TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

# "Deep learning Approaches for Predicting Gold, Silver, and Platinum Prices"

## TEAM 16

Business
Data
Analysis

### MEMBERS

Phạm Huy Hùng - 20521372

Nguyễn Văn Khang - 20521438

Đặng Trần Tuấn Anh - 20521058

# Deep Learning Approaches for Predicting Gold, Silver, and Platinum Prices

1st Nguyen Van Khang
IS403.N22.HTCL
University of Information Technology
Ho Chi Minh City
20521438@gm.uit.edu.vn

2st Pham Huy Hung
IS403.N22.HTCL
University of Information Technology
Ho Chi Minh City
20521372@gm.uit.edu.vn

3st Dang Tran Tuan Anh
IS403.N22.HTCL
University of Information Technology
Ho Chi Minh City
20521058@gm.uit.edu.vn

***Abstract*—In the context of an increasingly developed market economy, more and more people are interested in the prices of precious metals. Since precious metals market trends are complex and volatile, investors can easily lose their investment. Along with that, they carry with them the attractive value of jewelry and luxury. Therefore, in this research, our team hopes to create a sustainable, error-free, risk-free and interactive tool. can freely interact using models ARIMA, GRU, RF, LN, LSTM, HNN, RNN, NNAR, MRNN to compare stock prices and draw conclusions.***

***Keywords*: ARIMA, GRU, RF, LN, LSTM, HNN, RNN, NNAR, MRNN.**

## I. INTRODUCTION

This study focuses on predicting the prices of gold, silver, and platinum, which are crucial metals in the financial and investment markets. Predicting the prices of these metals poses a significant challenge due to the influence of complex factors such as politics, economics, and global market conditions.

Our dataset consists of real-world data, encompassing the historical prices of gold, silver, and platinum over an extended period. This data is organized as time series, including information on the opening, highest, lowest, and closing prices. Additionally, we have gathered various economic indicators and market factors to examine their correlations and impacts on the prices of these precious metals.

We have utilized a range of algorithms and models to predict the prices of gold, silver, and platinum. These algorithms include ARIMA, RNN, GRU, Linear Regression, VAR, Random Forest, Hybrid Neural Networks (HNN), Neural Network Autoregression (NNAR), and Multiscale Recurrent Neural Networks (MRNN). Each algorithm employs its unique approach to handle time series data and correlate various factors.

During the research process, we conducted data preprocessing, trained the models, and evaluated their performance on the test dataset. Performance metrics such as mean error and correlation coefficient were employed to assess the accuracy of the prediction models.

The findings of this study hold significant value for investors, financial experts, and risk management organizations. By understanding and predicting the trends and price fluctuations of these metals, they can make informed decisions and optimize their investment strategies to maximize profits and minimize risks in the precious metals market.

## II. RELATED WORK

In Dias Satria's research he concludes that the ARIMA Box-Jenkins modeling is unsuitable for predicting stock prices. This is due to the data's nonlinear characteristics, which causes the assumption of white noise in the estimation of the ARIMA Box-Jenkins parameter to be violated. Also, he compared three models: RNN, LSTM, GRU, within GRU presented the best performance in the case of predicting the stock prices based on RMSE value [1].

S.L. Ho et al. identified both the ARIMA and the recurrent neural network (RNN) models outperform the feed-forward model; in terms of lower predictive errors and higher percentage of correct reversal detection. However, both models perform better with short term forecasting. The effect of varying the damped feedback weights in the recurrent net is also investigated and it was found that RNN at the optimal weighting factor gives satisfactory performances compared to the ARIMA model. [2]

EL Houssainy A. Rady et al. used the tree-based methods for time series data forecasting and compare between Decision Tree (DT), Random Forest (RF), Gradient Boosted Trees (GBT) and ARIMA model to predict monthly gold prices. The results indicated that RF was better than DT, GBT and ARIMA(0,1,1) in predicting future gold prices, based on RMSE= 38.52.[3]

The authors, Ge Chenghan and Wang Tao (2020) conducted research on the "Improvement of Bayesian Dynamic Linear Model for Predicting Missing Data of Bridges." In this paper, they employed the Bayesian Dynamic Linear Model to predict missing data in bridge-related studies. By comparing the predicted result with the observed value, it is found that the absolute error is less than 14.05Hz and the relative error is less than 1.82% when the training frequency value varies from 756 Hz to 773.4 Hz.[4]

Empirical evaluation of Gated Recurrent Neural Networks on Sequence model by Junyoung Chung, Caglar Gulcehre, KyungHyun Cho and Yoshua Bengio [5] used the Recurrent Neural Network machine learning model along with their variants LSTM and GRU to evaluate. This repeater unit on polyphonic music modeling and speech signal modeling tasks. This test was performed on the music dataset [Boulanger Lewandowski et al., 2012].

