VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF INFORMATION TECHNOLOGY

FINAL PROJECT

BUSINESS ANALYSIS

Instructor:

Assoc. Prof. Nguyễn Đình Thuân

Student performance: Team 3

Phạm Lê Trưởng	20522090
Lương Lý Công Thịnh	20521960
Nguyễn Ngọc Tín	20522015

Ho Chi Minh City, June 21 2023

TEACHER'S COMMENTS

MỤC LỤC

I. INTRODUCTION

Currently, with the world economic situation in general and Vietnam's economy in particular is complicated and difficult to understand and gold price is one of those fluctuations. According to Tuoi Tre newspaper [1], the increase in the world gold price has little impact on the domestic gold price because many sellers sell when the price increases a little, so the price of gold does not increase sharply compared to the world. In addition, according to VTV news [2], the gold price recovered thanks to the weakening USD, leading many people to increase their investment demand in precious metals. It can be seen that the gold market is volatile as Tien Phong newspaper [3] mentioned that when the price of gold increases, buyers still lose because the buying price is higher than the selling price. From the above results, in this project, our team has chosen to predict gold price using machine learning and deep learning models in the hope of helping Vietnam's economy.

II. RELATED WORKS

Gold Price Forecasting Using ARIMA Model [4], Research by Banhi Guha and Gautam Bandyopadhyay. In this paper, they tested six sets of parameters p,d,q for the ARIMA model and the best results were obtained with ARIMA(1,1,1) satisfying the statistical criteria.

Forecasting Gold Price in Rupiah using Multivariate Analysis with LSTM and GRU Neural Networks [5], Research by Sebastianus Bara Primananda and Sani Muhamad Isa. In this paper, the best model to use in gold price forecasting problems over a period of less than three years is GRU. On the other hand, for periods over three years, the LSTM is displayed with greater accuracy. From that information, this study indicates that hyperparameter tuning is more effective in optimizing the LSTM Model than the GRU Model for this study the gold price prediction problem.

Modeling Gold Price via Artificial Neural Network [6], Hossein Mombeini and Abdolreza Yazdani-Chamzini. Research results show that ANN method is a powerful tool for

modeling gold price and can give better forecasting performance than ARIMA method with R square of 0.965.

Gold Price Forecasting Using LSTM, Bi-LSTM and GRU [7], Research by Mustafa Yurtsever on comparing the performance of three multivariate models (LSTM, Bi-LSTM and GRU) to predict gold price using MAE, RMSE and MAPE measures. The results show that LSTM works best with batch size parameters of 128, number of epochs of 1000, resulting in MAPE = 3.18, RMSE = 61,728 and MAE = 48.85

Forecasting Gold Prices Using Temporal Convolutional Networks [8], Research by Justin Fajou and Andrew McCarren, this research undertook a comparative analysis between Temporal Convolutional Networks (TCNs), the current state of the art machine learning approaches and a traditional time series model in gold price prediction. The results demonstrated that the TCNs consistently outperformed the other approaches chosen in this study with result RMSE = 15.26, MAE = 10.05, R2 = 0.9954 and reduced the error by more 27% in comparison with the best preforming non-TCN approach.

Prediction of gold price with ARIMA and SVM [9], Research by D Makala and Z Li, In this research paper, we found out how to predict gold price using Arima and SVM model. The results of this study show that SVM(Poly) with MAPE = 2.49, RMSE = 0.0275 and R2 = 0.9978 is said to perform much better than other SVM(RBF) and Arima models.

III. MODELING

1. LSTM

LSTM (Long Short-Term Memory) model is a specialized neural network architecture widely used in time series processing. This model was introduced by Hochreiter and Schmidhuber in 1997 and has become one of the most important models in the field of deep learning for time series.

The LSTM algorithm was developed to address the issue of long-term information loss in traditional recurrent neural networks (RNNs). In RNNs, information can only propagate

through a limited number of neurons, and it tends to vanish as the length of the sequence increases. LSTM tackles this problem by utilizing a memory cell and gates to regulate the flow of information during the processing of time series.

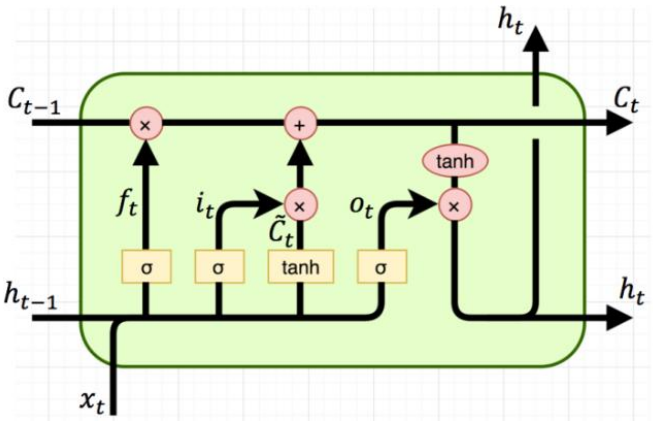

Figure 1: Architecture of the LSTM model

The reason for using the LSTM model in time series processing is its ability to handle long-term dependencies. Thanks to the memory cell and gates, LSTM can retain important information from the past during training, enabling the model to make better predictions on long and complex sequences, the specific formula is as follows[10]:

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

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

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

Memory cell (C): $\tilde{C}_t = tanh(W_C . [h_{t-1}, x_t] + b_C)$

$$C_t = f_t . C_{t-1} + i_t . \tilde{C}_t$$

Hidden state (h): $h_t = o_t . tanh(C_t)$

Where:

x_t is the input at time t.

h_{t-1} is the hidden state from the previous layer.

i_t, f_t, o_t are the values of the gates at time t.

C_t is the memory cell state at time t.

h_t is the hidden state at time t.

W_i, W_f, W_o, W_C are weight matrices.

b_i, b_f, b_o, b_C are bias vectors.

The formulas describe how the gates and states of an LSTM are computed based on the current input and the previous state. This process allows the LSTM to process and store crucial information from the past and influence the prediction outcomes.

2. CNN-LSTM

The CNN-LSTM model is a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers, used for time series problems.

The CNN-LSTM model was developed by leveraging the strengths of both CNN and LSTM. CNN was primarily developed for spatial data processing, such as images, while LSTM was designed for sequence data processing, like time series. When applied to time series problems, the CNN-LSTM model combines the spatial feature extraction capability of CNN with the sequence understanding and prediction capability of LSTM.

The CNN-LSTM model operates by using CNN layers to extract features from the input data and then passing them through LSTM layers to understand and predict the time series. The main steps involved in its operation are as follows:

- Firstly, the time series data is divided into fixed-size windows.
- Each data window is passed through the CNN layers to extract spatial features.
- The output of the CNN layers is then fed into the LSTM layers to understand and predict the time series.
- Finally, the output of the LSTM layers is used to make predictions for future time series values.

The CNN-LSTM model is commonly used in time series problems due to the following benefits:

- Spatial feature extraction: The CNN layers in the model help extract spatial features from the time series data. This is useful for processing structured spatial data like images or sound.
- Sequence understanding and prediction: The LSTM layers in the model have the ability to understand and predict time series data. LSTMs can store long-term information and learn dependencies on previous patterns in the time series.
- Combined advantages: The CNN-LSTM model combines the feature extraction capability of CNN with the sequence understanding and prediction ability of LSTM, improving prediction performance in time series problems compared to using either network alone.

