**Sigmoid Kernel**: The Hyperbolic Tangent Kernel is also known as the Sigmoid Kernel and as the Multilayer Perceptron (MLP) kernel. The Sigmoid Kernel comes from the Neural Networks field, where the bipolar sigmoid function is often used as an activation function for artificial neurons. We can use it as the proxy for neural networks.

* **Implementation in Python language**

1. **First step we need to add the necessary libraries**

**Graphical user interface, text

Description automatically generated**

Figure 89. Import libraries

1. **Next, we need to get the data for analysis**

****

Figure 90. Import data

1. **To be able to predict by time series we need to create an additional column "Timestamp"**

**Table

Description automatically generated with medium confidence**

Figure 91. Add 'Timestamp'

1. **Draw graphs to visualize input data**

**Chart, line chart

Description automatically generated**

Figure 92. Visualize data

1. **Then we start normalizing / scaling the input data with the StandardScaler() method.**

**Text

Description automatically generated**

Figure 93. Scale data

1. **To have data for train and test, we need to divide the normalized data set into 2 fractions with a ratio of 9/1 corresponding to train and test.**

**Text

Description automatically generated**

Figure 94. Split data

**Chart, line chart

Description automatically generated**

Figure 95. Visualize data

1. **Next, we create auxiliary functions to help find suitable parameters for the model**

**Graphical user interface, text, application

Description automatically generated**

Figure 96. Find parameters

**Text

Description automatically generated**

Figure 97. Code find parameters

**Text

Description automatically generated**

Figure 98. Code find parameters

1. **Initialize values ​​for the parameter finding process**

**Text

Description automatically generated**

Figure 99. Generate values

1. **From the above results, we use to write a function to get the most optimal parameters for the model**

**Text

Description automatically generated with low confidence**

Figure 100. Find best fit model

1. **Train and run the prediction results**

**Text

Description automatically generated**

Figure 101. Build model

1. **After training and predicting the value, we return the data to its original form.**

**Text

Description automatically generated**

Figure 102. Inverse model

1. **Plot graphs to see predicted results against test data.**

**Chart, line chart

Description automatically generated**

Figure 103. Visualize predictive data

1. **Finally, we evaluate the model through MAE, MAPE, MSE, RMSE and R2 indexes.**

Text

Description automatically generated

Figure 104. Calculate MAPE,RMSE

1. **Predicting next 30 days price**

**Graphical user interface, text, application, email

Description automatically generated**

Figure 105. Predict the next 30 days

# **CONCLUSION AND DISCUSSION**

The following table are the experimental results of six models for training and testing we got it:

**Table 6.** Measuring the six models according to the past values

|  |  |  |
| --- | --- | --- |
| **Model** | **RMSE** | **MAPE** |
| **LR** | 4456279.83 | 9.33% |
| **SVR** | 1694564.28 | 3.42% |
| **ARIMA** | 2074521.9 | 3.94% |
| **PROPHET** | 1573796.94 | 2.99% |
| **LSTM** | 404883 | 0.74% |
| **BI-LSTM** | 397070 | 0.73% |

After analysis of six models of time series forecasting and its corresponding RMSE and MAPE. This can be easily verified that the Recurrent Neural Network (RNN) based Bidirectional - Long Short-Term Memory (Bi - LSTM) Model-design gives the best forecast on test data with RMSE error of 397070 and MAPE Error of 0.73%.

**Table 7.** Compare actual price and predicted price check the accuracy of 6 models

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Actual | Bi-LSTM | LSTM | ARIMA | PROPHET | SVR | Linear |
| **12/31/2022** | **41927010** | 42069188.21 | 42054234.06 | 42129308 | 37603478.8 | 35288112 | 26603497 |
| **1/1/2023** | **41927010** | 42004603.29 | 41944918.55 | 42121773 | 37967170.6 | 34008069 | 26963968 |
| **1/2/2023** | **41908626** | 41921593.08 | 41826965.05 | 42115622 | 39736599.1 | 32536728 | 27324439 |
| **1/3/2023** | **42030420** | 41845726.59 | 41723981.66 | 42110681 | 39762463.1 | 30984957 | 27684910 |
| **1/4/2023** | **42237240** | 41786211.42 | 41642517.87 | 42106798 | 39767633.4 | 29479879 | 28045381 |