Comparison of LSTM and GRU models to predict the condition of pulp presses performed by Dr. Balduino Cesar Mateus, Prof. mateus mendes, Prof. José Torres Farinha, Rui Assis, Prof. Dr. Antonio J. Marques Cardoso. The study presented in this paper compares the performance of LSTM and GRU models, predicting future values of six sensors, installed at an industrial press 30 days in advance. In general, GRU models work with less data and yield better results, with a wider range of parameters, as exemplified in the press-based case study pulp [6].

## III. MATERIALS

### A. Data soucre

This dataset contains information on three major precious metals: Gold, Silver, and Platinum, collected from the Investing.com website. The data was collected from May 19, 2018, to May 19, 2023. The dataset provides detailed trading information including the highest price, lowest price, opening price, closing price, and trading volume.

The columns in the dataset are defined as follows:

- Date: Time interval representing the data.
- Close: Closing price of the metal at each trading period
- Open: Opening price of the metal at each trading period.
- High: Highest closing price of the metal within a trading day.
- Low: Lowest closing price of the metal within a trading day.
- Volume: Volume of the metal traded within a specific day.

With this dataset, you can explore and analyze price trends and fluctuations of these precious metals from 2018 to 2023.

- Gold price: The dataset of gold prices includes the 6 attributes mentioned above and 1296 rows.
- Silver price: The dataset of silver prices includes the 6 attributes mentioned above and 1280 rows.
- Platinum price: The dataset of platinum prices includes the 6 attributes mentioned above and 1558 rows.

### B. Descriptive Statistics

| | *Gold* | *Silver* | *Platinum* |
|---|---|---|---|
| *Count* | 1296 | 1280 | 1558 |
| *Mean* | 1657.15 | 20.41 | 944.06 |
| *STD* | 247.03 | 4.32 | 116.99 |
| *Min* | 1184.00 | 11.77 | 595.20 |
| *Max* | 2069.40 | 29.42 | 1318.75 |
| *25%* | 1467.67 | 16.22 | 852.99 |
| *50%* | 1751.75 | 20.35 | 930.87 |
| *75%* | 1841.32 | 24.25 | 1018.2 |