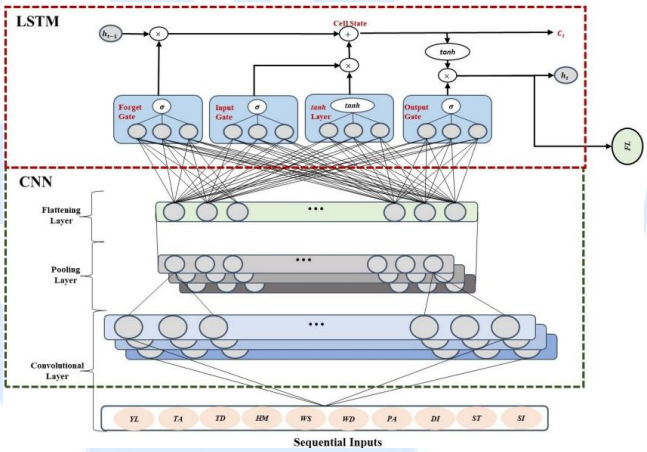

Figure 2: Architecture of the CNN-LSTM model

CNN-LSTM model [11] calculation for time series forecasting:

Convolutional Layers:

- Firstly, the initial time series is divided into fixed-size windows.
- Each window is a 2D array with the vertical dimension representing the length of the time series and the horizontal dimension representing the number of features or channels.

• Apply Convolutional layers on each window to extract spatial features from the time series.

• After applying Convolutional layers, the output of each window becomes a 2D input for the LSTM layer.

Pooling Layer:

• After applying Convolutional layers, you can apply Pooling layers to reduce the spatial size of the input and increase the model's generalization.

• Pooling layers such as Max Pooling or Average Pooling perform downsampling of the input by taking the maximum or average value within a window.

Flatten Layer:

• After applying Pooling layers, the 2D input needs to be flattened into a 1D vector to be fed into the LSTM layer.

• The Flatten layer performs this operation by reshaping the input matrices into a single vector by concatenating the rows of the matrix.

LSTM Layer:

• After applying the Flatten layer, the 1D input is passed through LSTM layers to process the time series and make predictions.

• The LSTM layer has the ability to retain and learn information from the past in the time series for forecasting future values.

• The final output of the LSTM layer can be passed to fully connected layers or other layers to make the final prediction.

3. Gated Recurrent Unit

The GRU model has a gate mechanism to regulate the flow of information so as to remember context in multiple time steps (Cho et al., 2014). It uses an update gate and reset gate to determine what past information can be kept or forgotten. While GRU is similar to LSTM, it combines LSTM's forget and input gates into a single update gateway. The update gate decides how much past information is passed on while the reset gate decides how much

is discarded. Figure 2 shows the structure of a GRU unit. GRU outperforms LSTM in terms of training time and prediction accuracy due to its very simple structure (Jianwei et al., 2019).

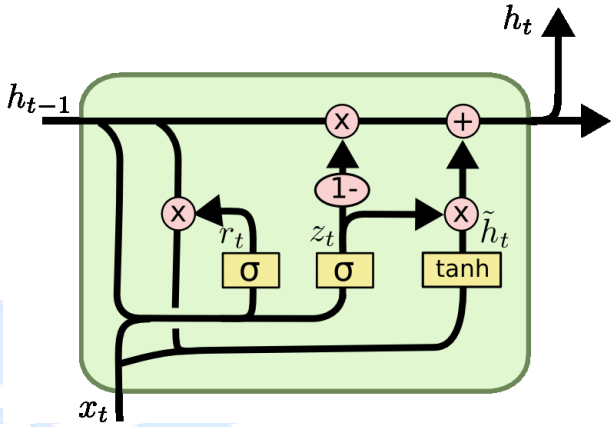

Figure 3: Architecture of the GRU model

There are two gates in the GRU: the update and reset gates. These two gates, which decide what information is transmitted to the output, decide the information to be deleted and transferred. These operations are performed by the following equations[12]:

- $\mathrm{r}_t = \sigma(W_r h_{t-1} + U_r x_t)$
- $\hat{h}_t = tanh(W(r_t * h_{t-1}) + U x_t)$
- $z_t = \sigma(W_z h_{t-1} + U_z x_t)$
- $h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t$

  h_t and h_{t-1} represent the output of the current and previous states, respectively while r_t and z_t indicate the reset and update gates, respectively. a is the logistic sigmoid function while W_r, and U_r are the weight matrices.

These equations describe how the update gate, reset gate, and hidden state of a GRU are computed based on the current input and the previous state. The GRU's gating mechanisms allow it to selectively retain important information and adapt to changing input patterns.

The GRU model offers a trade-off between complexity and performance compared to the LSTM. It is computationally efficient and has fewer parameters, making it suitable for various time series processing tasks, including those involving long-term dependencies.

4. Bayesian Neural Network

In traditional neural networks, each parameter has a fixed value and is determined by the classical error backpropagation. In Bayesian Neural Networks, each parameter has a posterior distribution instead of a fixed value, which is obtained by bayes backpropagation, so that the uncertainty can be introduced to the neural network prediction. Neural network based on bayes backpropagation is shown in image below, where $\mathbf{w}$ is the neural network weight, and the black curves are the neural network connections. The distribution of $\mathbf{w}$ is $P(\mathbf{w}|D)$.

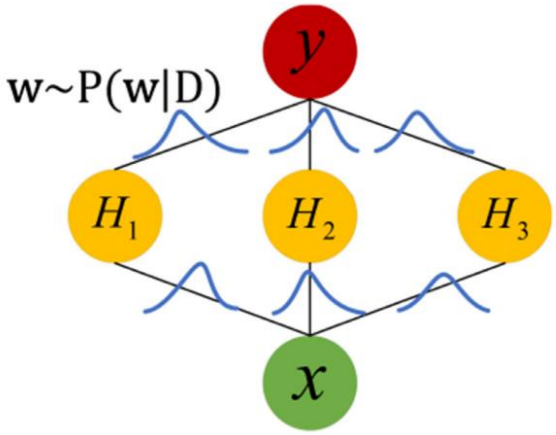

Figure 4: Neural network based on bayes backpropagation

The posterior distribution on weights $P(\mathbf{w}|D)$ cannot be obtained directly, so researchers use the distribution $q(\mathbf{w}|\theta)$ to approximate $P(\mathbf{w}|D)$ through variational learning (Graves, 2011). The variational approximation finds the parameter θ of $q(\mathbf{w}|\theta)$ that minimizes the Kullback-Leibler (KL) divergence with the true bayesian posterior on weights $P(\mathbf{w}|D)$:

$$\theta* = \underset{\theta}{\operatorname{argmin}}\, \mathrm{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w}|D)]$$

$$= \underset{\theta}{\operatorname{argmin}} \int q(\mathbf{w}|\theta)\, log\, \frac{q(\boldsymbol{w}|\theta)}{P(\boldsymbol{w})P(D|\boldsymbol{w})}\, d\mathbf{w}$$

$$= \underset{\theta}{\operatorname{argmin}}\ \mathrm{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w})] - E_{q(\mathbf{w}|\theta)}[\log(D|\mathbf{w})]$$

The loss of BNN is composed of a prior-dependent part and a data dependent part. The prior distribution of parameters is usually an independent gaussian prior with mean μ_o and variance $\sigma^2{}_o$:

$P(\mathbf{w}) = \prod j\ (w_j\ |\mu_o, \sigma_o^2)$

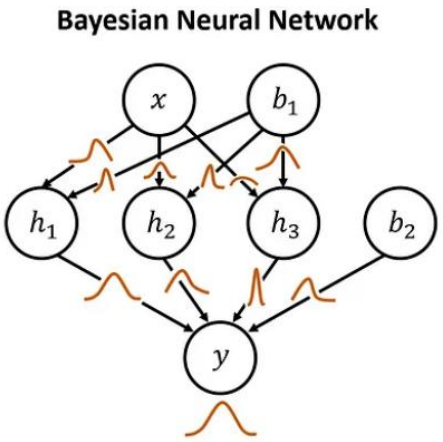

Figure 5: Architecture of the BNN model

5. Hidden Markov Model

The Hidden Markov model (HMM) is a statistical model that was first proposed by Baum L.E. (Baum and Petrie, 1966) and uses a Markov process that contains hidden and unknown parameters.