**Table 8.** Calculate percentage error of 6 models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Percentage Error (%) | | | | | |
| Bi-LSTM | LSTM | ARIMA | PROPHET | SVR | Linear |
| **12/31/2022** | -0.339 | -0.303 | -0.483 | 10.312 | 15.834 | 36.548 |
| **1/1/2023** | -0.185 | -0.043 | -0.465 | 9.445 | 18.887 | 35.688 |
| **1/2/2023** | -0.031 | 0.195 | -0.494 | 5.183 | 22.363 | 34.8 |
| **1/3/2023** | 0.439 | 0.729 | -0.191 | 5.396 | 26.28 | 34.131 |
| **1/4/2023** | 1.068 | 1.408 | 0.309 | 5.847 | 30.204 | 33.6 |

After collecting the actual price and getting the forecast price, we calculate the percentage error, we see that the lowest percentage error belongs to the Bi-LSTM model and the highest percentage error belongs to the Linear Regression model. So the most optimal model is the Bi-LSTM model and the least optimal model is the Linear Regression model.

# **WORK ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Members**  **Work** | **Phạm Thành Đạt**  **(Leader)** | **Thiều Huy Hoàng** | **Nguyễn Quang Vy** |
| **Linear Regression** |  |  | **x** |
| **SVR** |  |  | **x** |
| **ARIMA** |  | **x** |  |
| **FBPROPHET** |  | **x** |  |
| **LSTM** | **x** |  |  |
| **Bi-LSTM** | **x** |  |  |

# 

# **Reference:**

**[1]:** [**Global Gold price - Historical Data (1979-Present)**](https://www.kaggle.com/datasets/jishnukoliyadan/gold-price-1979-present)

**[2], [3]:** Chicco, D., Warrens, M.J. and Jurman, G., 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, p.e623.

**[4]:** [**Autoregressive Integrated Moving Average - ARIMA**](https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp)

**[5]**: Nguyễn Minh Nhựt – LAB4\_PhanTichDuLieuChuoiThoiGian.

**[6]:** [Data Science - Autoregressive Integrated Moving Average - ARIMA](https://corporatefinanceinstitute.com/resources/data-science/autoregressive-integrated-moving-average-arima/)

**[7]:** [Facebook Prophet Model - Time-series Analysis](https://www.geeksforgeeks.org/time-series-analysis-using-facebook-prophet/)

**[9]:** [Advanced Recurrent Neural Networks](https://gamedevacademy.org/advanced-recurrent-neural-networks/)

**[10]:** [Univariate Time-series with BiLSTM](https://towardsai.net/p/l/univariate-time-series-with-stacked-lstm-bilstm-and-neuralprophet)

**[11]:** Cai, Changchun, Yuan Tao, Tianqi Zhu, and Zhixiang Deng. 2021. "Short-Term Load Forecasting Based on Deep Learning Bidirectional LSTM Neural Network" Applied Sciences 11, no. 17: 8129. <https://doi.org/10.3390/app11178129>

**[12]:** <https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/>

**[13]:** <https://www.saedsayad.com/support_vector_machine_reg.htm>

**[14]:** [Support Vector Regression and it’s Mathematical Implementation | by Priyanka Parashar | The Startup | Medium](https://medium.com/swlh/support-vector-regression-and-its-mathematical-implementation-4800456e4878)

* **Reference Of Related Work:**

[1]. Risse, M., (2019) Combining wavelet decomposition with machine learning to forecast gold returns. International Journal of Forecasting, [online] 352, pp.601–615. Available at: <https://doi.org/10.1016/j.ijforecast.2018.11.008>.

[2].Vidya, G.S. and Hari, V.S., (2020) Gold Price Prediction and Modelling using Deep Learning Techniques. 2020 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2020, pp.28–31.

[3]. Tripathy, N., (2017) Forecasting Gold Price with Auto-Regressive Integrated Moving Average Model. International Journal of Economics and Financial Issues, 74, pp.324–329.

[4]. Guha, B., & Bandyopadhyay, G. (2016). Gold price forecasting using ARIMA model. *Journal of Advanced Management Science*, *4*(2).

[5] .Yang, X. (2019, January). The prediction of gold price using ARIMA model. In *2nd Conference on Social Science, Public Health and Education (SSPHE 2018)* (pp. 273-276). Atlantis Press.

[6]. He, Z., Zhou, J., Dai, H. N., & Wang, H. (2019, August). Gold price forecast based on LSTM-CNN model. In *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)* (pp. 1046-1053). IEEE.