*Table 1 Descriptive Statistics dataset*

### *1. Gold price*

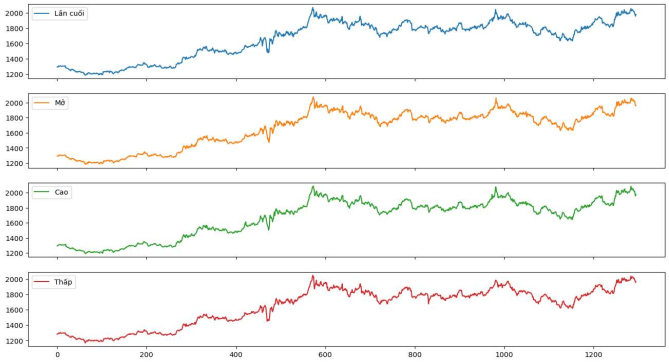

*Figure 1 Visualization of Gold price*

### *2. Silver price*

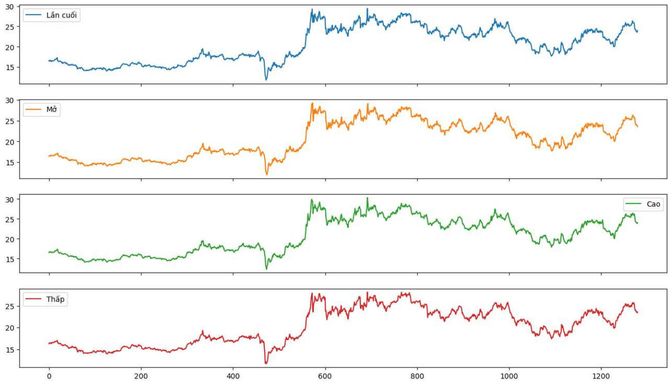

*Figure 2 Visualization of Silver price*

### 3. Platinum price

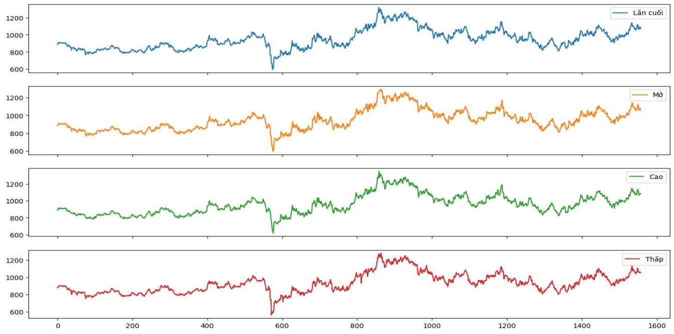

*Figure 3: Visualization of Platinum price*

## IV. METHODS

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

*1. Linear Regression (LR).*

Linear Regression [7] or linear regression is a data analysis technique used to predict the value of one variable based on the value of another variable. It allows us to determine which factors are most important, which factors can be ignored, and how these factors interact with each other. The Linear Regression algorithm describes the relationship between a dependent variable and one or more independent variables. This technique is used to find the best prediction equation for y as a linear function of the independent variables, represented by the following formula:

$$y = \beta_0 + \beta_1 X + \varepsilon \ (1)$$

Where:

y is the predicted value of the dependent variable for any given value of the independent variable.

$\beta_0$ is the intercept coefficient, the predicted value of y when x is 0.

$\beta_1$ is the regression coefficient, the amount by which y changes when x increases.

X is the independent variable (the variable that influences the predicted value y).

*2. ARIMA*

The ARIMA [8] (AutoRegressive Integrated Moving Average) algorithm is a self-regressive algorithm used in time series data forecasting.

The algorithm combines three components: Auto-Regressive (AR), Integrated (I), and Moving Average (MA), corresponding to the parameters p, d, and q, representing the three main components of the model, where:

p (AR Order): The parameter p represents the number of autoregressive processes in the autoregressive (AR) component of the ARIMA model. It indicates the number of past days of the data series used to predict the current value. Each past value is used as a coefficient in the autoregressive model. The value of p depends on the temporal dependence in the data series and can be determined using methods such as autocorrelation function (ACF) plot or partial autocorrelation function (PACF) plot. The general equation for AR autoregression is as follows:

$$y_t = \mathrm{c} + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t \ (2)$$

Where:

$y_t$: Represents the data value at time t.

c: Intercept coefficient.

$\varphi_t$: Autoregressive parameter.

$\varepsilon_t$: Random error term, usually expected to have a mean of 0.

.

d (Differencing Order): The parameter d represents the number of times differencing is applied to the original data series to remove trends and/or seasonality and transform the data into a stationary series. A stationary series is a time series with a constant mean and variance over time, and the covariance between two periods depends only on the time lag between them, not on the actual time points at which the covariance is calculated. The differencing formula at time t is as follows:

$$\nabla y_t = y_t - y_{t-1} \ (3)$$

Where:

$y_t$: Data value at time t.

q (MA Order): The parameter q represents the number of moving average components in the moving average (MA) part of the ARIMA model. It indicates the number of past moving average values used to predict the current value. The value of q depends on the random dependence in the data series and can be determined using methods such as autocorrelation function (ACF) plot or partial autocorrelation function (PACF) plot. The equation for the MA moving average part is as follows:

$$\mathrm{y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}} \ (4)$$

Where:

C: Intercept coefficient.

$\varepsilon_t$: White noise value at time t.

$\theta_1$: Correlation coefficient.

*3. Recurrent Neural Network (RNN).*

Recurrent neural networks (RNNs)[9] are a rich class of dynamic models that have been used to generate sequences in domains as diverse as music, text and motion capture data. RNNs can be trained for sequence generation by processing real data sequences one step at a time and predicting what comes next. Assuming the predictions are probabilistic, novel sequences can be generated from a trained network by iteratively sampling from the network's output distribution, then feeding in the sample as input at the next step. In other words by making the network treat its inventions as if they were real, much like a person dreaming. Although the network itself is deterministic, the stochasticity injected by picking samples induces a distribution over sequences. The function of RNN can be expressed as:

$$h_t = \sigma(W h_{t-1} + U x_t) \ (5)$$

$$o_t = \sigma(V h_t) \ (6)$$

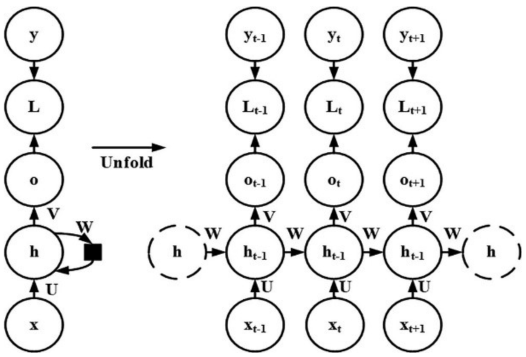

*Figure 4 A Standard RNN structure diagram.*

Although RNN was designed to predict time series theoretically, it is difficult to predict long time series due to the length of information varies in practical applications, which can cause gradients to disappear or explode.