In this model, the observed parameters are used to identify the hidden parameters. These parameters are then used for further analysis. The HMM is a type of Markov chain. Its state cannot be directly observed but can be identified by observing the vector series. Since the 1980s, HMM has been successfully used for speech recognition, character recognition, and mobile communication techniques. It has also been rapidly adopted in such fields as bioinformatics and fault diagnosis.

The basic principle of HMM is that the observed events have no one-to-one correspondence with states but are linked to states through the probability distribution. It is a

doubly stochastic process, which includes a Markov chain as the basic stochastic process, and describes state transitions and stochastic processes that describe the statistical correspondence between the states and observed values. From the perspective of observers, only the observed value can be viewed, while the states cannot. A stochastic process is used to identify the existence of states and their characteristics. Thus, it is called a "hidden" Markov model.

Statistical methods are used to build state changes in HMM to understand the most possible trends in the surveillance data. HMM can automatically and flexibly adjust the trends, seasonal, covariant, and distributional elements. HMM has been used in many studies on time series surveillance data.

The HMM model has probability of any sequence of observations occurring when following a given sequence of states can be stated as follows[14]:

$$p(x,y) = \prod_{t=1}^{T} \underbrace{p(y_t|y_{t-1})}_{A} \underbrace{p(x_t|y_t)}_{B}$$

in which the probabilities $p(y_t|y_{t-1})$ can be read as the probability of being currently in state y_t given we just were in the state y_{t-1} at the previous instant t-1, and the probability $p(x_t|y_t)$ can be understood as the probability of observing x_t at instant t given we are currently in the state y_t

6. *Linear Regression*

Linear regression is a statistical procedure for calculating the value of a dependent variable from an independent variable. Linear regression measures the association between two variables. It is a modeling technique where a dependent variable is predicted based on one or more independent variables. Linear regression analysis is the most widely used of all statistical techniques.[15]

The formula for a simple linear regression is:[16]

$$y = \beta_0 + \beta_1 x + \epsilon$$

Where:

- y dependent variable
- β_0 is the intercept term.
- β_1 is the regression coefficient
- x is the independent variable
- ϵ is the error ter

7. *ARIMA*

ARIMA models (Autoregressive Integrated Moving Average) were first introduced by George Box and Gwilym Jenkins in the early 1970s. They have since become a fundamental tool in time series analysis and forecasting. ARIMA model is commonly denoted as (p, d, q):

- Auto-Regressive (AR): using a linear combination of past values of the variable. An autoregressive model of order p can be written[16]

  $$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots\cdots + \phi_p Y_{t-p} + e_t$$

  Where Y_t is current value, $\phi_1, \phi_2, \ldots \phi_p$ are model parameters, e_t is random error.

- Integrated (I): refers to the differentiation of the time series data.
- Moving Average (MA): uses past forecast errors in a regression-like model. "q" is the number of previous error values to consider for the forecast[16]

  $$Y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q}.$$

  Where Y_t is current value, e_t is random error and $\theta_1, \theta_2, \ldots \theta_q$ are coefficients, c is constants.

8. *Vector Autoregression*

Vector Autoregressive models were introduced by Sims in 1980, revolutionized time series analysis in economics.. They are widely used for forecasting, analyzing relationships between variables, and have applications in finance and meteorology.

VAR is a forecasting algorithm that can be used when two or more time series influence each other, i.e. the relationship between the time series involved is bi-directional.[17]

The VAR(p) model of order p can be represented in the following formula[18]:

$$Y_t = C + A_1 Y_{t-1} + A_2 Y_{t-2} + \cdots + A_p Y_{t-p} + \mathcal{E}_t$$

Where:

- Y_t is a vector containing the values of the variables at time t.
- C is the intercept.
- A_1 , A_2, ..., A_p are matrices of parameters.
- $\mathcal{E}_t$ is the vector of random errors.

9. *Singular Spectrum Analysis*

SSA (Singular Spectrum Analysis) is a non-parametric time series analysis method used to analyze and predict the components in a time series.After the books that were published by Vautard et al. (1989) SSA became widespread in the field of climatology and the book by Elsner (1996) shortly after, combined the prevalent information from other books related to SSA aiding the scientific communities and research institutions.[19]

The basic SSA methodology as applied to a given time series consists of two important stages[20]:

(1) Decomposition: comprises embedding and singular value decomposition (SVD)

(2) Reconstruction: comprises eigentriple grouping and diagonal averaging.

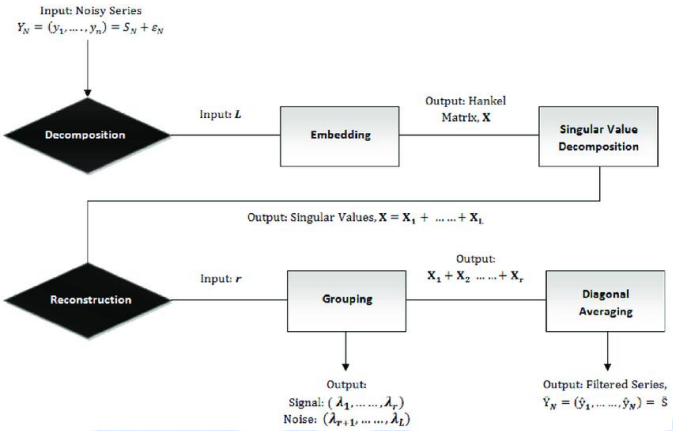

Figure 6: Flowchart of the SSA model

The VAR model operates as follows:

- Embedding step: the observed time series is mapped into a trajectory matrix (Hankel matrix)
- Lagged covariance matrix defined as is decomposed into eigentriplets (equal to win- dow length) by using SVD.
- Choosing suitable eigenvalues and corresponding eigenvectors for trend extraction from SVD and subsequent Hankelization matrix from selected components of the SVD.
- In grouping step, apply eading eigenfunction to select components explain the long-term trends in the original timeseries.
- Final, reconstruction of a time series by averaging the diagonal elements of selected matrices in the grouping stage.
- The PCs and corresponding eigenvectors are then considered to reconstruct the time series trend by the method of diagonal averaging.

IV. METHOD

The problem-solving method needs to go through the process, so we will build a framework diagram to represent the research method.

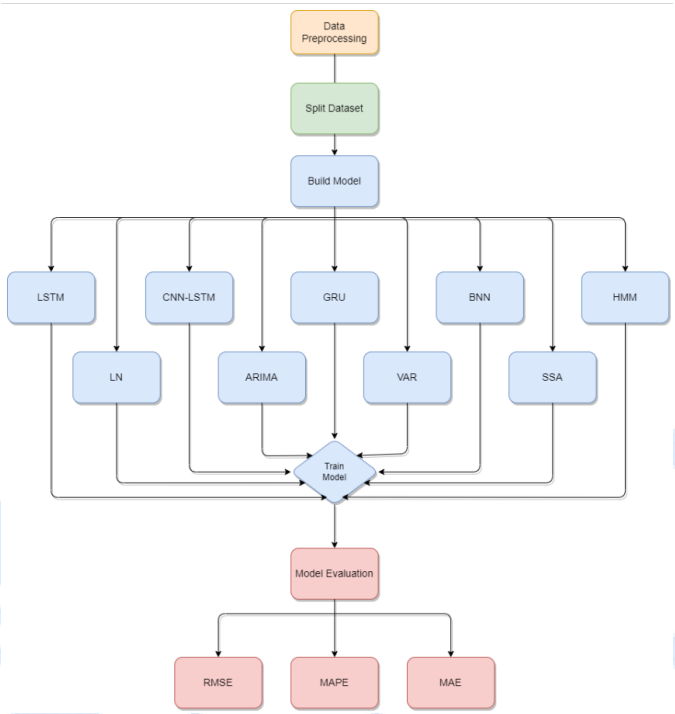

Figure 7: Diagram of research steps

They are divided into steps:

Step 1: Data preprocessing

Step 2: Split the dataset

Step 3: Build the model

Step 4: Train the model

Step 5: Evaluate the model

Model evaluation metrics[21]: RMSE, MAPE, MAE

$$\text{RMSE} = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2\right)}$$

$$\text{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Where N is the number of data points, $\hat{y}_i$ is a predicted value and y_i is a real value

V. EXPERIMENT

1. *First gold price dataset*

A. Data description

The dataset provided by kaggle[X] consists of 2680 rows, and time data columns and multi-country gold price columns.

Date	US dollar (	Euro (EUR	Japanese y	Pound ster	Canadian (	Swiss fran	Indian rup	Chinese re	Turkish lir	Saudi riyal	Indonesiar	UAE dirhar	Thai baht	Vietnames	Egyptian p	Korean
2/1/2012	1531	1179.37	117795.1	985.14	1558.94	1431.64	81303.75	9636.11	2891.45	5741.56	13882343	5623.44	48303.03	32202289	9233.08	1763
3/1/2012	1598	1224.24	122646.5	1022.07	1612.54	1489.74	85037.56	10057.81	2997.13	5992.82	14597730	5869.37	50416.88	33607538	9639.93	1838
4/1/2012	1613	1249.52	123781.6	1033.35	1636.31	1522.43	85416.39	10153.19	3042.44	6048.91	14750885	5924.95	50753.04	33923003	9733.24	1853
5/1/2012	1599	1249.9	123298.9	1032.98	1632.98	1522.57	84722.99	10076.42	3011.96	5996.73	14590875	5873.29	50616.34	33628569	9651.96	1843
6/1/2012	1616.5	1271.43	124656.4	1049.16	1655.7	1545.94	85221.86	10199.31	3036.19	6062.2	14702068	5937.4	51121.81	34000653	9757.6	1879
9/1/2012	1615	1267.91	124112.7	1045.61	1657.88	1538.93	84795.56	10198.08	3025.62	6056.49	14777250	5931.89	51292.39	33970718	9750.56	1879
10/1/2012	1637	1281.11	125713.4	1057.05	1667.53	1554.99	84641.06	10337.65	3055.62	6138.83	14986735	6012.62	51835.59	34406466	9883.39	1893
11/1/2012	1634.5	1288.38	125725.7	1064.47	1668.74	1561.11	84838.72	10322.68	3044.99	6129.7	14972020	6003.52	51895.38	34374352	9867.48	1893
12/1/2012	1661	1297.96	127490	1084.27	1695.22	1571.31	85715.9	10493.87	3076.84	6229.33	15214760	6101.27	52869.61	34939135	10031.61	1923
13/1/2012	1635.5	1291.2	125900.8	1070	1675.81	1562.15	84285.47	10314.44	3043.66	6133.7	14850340	6007.27	52017.07	34400289	9874.33	1877
16/1/2012	1641	1294.93	125881.1	1070.63	1670.78	1564.94	84289.95	10365.38	3040.77	6154.49	14998740	6027.47	52339.68	34490538	9910	1895

Figure 8: Original dataset

B. Data preprocessing

- Keep the time series column and the gold price column of Viet Nam then normalize and rearrange the time series column in order
- Descriptive statistics:

Price (VND)	
Mean	32001941.83
Standard Error	113441.0681
Median	29742172.1
Mode	34542300
Standard Deviation	5872698.485
Sample Variance	3.44886E+13
Kurtosis	-0.650201655
Skewness	0.756642623
Range	24315676.1
Minimum	23588455.74
Maximum	47904131.84
Sum	85765204116
Count	2680

Figure 9: Descriptive statistics

- Visualize data

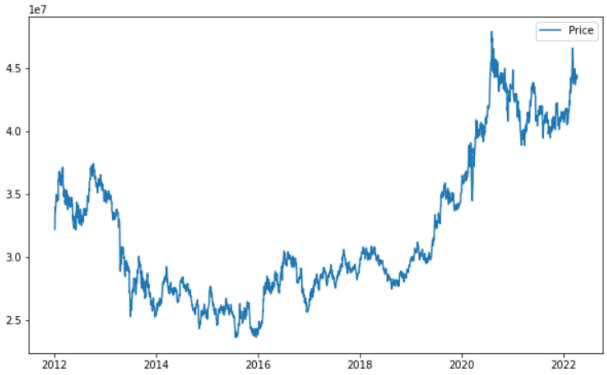

Figure 10: Dataset visualization

C. Split the dataset

We divide into 3 datasets: train data, test data and valid data in the ratio of 6:3:1 and 7:2:1 and 8:1:1 respectively.

D. Building model

1. LSTM

Architecture of the LSTM model

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100, 128)	66560
lstm_1 (LSTM)	(None, 64)	49408
dense (Dense)	(None, 25)	1625
dense_1 (Dense)	(None, 1)	26

Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0

Figure 11: Summary of LSTM model

2. CNN-LSTM

Layer (type)	Output Shape	Param #
conv_lstm1d (ConvLSTM1D)	(None, 50, 64)	16896
flatten (Flatten)	(None, 3200)	0
repeat_vector (RepeatVector)	(None, 1, 3200)	0
lstm (LSTM)	(None, 1, 200)	2720800
dense (Dense)	(None, 1, 100)	20100
dense_1 (Dense)	(None, 1, 1)	101

Total params: 2,757,897
Trainable params: 2,757,897
Non-trainable params: 0

Figure 12: Summary of CNN-LSTM model

3. GRU

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 100, 64)	12864
gru_1 (GRU)	(None, 64)	24960
dense (Dense)	(None, 1)	65

Total params: 37,889
Trainable params: 37,889
Non-trainable params: 0

Figure 13: Summary of GRU model

4. BNN

```
Sequential(
  (0): BayesLinear(prior_mu=0, prior_sigma=0.1, in_features=100, out_features=64, bias=True)
  (1): ReLU()
  (2): BayesLinear(prior_mu=0, prior_sigma=0.1, in_features=64, out_features=1, bias=True)
)
```