### *4. Gated Recurrent Unit (GRU).*

GRU, which was first proposed by Cho et al. in 2014 to adaptively capture dependencies of different time scales, has only two gating units: the update gate and the reset gate. The update gate controls the degree to which the status information of the previous moment is brought into the current state. The reset gate controls the degree of ignoring the status information of the previous moment. [11]

The structure of a GRU[12] neuron is shown in Fig. 1. The values of the update gate and the reset gate are determined by the previous hidden state ht-1 and current input xt. The process of data transmission in a GRU neuron can be described as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \ (7)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \ (8)$$

$$\tilde{h}_t = tanh(W_h \cdot [r \odot h_{t-1}, x_t]) \ (9)$$

$$h_t = (1 - z_t) \odot h_{t-1} + \tilde{h}_t \odot z_t (10)$$

While $r_t$ is the output of the reset gate at time t, $z_t$ is the output of the update gate at time t, $h_t$ and $h_{t-1}$ are the output at time *t* and *t* − *1*, respectively, $x_t$ is the input at time $_t$, and σ is the activation function. The calculation process of the memory unit is expressed by Eqs.

Two activations σ (sigmoid) and tanh are used to simulate the gate and normalize the input, respectively. $r_t$ and $z_t$ are the outputs of the reset gate and the update gate, respectively. $h_t$ is the Candidate hidden state and $h_t$ is the hidden state $W_r$ , $W_z$ , and $W_h$ are weight matrices that will be updated by optimizations.

### *5. Vector Autoregression (VAR).*

Vector auto-regression (VAR) time series model has wide application in econometric forecasting model; VAR can capture the evolution and the inter-dependencies between multiple time-series. All the variables in a VAR are treated symmetrically by including for each variable an equation explaining its evolution based on its own lags and the lags of all the other variables in the model. [13]

In other words, VAR is a multivariate prediction algorithm used when two or more time series influence each other. That is to say, the basic requirements for using VAR are:

1. You need at least two time series (variables)
2. The time series should influence each other.

Also, It is considered as an Autoregressive model because, each variable (Time Series) is modeled as a function of the past values, that is the predictors are nothing but the lags (time delayed value) of the series.[14]

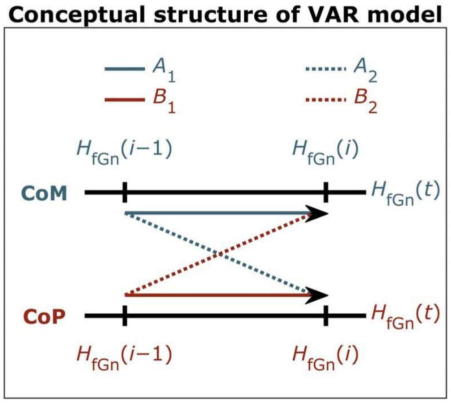

*Figure 5 A Standard VAR structure diagram.*

A VAR(p) model considers p dependent variables, denoted VAR(p). This model is defined by a system of p equations:

$$Y_t = \alpha + B_1 Y_{t-1} + B_2 Y_{t-2} + B_3 Y_{t-3} + \cdots + B_p Y_{t-p} + \varepsilon_t \ (11)$$

Where:

- $Y_t$ is a vector of the dependent variable at time t.
- $\alpha$ is a constant vector.
- $B_1, B_2, \ldots, B_p$ are coefficient matrices, each of which size is (k x k), where k is the number of dependent variables.
- $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$ are vectors of dependent variables at previous times.
- $\varepsilon_t$ is the error, which is considered as white noise.

The goal of VAR is to estimate the coefficient matrices B1, B2, …, Bp to model the relationship between dependent variables. These coefficient matrices can be estimated by methods such as least squares regression (OLS)[15] or the method of maximum likelihood (MLE).[16]

### *6. Random Forest (RF)*

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [...] to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. [17]

Random forests are used for classification and regression tasks. For classification tasks, the output of the random forest is the class chosen by most trees. For regression tasks, the mean or median prediction of the individual trees is returned.