Figure 14: Summary of BNN model

5. HMM

```
GaussianHMM(covariance_type='full',
n_components=10, random_state=42)
```

Figure 15: Summary of HMM model

6. LN

```
Coefficients to independent variables: [-3722.43813541]
Intercept (bias): 31885002.791497204
```

Figure 16: Coefficient and intercept of LN model

7. ARIMA

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1608
Model:               SARIMAX(1, 1, 0)   Log Likelihood              -22456.834
Date:                Wed, 21 Jun 2023   AIC                        44917.669
Time:                        03:06:18   BIC                        44928.433
Sample:                             0   HQIC                       44921.665
                                 - 1608
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.0381      0.012      3.190      0.001       0.015       0.061
sigma2      8.032e+10   1.94e-14   4.14e+24      0.000    8.03e+10    8.03e+10
===================================================================================
Ljung-Box (L1) (Q):                   3.28   Jarque-Bera (JB):              6015.53
Prob(Q):                              0.07   Prob(JB):                         0.00
Heteroskedasticity (H):               0.40   Skew:                            -0.57
Prob(H) (two-sided):                  0.00   Kurtosis:                        12.41
===================================================================================
```

Figure 17: Summary of ARIMA model

8. VAR

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 21, Jun, 2023
Time:                     02:56:59
--------------------------------------------------------------------
No. of Equations:         2.00000    BIC:                   -10.7628
Nobs:                     1603.00    HQIC:                  -10.8092
Log likelihood:           4158.47    FPE:                1.96652e-05
AIC:                     -10.8367    Det(Omega_mle):     1.93981e-05
--------------------------------------------------------------------
```

Figure 18: Summary of VAR model

9. SSA

```
----------------------------------------
EMBEDDING SUMMARY:
Embedding dimension     :  84
Trajectory dimensions   : (84, 1525)
Complete dimension      : (84, 1525)
Missing dimension       : (84, 0)
```

Figure 19: Summary of embedding step

```
----------------------------------------
DECOMPOSITION SUMMARY:
Rank of trajectory              : 84
Dimension of projection space   : 5
Characteristic of projection    : 0.9998
```

Figure 20: Summary of decomposition step

E. Evaluate the model

Model	Train: Test:Val	RMSE	MAPE (%)	MAE
LSTM	6:3:1	**396747**	**0.73**	**269878**
	7:2:1	450611	0.78	312180
	8:1:1	497241	0.87	376123
CNN-LSTM	6:3:1	912149	1.78	666847
	7:2:1	892371	1.75	697204
	8:1:1	940909	1.74	748217
GRU	6:3:1	516125	0.93	348108
	7:2:1	637469	1.14	456499
	8:1:1	548157	0.96	411844
BNN	6:3:1	595907	1.16	427409
	7:2:1	813854	1.67	655251
	8:1:1	638168	1.21	519933
HMM	6:3:1	14646296	29.55	11910256
	7:2:1	6427139	13.27	5468529
	8:1:1	4772260	9.35	3974343
LN	6:3:1	12445516	28.11	10602195
	7:2:1	12712022	29.36	11612709
	8:1:1	12780731	29.66	12581605
ARIMA	6:3:1	7607040	14.20	5612101
	7:2:1	9175055	19.31	7819175
	8:1:1	8026028	18.05	7704910
VAR	6:3:1	6352238	30.55	3807911
	7:2:1	6227212	21.97	3708145

	8:1:1	3622752	11.18	2280883
SSA	6:3:1	9129563	17.24	6809718
	7:2:1	10688670	23.07	9286702
	8:1:1	6956563	15.38	6579275

On the first gold price dataset, the best LSTM model results with the ratio of train, test, valid is 6:3:1 so we use this model to forecast the next 30 days

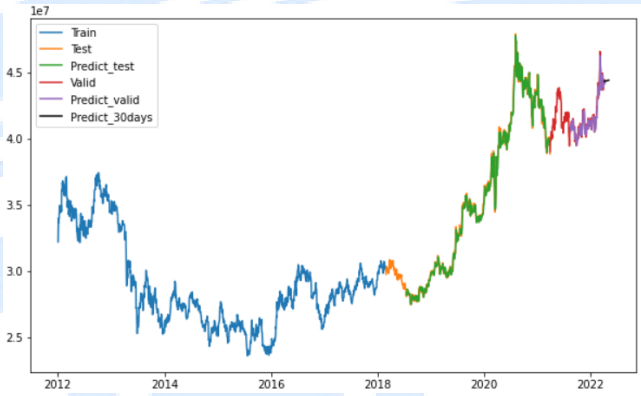

Figure 21: Visualize forecast results using LSTM model with first gold price dataset

2. *Second gold price dataset*

A. Data description

This dataset is provided by kaggle with 2290 rows and 6 columns, where one column is the time series and the other 5 columns are the values of SPX, GLD USO, SLV, EUR/USD, here we do gold price forecast should use only GLD (GOLD)

Date	SPX	GLD	USO	SLV	EUR/USD
1/2/2008	1447.16	84.86	78.47	15.18	1.471692
1/3/2008	1447.16	85.57	78.37	15.285	1.474491
1/4/2008	1411.63	85.13	77.31	15.167	1.475492
1/7/2008	1416.18	84.77	75.5	15.053	1.468299
1/8/2008	1390.19	86.78	76.06	15.59	1.557099
1/9/2008	1409.13	86.55	75.25	15.52	1.466405

Figure 22: Original dataset

B. Data preprocessing

- Process and return to the form of a column of time series and a column of the gold price keep the time series column and the gold price (GLD) column then normalize and put the time series in the correct order
- Descriptive statistics:

	GLD
Mean	122.7328751
Standard Error	0.48655019
Median	120.580002
Mode	115.940002
Standard Deviation	23.28334575
Sample Variance	542.1141892
Kurtosis	-0.27508052
Skewness	0.334138347
Range	114.589996
Minimum	70
Maximum	184.589996
Sum	281058.2839
Count	2290

Figure 23: Descriptive statistics

- Visualize data

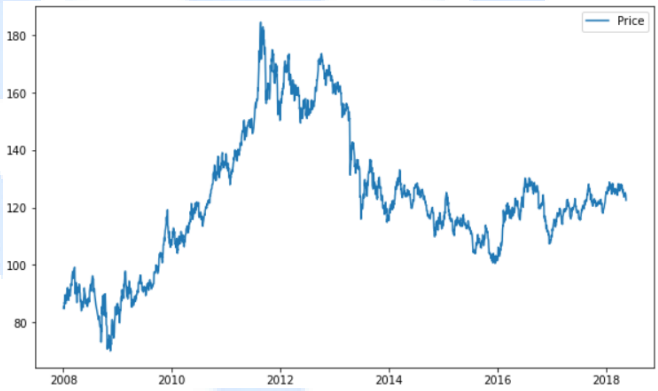

Figure 24: Dataset visualization

C. Split the dataset

We divide into 3 datasets: train data, test data and valid data in the ratio of 6:3:1 and 7:2:1 and 8:1:1 respectively.