Random forest arrives at a decision or prediction based on the maximum number of votes received from the decision trees. The outcome which is arrived at, for a maximum number of times through the numerous decision trees is considered as the final outcome by the random forest.[18]

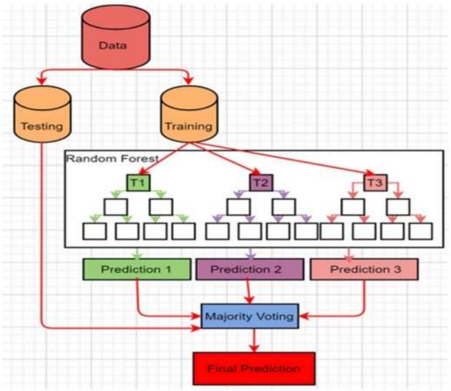

*Figure 6 The structure and the prediction  of the RF model*

The steps involved in the Random Forest algorithm are:

1. Randomly select n samples from the dataset.
2. Build a decision tree based on these n samples.
3. Choose the number of decision trees you want to build and repeat steps 1 and 2 for each tree.
4. For each new data point, predict the class based on the majority vote of all the decision trees.

The training algorithm for random forests applies the general technique of boostrap aggregating [19], or bagging, to tree learners. Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly ($B$ times) selects a random sample with replacement of the training set and fits trees to these samples:

$$For\ b = 1, ..., B \quad (12)$$

1. Sample, with replacement, n training examples from X, Y; call these $X_b$, $Y_b$.
2. Train a classification or regression tree $f_b$ on $X_b$, $Y_b$.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on $x'$:

$$f(x') = \frac{1}{B}\, f_b(x') \quad (13)\ [20]$$

7. *Hybird Neural Network (HNN).*

Based on the above analysis, this paper proposes an HNN [21] model based on CNN and LSTM. The architecture of the hierarchical feature extraction model is shown in Figure 4, and the algorithm is described below in detail.

A Convolutional Neural Network is a special type of neural network for processing data with a grid-matrix structure. CNN consists of two main parts: Convolutional layer (convolutional layer) and pooling layer (pooling layer). The input data in the form of time series will be treated as a one-dimensional matrix, processed by convolutional layers and activation functions. The output data is then extracted by the aggregate layers to extract the features of the data. Combining a large number of classes gives CNN an advantage in extracting features of the past, continuing to include layers of the LSTM model to train the model. In general, a CNN-LSTM structure is generalized as follows:

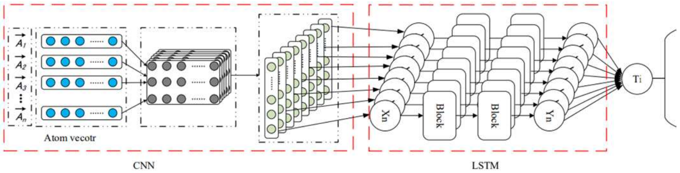

*Figure 7 The architecture of the HNN model.*

The model starts with 3 Conv1D layers combined with 3 alternating MaxPooling1D layers. Conv1D layers are capable of finding important features in time series data. The model starts with 3 Conv1D layers combined with 3 alternating MaxPooling1D layers. Conv1D layers are capable of finding important features in time series data. Then, the MaxPooling1D layers are used to reduce the output size from the Conv1D layers and create a matrix of data features. Next, the Flatten layer is used to transform the output from the final Conv1D layer into a 1-dimensional tensor. This process helps to prepare the data for transmission into the LSTM layers.

Finally, the LSTM layers used to understand and model long-term patterns and dependencies in the time series data were drawn by the CNN. The model uses two layers of LSTM stacked on top of each other and surrounded by a Bidirectional layer, allowing the transmission of information over time and enhancing the model's learning ability. The Bidirectional class has a function that allows the model to look at string data from both directions: left-to-right and right-to-left. This allows the model to aggregate information from both the past and future of each data point, which improves predictability. Dropout layer is applied after each LSTM layer to prevent overfitting during training by randomly removing part of the neurons in the network.

8. *Neural Network Autoregression (NNAR).*

In an auto-regressive model, we forecast the variable of interest using a linear combination of past values of the variable. The term "auto" in Auto-Regression (AR)[22] indicates that the variable is regressed against itself . This is like a multiple regression but with lagged values of the time-series yt as predictors. We refer to this as an AR(p) model, an auto-regressive model of order p. An AR model of order p can be written as:

$$y_t = c + \sum_{i=1}^{i=p} \omega_i * y_{t-1} + e_t \ (14)$$

Where $y_{t-1,\dots,t-p}$ are the p lag terms used to predict $y_t$ and $e_t$ is white noise. The $p$ weights $\omega_i$, by which each of the p lags $y_{t-i}$ is multiplied, are also referred to as the AR-coefficients. As baseline, we use a traditional implementation of the auto-regressive model, fitted with least squares. We will refer to this model as *Classic-AR.*[23]

We propose *AR-Net* which mimics the traditional AR process with a neural network. It is designed such that that the parameters of its first layer are equivalent to the AR-coefficients (see figure 1a). *AR-Net* can optionally be extended with hidden layers to achieve greater forecasting accuracy, at the cost of direct interpretability (see figure 1b). In this paper, we will only evaluate the *AR-Net* model structure without hidden layers.

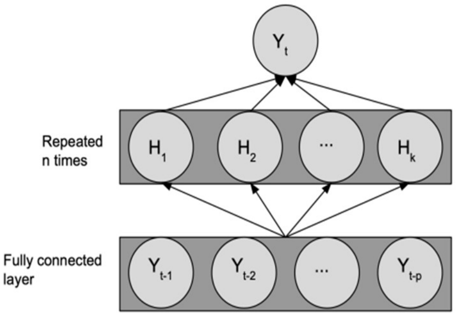

*Figure 8 An AR equivalent neural network architecture.*

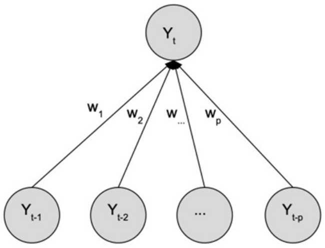

*Figure 9:An AR inspired neural network with n hidden layers of size k*

*9. Multiscale Recurrent Neural Networks (MRNN)*

Multivariate time series often involve a mixture of long-term and short-term patterns[24]. The different complexity of the patterns could lead to the failure of a single model. This is because complex models that can capture the local nonlinear patterns may face difficulty in simultaneously capturing long-term smooth patterns. Therefore, the generalizability of these models is limited when they are used to predict patterns with multiple scales.

The structure of the proposed multiscale model is shown in Figure 12, in which the RNN model is used to learn short-term nonlinear features, while an overly simple FCNN model is responsible for detecting and predicting long-term smooth features. The RNN component has access to recent historical data as well as the control variables for the prediction steps. On the other hand, the FCNN model only has access to the cumulative controls (coarse-scale information) to prevent it from learning short-term trends. The final prediction is obtained by adding the outputs from the short-term (RNN) and long-term (FCNN) model. To force the two models to detect their intended features, two loss functions are defined for training, short-term loss, and a long-term loss. The long-term loss is the difference between the actual observation and the long-term prediction from the FCNN. Although the short-term loss minimizes the difference between the final prediction and the actual observations, its gradient is only allowed to backpropagate through the short-term model.

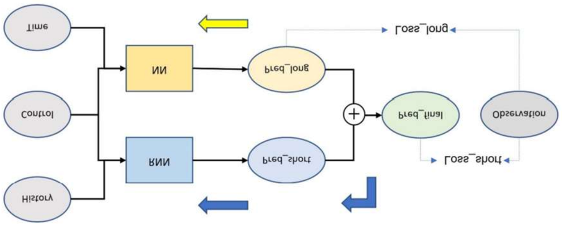

*Figure 10 The structure and the training process of the multiscale model*

*a) Positioning Figures and Tables:* Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence.

## V. RESULT

### A. Measure

With a set of forecast results consisting of n forecast days, where $Y_A$ is the actual value of day i, $Y_B$ is the actual value of day *i-1*, f is the forecast value of day i, we can consider the effectiveness of model on three measures as follows:

**Root Mean Square Error – RMSE:** RMSE is the average size of the absolute error between the actual value and the predicted value.

$$RMSE = \sqrt{\frac{\sum(Y_A - f)^2}{n}} \quad (15)$$

**Mean Absolute Percentage Error – MAPE:** MAPE is the average percentage between the absolute error and the actual value.

$$MAPE = \frac{1}{n}\sum\left|\frac{(Y_A - f)}{Y_A}\right| (16)$$

**Mean Absolute Error (MAE):** Average of the absolute error between the actual value $y_i$ and the predicted value $\tilde{y}_i$.