D. Building model

1. LSTM

Architecture of the LSTM model

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100, 128)	66560
lstm_1 (LSTM)	(None, 64)	49408
dense (Dense)	(None, 25)	1625
dense_1 (Dense)	(None, 1)	26

Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0

Figure 25: Summary of LSTM model

2. CNN-LSTM

Architecture of the CNN-LSTM model:

Layer (type)	Output Shape	Param #
conv_lstm1d (ConvLSTM1D)	(None, 50, 64)	16896
flatten (Flatten)	(None, 3200)	0
repeat_vector (RepeatVector)	(None, 1, 3200)	0
lstm (LSTM)	(None, 1, 200)	2720800
dense (Dense)	(None, 1, 100)	20100
dense_1 (Dense)	(None, 1, 1)	101

Total params: 2,757,897
Trainable params: 2,757,897
Non-trainable params: 0

Figure 26: Summary of CNN-LSTM model

3. GRU

```
 Layer (type)                 Output Shape              Param #
=================================================================
 gru (GRU)                    (None, 100, 64)           12864

 gru_1 (GRU)                  (None, 64)                24960

 dense (Dense)                (None, 1)                 65

=================================================================
Total params: 37,889
Trainable params: 37,889
Non-trainable params: 0
_________________________________________________________________
```

Figure 27: Summary of GRU model

4. BNN

```
Sequential(
  (0): BayesLinear(prior_mu=0, prior_sigma=0.1, in_features=100, out_features=64, bias=True)
  (1): ReLU()
  (2): BayesLinear(prior_mu=0, prior_sigma=0.1, in_features=64, out_features=1, bias=True)
)
```

Figure 28: Summary of BNN model

5. HMM

```
GaussianHMM(covariance_type='full',
n_components=10, random_state=42)
```

Figure 29: Summary of HMM model

6. LN

```
Coefficients to independent variables: [0.05426891]
Intercept (bias): 88.53294886393014
```

Figure 30: Coefficient and intercept of LN model

7. ARIMA

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1374
Model:               SARIMAX(2, 1, 2)   Log Likelihood               -2735.163
Date:                Wed, 21 Jun 2023   AIC                           5480.327
Time:                        03:08:29   BIC                           5506.451
Sample:                             0   HQIC                          5490.102
                                - 1374
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.3277      0.038    -34.568      0.000      -1.403      -1.252
ar.L2         -0.9130      0.039    -23.619      0.000      -0.989      -0.837
ma.L1          1.3101      0.035     37.117      0.000       1.241       1.379
ma.L2          0.9295      0.036     26.173      0.000       0.860       0.999
sigma2         3.1463      0.045     69.411      0.000       3.057       3.235
===================================================================================
Ljung-Box (L1) (Q):                   0.11   Jarque-Bera (JB):             10754.12
Prob(Q):                              0.74   Prob(JB):                         0.00
Heteroskedasticity (H):               1.27   Skew:                            -1.31
Prob(H) (two-sided):                  0.01   Kurtosis:                        16.46
===================================================================================
```

Figure 31: Summary of ARIMA model

8. VAR

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 21, Jun, 2023
Time:                     02:58:57
--------------------------------------------------------------------
No. of Equations:         2.00000    BIC:                    1.59249
Nobs:                     1362.00    HQIC:                   1.47267
Log likelihood:          -4769.26    FPE:                    4.05922
AIC:                      1.40098    Det(Omega_mle):         3.91421
--------------------------------------------------------------------
```

Figure 32: Summary of VAR model

9. SSA

```
----------------------------------------
EMBEDDING SUMMARY:
Embedding dimension     :  84
Trajectory dimensions   : (84, 1291)
Complete dimension      : (84, 1291)
Missing dimension       : (84, 0)
```

Figure 33: Summary of embedding step

```
----------------------------------------
DECOMPOSITION SUMMARY:
Rank of trajectory              : 84
Dimension of projection space   : 6
Characteristic of projection    : 0.9997
```

Figure 34: Summary of decomposition step

E. Evaluation the model

Model	Train:Test:Val	RMSE	MAPE(%)	MAE
LSTM	6:3:1	1.2	0.76	0.9
	7:2:1	1.2	0.79	0.9
	8:1:1	**0.9**	**0.6**	**0.7**
CNN-LSTM	6:3:1	1.7	1.16	1.3
	7:2:1	1.6	1.05	1.2

	8:1:1	1.5	1.04	1.2
GRU	6:3:1	1.26	0.08	0.96
	7:2:1	1.56	1.06	1.24
	8:1:1	1.03	0.9	1.04
BNN	6:3:1	2.65	1.8	2.1
	7:2:1	2.58	1.76	2.07
	8:1:1	2.37	1.77	2.02
HMM	6:3:1	31.77	22.39	25.71
	7:2:1	31.26	23.5	27.26
	8:1:1	14.04	9.6	11.74
LN	6:3:1	233	11.32	211
	7:2:1	77	3.23	58
	8:1:1	95	4.28	77
ARIMA	6:3:1	12	8.90	10
	7:2:1	7	5.27	6
	8:1:1	7	3.81	5
VAR	6:3:1	25	81.82	23
	7:2:1	19	60.35	17
	8:1:1	9	19.98	7
SSA	6:3:1	23	19.38	22
	7:2:1	7	4.98	6
	8:1:1	7	4.3	5

On the second gold price dataset, the best LSTM model results with the ratio of train, test, valid is 8:1:1 so we use this model to forecast the next 30 days

Figure 35: Visualize forecast results using LSTM model with second gold price dataset

3. *Third gold price dataset*

A. Data description

The dataset provided by Yahoo Finance contains 1762 rows and is a collection of data on gold futures prices on the COMEX market in the United States, with the ticker symbol GC=F.

Date	Open	High	Low	Close	Adj Close	Volume
03/05/2016	1292.7	1301.5	1284.5	1290.7	1290.7	151
04/05/2016	1286.9	1290	1273	1273.3	1273.3	109
05/05/2016	1282.9	1286	1270.8	1271.4	1271.4	330
06/05/2016	1278.6	1295.6	1276.9	1292.9	1292.9	305
09/05/2016	1286.5	1286.5	1262.3	1265.6	1265.6	341
10/05/2016	1265.1	1267.4	1258.7	1263.9	1263.9	23
11/05/2016	1266.9	1279.2	1266.9	1274.6	1274.6	50
12/05/2016	1267	1281.2	1264	1270.3	1270.3	101
13/05/2016	1271.4	1275.7	1264.6	1271.9	1271.9	66
16/05/2016	1272.8	1287.8	1272.8	1273.4	1273.4	44
17/05/2016	1272.5	1281.6	1270.8	1276.2	1276.2	16
18/05/2016	1276.4	1276.4	1256.8	1273.7	1273.7	17
19/05/2016	1248	1255.5	1247.5	1254.2	1254.2	69
20/05/2016	1256.6	1256.6	1252.4	1252.4	1252.4	44
23/05/2016	1251.6	1251.6	1247.5	1251.1	1251.1	56
24/05/2016	1240	1240	1228.2	1228.9	1228.9	20
25/05/2016	1219.3	1223.5	1218.6	1223.5	1223.5	5

Figure 36: Original dataset

B. Data preprocessing

- Descriptive statistics:

A	B	C	D
Open	*v*	*Close*	
Mean	1540.356786	Mean	1540.42226
Standard Error	6.404344131	Standard Error	6.408336123
Median	1497.5	Median	1498.699951
Mode	1279.400024	Mode	1265.599976
Standard Deviation	268.7536296	Standard Deviation	268.9211506
Sample Variance	72228.51341	Sample Variance	72318.58526
Kurtosis	-1.598661638	Kurtosis	-1.599599234
Skewness	0.153003116	Skewness	0.151998027
Range	926.7000732	Range	923.6999512
Minimum	1126.900024	Minimum	1127.800049
Maximum	2053.600098	Maximum	2051.5
Sum	2712568.301	Sum	2712683.6
Count	1761	Count	1761

Figure 37: Descriptive statistics

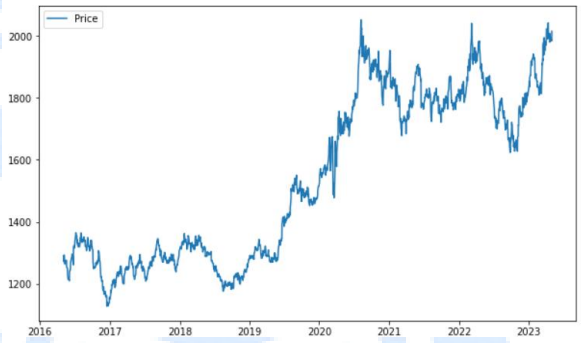

Figure 38: Dataset visualization

C. Split the dataset

We divide into 3 datasets: train data, test data and valid data in the ratio of 6:3:1 and 7:2:1 and 8:1:1 respectively.