$$AE = \frac{1}{n}\sum|Y_A - f| \; (17)$$

### B. Predict the next 30 days.

After comparing the accuracy of the models and having the results table as Table of testdata. The 3 models with the best accuracy according to RMSE measure used to forecast gold price in the next 30 days of 3 data sets Gold, Silver, Platinum are RNN at the rate of 7-2-1, MRNN at the rate of 6-3-1 and GRU at 5-3-2. The forecast results are shown in the figures below. From the figure below, RNN and GRU both give similar prediction results when there is a downward trend in the Gold and Silver data sets. As for the Platinum set, RNN tends to decrease sharply, and GRU is opposite by increasing slightly. As for MRNN, it tends to stay stable in the next 30 days. Besides, there are also a number of models with high volatility values that are as accurate as the 3 algorithms mentioned above, which are HNN and NNAR, which will be predicted with a symbolic chart to easily observe the volatility.

*Result:*

| Model | | *Gold* | | | *Silver* | | | *Platinum* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **5-3-2** | **6-3-1** | **7-2-1** | **5-3-2** | **6-3-1** | **7-2-1** | **5-3-2** | **6-3-1** | **7-2-1** |
| **LR** | **RMSE** | 322.46 | 379.92 | 313.81 | 2.82 | 7.03 | 7.48 | 178.47 | 178.03 | 196.54 |
| | **MAPE** | 16.25% | 19.40% | 15.99% | 9.82% | 24.13% | 32.95% | 13.04% | 16.15% | 19.35% |
| | **MAE** | 295.84 | 344.52 | 281.54 | 2.47 | 5.14 | 6.78 | 146.44 | 151.56 | 177.88 |
| **ARIMA** | **RMSE** | 73.44 | 110.68 | 89.63 | 1.76 | 5.73 | 3.9 | 200.6 | 252.9 | 154.03 |
| | **MAPE** | 3.46% | 5.36% | 3.97% | 5.9% | 24.13% | 16.39% | 15.85% | 25.00% | 15.17% |
| | **MAE** | 62.57 | 94.46 | 71.49 | 1.47 | 5.14 | 3.32 | 175.7 | 238.03 | 139.28 |
| **GRU** | **RMSE** | 18.58 | 24.66 | 19.5 | 0.5 | 0.54 | 0.48 | 20.82 | 21.5 | 21.68 |
| | **MAPE** | 0.77% | 1.12% | 0.84% | 1.49% | 1.94% | 1.76% | 1.57% | 1.79% | 1.88% |
| | **MAE** | 13.99 | 19.96 | 15.15 | 0.37 | 0.42 | 0.37 | 16.62 | 17.27 | 17.75 |
| **RNN** | **RMSE** | 17.66 | 19.02 | 18.65 | 0.49 | 0.44 | 0.45 | 20.74 | 18.09 | 18.09 |
| | **MAPE** | 0.71% | 0.81% | 0.79% | 1.43% | 1.55% | 1.57% | 1.51% | 1.44% | 1.46% |
| | **MAE** | 12.98 | 14.64 | 14.22 | 0.35 | 0.34 | 0.34 | 16.02 | 13.97 | 13.95 |
| **VAR** | **RMSE** | 199.44 | 127.06 | 117.74 | 2.37 | 5.86 | 3.63 | 126.38 | 139.54 | 100.45 |
| | **MAPE** | 9.52% | 5.87% | 5.56% | 9.43% | 25.64% | 15.04% | 10.55% | 12.53% | 9.03% |
| | **MAE** | 170.06 | 104.32 | 99.49 | 1.99 | 5.44 | 3.03 | 107.07 | 118.2 | 84.97 |
| **RF** | **RMSE** | 39.46 | 34.25 | 31.8 | 3.24 | 3.24 | 3.24 | 40.47 | 34.87 | 31.98 |
| | **MAPE** | 4.24% | 3.51% | 3.28% | 7.75% | 7.29% | 7.09% | 7.97% | 6.73% | 5.95% |
| | **MAE** | 31.12 | 25.9 | 25.55 | 3.16 | 3.15 | 3.15 | 31.4 | 28.21 | 25.62 |
| **HNN** | **RMSE** | 47.93 | 38.43 | 43.37 | 1.22 | 1.19 | 1.75 | 91.04 | 43.04 | 38.05 |
| | **MAPE** | 2.03% | 1.7% | 1.91% | 4.04% | 4.48% | 6.93% | 6.47% | 3.66% | 3.11% |
| | **MAE** | 36.68 | 30.49 | 34.92 | 1.00 | 0.96 | 1.45 | 72.87 | 35.63 | 29.6 |
| **NNAR** | **RMSE** | 58.27 | 58.74 | 57.9 | 1.25 | 0.87 | 0.99 | 51.3 | 32.14 | 42.01 |
| | **MAPE** | 2.6% | 2.61% | 2.65% | 4.08% | 3.25% | 3.81% | 3.78% | 2.66% | 3.67% |
| | **MAE** | 47.42 | 46.92 | 47.7 | 1.02 | 0.71 | 0.81 | 41.63 | 25.56 | 34.63 |
| **MRNN** | **RMSE** | 25.7 | 21.25 | 25.02 | 0.54 | 0.77 | 0.69 | 25.44 | 18.02 | 22.58 |
| | **MAPE** | 1.06% | 0.9% | 1.07% | 1.64% | 2.81% | 2.54% | 1.84% | 1.43% | 1.86% |
| | **MAE** | 19.23 | 16.12 | 19.6 | 0.41 | 0.62 | 0.54 | 20.01 | 13.77 | 17.78 |