D. Building model

1. LSTM

Architecture of the LSTM model

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100, 128)	66560
lstm_1 (LSTM)	(None, 64)	49408
dense (Dense)	(None, 25)	1625
dense_1 (Dense)	(None, 1)	26

Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0

Figure 39: Summary of LSTM model

2. CNN-LSTM

   Architecture of the CNN-LSTM model:

Layer (type)	Output Shape	Param #
conv_lstm1d (ConvLSTM1D)	(None, 50, 64)	16896
flatten (Flatten)	(None, 3200)	0
repeat_vector (RepeatVector)	(None, 1, 3200)	0
lstm (LSTM)	(None, 1, 200)	2720800
dense (Dense)	(None, 1, 100)	20100
dense_1 (Dense)	(None, 1, 1)	101

Total params: 2,757,897
Trainable params: 2,757,897
Non-trainable params: 0

Figure 40: Summary of CNN-LSTM model

3. GRU

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 100, 64)	12864
gru_1 (GRU)	(None, 64)	24960
dense (Dense)	(None, 1)	65

Total params: 37,889
Trainable params: 37,889
Non-trainable params: 0

Figure 41: Summary of GRU model

4. BNN

```
Sequential(
  (0): BayesLinear(prior_mu=0, prior_sigma=0.1, in_features=100, out_features=64, bias=True)
  (1): ReLU()
  (2): BayesLinear(prior_mu=0, prior_sigma=0.1, in_features=64, out_features=1, bias=True)
)
```

Figure 42: Summary of CNN-LSTM model

5. HMM

```
GaussianHMM(covariance_type='full',
n_components=10, random_state=42)
```

Figure 43: Summary of HMM model

6. LN

```
Coefficients to independent variables: [0.35097937]
Intercept (bias): 1161.8531773287968
```

Figure 44: Coefficient and intercept of LN model

7. ARIMA

SARIMAX Results

Dep. Variable:	y	No. Observations:	1056
Model:	SARIMAX(0, 1, 0)	Log Likelihood	-4151.583
Date:	Wed, 21 Jun 2023	AIC	8305.165
Time:	03:09:34	BIC	8310.127
Sample:	0	HQIC	8307.046
	- 1056		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	153.3055	2.737	56.003	0.000	147.940	158.671

Ljung-Box (L1) (Q):	0.23	Jarque-Bera (JB):	4301.69
Prob(Q):	0.63	Prob(JB):	0.00
Heteroskedasticity (H):	2.81	Skew:	0.44
Prob(H) (two-sided):	0.00	Kurtosis:	12.85

Figure 45: Summary of ARIMA model

8. VAR

Summary of Regression Results

Model:	VAR		
Method:	OLS		
Date:	Wed, 21, Jun, 2023		
Time:	03:00:05		
No. of Equations:	2.00000	BIC:	8.69710
Nobs:	1048.00	HQIC:	8.59731
Log likelihood:	-7413.15	FPE:	5096.77
AIC:	8.53636	Det(Omega_mle):	4935.36

Figure 46: Summary of VAR model

9. SSA

```
----------------------------------------
EMBEDDING SUMMARY:
Embedding dimension     :  84
Trajectory dimensions   : (84, 973)
Complete dimension      : (84, 973)
Missing dimension       : (84, 0)
```

Figure 47: Summary of embedding step

```
----------------------------------------
DECOMPOSITION SUMMARY:
Rank of trajectory              : 84
Dimension of projection space   : 3
Characteristic of projection    : 0.9998
```

Figure 48: Summary of decomposition step

E. Evaluation the model

Model	Train:Test:Val	RMSE	MAPE(%)	MAE
LSTM	6:3:1	30.5	1.27	23.0
	7:2:1	31.7	1.36	25.0
	8:1:1	**15.7**	**0.66**	**11.8**
CNN-LSTM	6:3:1	62.0	2.93	52.7
	7:2:1	35.9	1.51	27.7
	8:1:1	29.6	1.34	23.9
GRU	6:3:1	20.59	0.83	15.23
	7:2:1	20.61	0.88	16.02
	8:1:1	25.08	1.12	20.08
BNN	6:3:1	72.83	3.45	65.08
	7:2:1	33.97	1.53	27.71
	8:1:1	71.87	3.49	65.03
HMM	6:3:1	424.27	21.75	392.34
	7:2:1	148.42	6.7	121.81
	8:1:1	218.24	10.21	182.37
LN	6:3:1	233	11.33	211

	7:2:1	76	3.23	59
	8:1:1	95	4.28	77
ARIMA	6:3:1	80	3.33	62
	7:2:1	126	5.87	108
	8:1:1	98	4.11	78
VAR	6:3:1	751	33.16	603
	7:2:1	77	3.07	57
	8:1:1	102	4.42	83
SSA	6:3:1	252	11.98	217
	7:2:1	81	3.35	62
	8:1:1	83	3.62	67

On the third gold price dataset, the best LSTM model results with the ratio of train, test, valid is 8:1:1 so we use this model to forecast the next 30 days

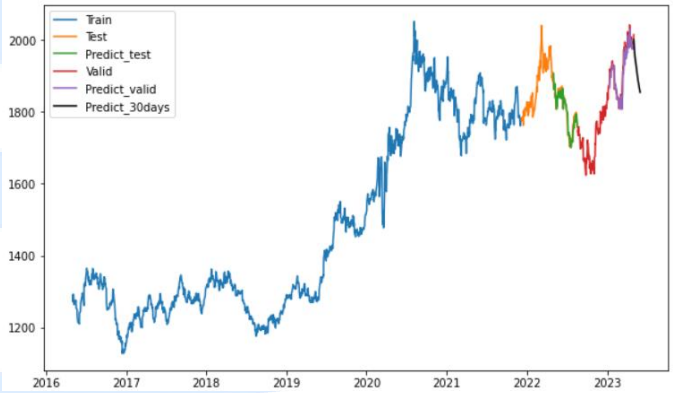

Figure 50: Visualize forecast results using LSTM model with third gold price dataset

VI. CONCLUSION

After testing nine algorithms: LSTM, CNN-LSTM, GRU, BNN, HMM, LN, ARIMA, VAR, SSA on three different datasets on gold prices, we found that the **LSTM** model is the

best and gives the best results. superior results based on three measures RMSE, MAPE, MAE on all nine algorithms.

There are also some difficulties such as the training time of the model, the number of parameters used, so in the future we will improve the model for good and optimal results in terms of training time. the number of parameters used, learn new architectures such as transformer, attention, bert, .. to apply improved results

VII. APPLICATION DEPLOYMENT

After studying and testing the above nine models, we will use the best model is LSTM to build web applications.

Step 1: allows users to select excel or csv files to upload:

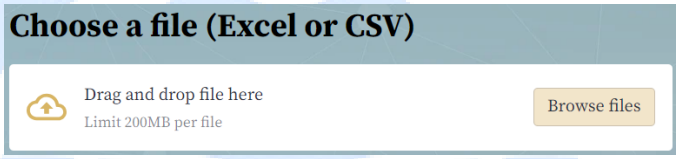

Figure 51: Upload file

Step 2: user selects a time series column and a price series column then clicks 'Train model'

Figure 52: The two column option is the time series and the price series

Step 3: After the training model is completed, it will show the user three evaluation measures RMSE, MAPE(%), MAE along with options to predict the next days such as 5 days, 10 days, 15 days, ... 30 days. A graph that visualizes the resulting data for the above options

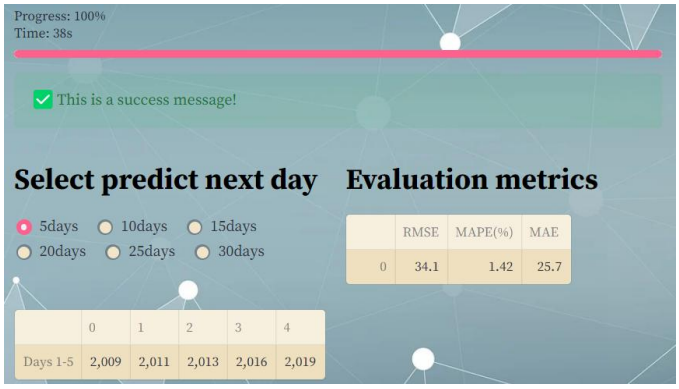

Figure 53: Price prediction results for the next 5 days and three evaluation metrics

Figure 54: Visual results forecast model next 5 days

VIII. GROUP WORK DISTRIBUTION

Member / Work	Phạm Lê Trưởng (Leader)	Lương Lý Công Thịnh	Nguyễn Ngọc Tín
Problem statement	✓	✓	✓

Build a report template	✓	✓	✓
Data collection	✓	✓	✓
Visualizing data	✓	✓	✓
Data Analysis	✓	✓	✓
Data preprocessing	✓	✓	✓
Models research	✓	✓	✓
LSTM, CNN-LSTM model	✓		
GRU, BNN, HMM model		✓	
LN, ARIMA, VAR, SSA model			✓
Compiling codes	✓	✓	✓
Write report	✓	✓	✓
Build web applications	✓		

Summarize and edit reports	✓		

REFERENCE

[1] Vàng Thế Giới tăng mạnh, Vàng Việt Nam không đu theo (2023) TUOI TRE ONLINE. Available at: https://tuoitre.vn/vang-the-gioi-tang-manh-vang-viet-nam-khong-du-theo-20230504193607667.htm (Accessed: 21 June 2023).

[2] Baodientuvtv (2023) Đồng USD Suy Yếu, Giá Vàng Phục Hồi, BAO DIEN TU VTV. Available at: https://vtv.vn/kinh-te/dong-usd-suy-yeu-gia-vang-phuc-hoi-20230421102057128.htm (Accessed: 21 June 2023).

[3] Nga, Q. (2023) Giá vàng bật tăng, Người mua Vẫn lỗ, Báo điện tử Tiền Phong. Available at: https://tienphong.vn/gia-vang-bat-tang-nguoi-mua-van-lo-post1539111.tpo (Accessed: 21 June 2023).

[4] Bandyopadhyay, G. (2016) 'Gold price forecasting using Arima model', Journal of Advanced Management Science, pp. 117–121. doi:10.12720/joams.4.2.117-121.

[5] Primananda, S.B. and Isa, S.M. (2021) 'Forecasting gold price in rupiah using multivariate analysis with LSTM and Gru Neural Networks', Advances in Science, Technology and Engineering Systems Journal, 6(2), pp. 245–253. doi:10.25046/aj060227.

[6] Mombeini, H. and Yazdani-Chamzini, A. (2015) 'Modeling gold price via Artificial Neural Network', Journal of Economics, Business and Management, 3(7), pp. 699–703. doi:10.7763/joebm.2015.v3.269.

[7] YURTSEVER, M. (2021) 'Gold price forecasting using LSTM, Bi-LSTM and gru', European Journal of Science and Technology [Preprint]. doi:10.31590/ejosat.959405.

[8] Justin , F. and Andrew , M. (no date) Forecasting gold prices using temporal convolutional networks - CEUR-WS.org. Available at: https://ceur-ws.org/Vol-3105/paper18.pdf (Accessed: 21 June 2023).

[9] Makala, D. and Li, Z. (2021) 'Prediction of gold price with Arima and SVM', Journal of Physics: Conference Series, 1767(1), p. 012022. doi:10.1088/1742-6596/1767/1/012022.

[10] Christopher , O. (no date) Understanding LSTM networks, Understanding LSTM Networks -- colah's blog. Available at: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (Accessed: 21 June 2023).

[11] Phyo, P.P. and Byun, Y.-C. (2021) 'Hybrid ensemble deep learning-based approach for time series Energy prediction', Symmetry, 13(10), p. 1942. doi:10.3390/sym13101942.

[12] YURTSEVER, M. (2021) 'Gold price forecasting using LSTM, Bi-LSTM and gru', European Journal of Science and Technology [Preprint]. doi:10.31590/ejosat.959405.

[13] Zhang, X., Zou, Y. and Li, S. (2022) 'Bayesian neural network with Efficient Priors for online quality prediction', Digital Chemical Engineering, 2, p. 100008. doi:10.1016/j.dche.2021.100008.

[14] Lan, Y. et al. (2017) 'Development of early warning models', Early Warning for Infectious Disease Outbreak, pp. 35–74. doi:10.1016/b978-0-12-812343-0.00003-5.

[15] Kumari, K. and Yadav, S. (2018) 'Linear Regression Analysis Study', Journal of the Practice of Cardiovascular Sciences, 4(1), p. 33. doi:10.4103/jpcs.jpcs_8_18.

[16] Hyndman, R.J. and Athanasopoulos, G. (2021) Forecasting: Principles and practice. Melbourne: OTexts. Available at: https://otexts.com/fpp2/AR.html.

[17] Gupta, A. (2021) Vector auto-regressive (VAR) models for multivariate time series forecasting, Medium. Available at: https://medium.com/geekculture/vector-auto-regressive-var-models-for-multivariate-time-series-forecasting-106bb6f74add (Accessed: 21 June 2023).

[18] Lemya , T.A. (2021) Forecasting time series using Vector Autoregressive Model [Preprint]. doi:10.22075/IJNAA.2022.5521.

[19] Wadekar, S. et al. (2022) 'A review on singular spectrum analysis', 2022 IEEE International Conference on Current Development in Engineering and Technology (CCET) [Preprint]. doi:10.1109/ccet56606.2022.10080082.

[20] Aswathaiah, U. and Nandagiri, L. (2020) 'Extraction of nonlinear trends in time series of rainfall using singular spectrum analysis', Journal of Hydrologic Engineering, 25(12). doi:10.1061/(asce)he.1943-5584.0002017.

[21] M, P. (2023) End-to-end introduction to evaluating Regression Models, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/ (Accessed: 21 June 2023).