*Table 2 Evaluation results of the set of TEST*

| Model | | *Gold* | | | *Silver* | | | *Platinum* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **5-3-2** | **6-3-1** | **7-2-1** | **5-3-2** | **6-3-1** | **7-2-1** | **5-3-2** | **6-3-1** | **7-2-1** |
| **LR** | **RMSE** | 697.74 | 523.14 | 358.74 | 6.53 | 9.47 | 8.26 | 72.47 | 200.46 | 183.93 |
| | **MAPE** | 38.29% | 27.55% | 18.77% | 30.17% | 40.55% | 35.32% | 6.15% | 19.28% | 17.59% |
| | **MAE** | 692.80 | 520.11 | 353.67 | 6.31 | 9.36 | 8.14 | 59.96 | 193.94 | 276.7 |
| **ARIMA** | **RMSE** | 115.79 | 87.93 | 128.22 | 3.59 | 4.78 | 2.17 | 109.54 | 211.24 | 87.65 |
| | **MAPE** | 5.44% | 4.04% | 5.27% | 14.99% | 19.29% | 7.92% | 8.85% | 20.35% | 7.47% |
| | **MAE** | 97.53 | 77.23 | 103.27 | 3.02 | 4.54 | 1.77 | 89.53 | 204.67 | 73.70 |
| **GRU** | **RMSE** | 19.47 | 23.38 | 20.4 | 0.49 | 0.51 | 0.44 | 18.91 | 20.5 | 20.39 |
| | **MAPE** | 0.84% | 0.97% | 0.8% | 1.75% | 1.68% | 1.39% | 1.57% | 1.59% | 1.59% |
| | **MAE** | 15.29 | 18.65 | 15.49 | 0.38 | 0.38 | 0.32 | 15.0 | 16.13 | 16.1 |
| **RNN** | **RMSE** | 18.45 | 21.14 | 20.82 | 0.45 | 0.42 | 0.42 | 19.24 | 17.07 | 17.26 |
| | **MAPE** | 0.76% | 0.81% | 0.81% | 1.54% | 1.37% | 1.33% | 1.61% | 1.28% | 1.31% |
| | **MAE** | 13.82 | 15.77 | 15.7 | 0.33 | 0.32 | 0.31 | 15.54 | 13.03 | 13.33 |
| **VAR** | **RMSE** | 178.51 | 102.19 | 67.37 | 2.36 | 3.43 | 4.72 | 245.13 | 112.08 | 66.97 |
| | **MAPE** | 9.01% | 5.18% | 3.24% | 7.57% | 12.56% | 6.45% | 19.81% | 9.77% | 5.48% |
| | **MAE** | 160.74 | 92.54 | 60.57 | 1.98 | 3.01 | 4.48 | 224.41 | 99.68 | 53.67 |
| **RF** | **RMSE** | 40.97 | 36.1 | 40.48 | 4.25 | 4.24 | 4.19 | 36.19 | 28.40 | 28.83 |
| | **MAPE** | 4.32% | 3.8% | 4.2% | 7.86% | 7.32% | 6.05% | 7.01% | 6.5% | 6.11% |
| | **MAE** | 28.23 | 24.54 | 28.22 | 4.16 | 4.15 | 4.13 | 24.75 | 24.08 | 20.79 |
| **HNN** | **RMSE** | 48.98 | 56.18 | 70.8 | 1.51 | 1.0 | 1.51 | 48.61 | 43.18 | 34.7934.79 |
| | **MAPE** | 2.25% | 2.27% | 3.0% | 5.86% | 3.78% | 5.96% | 3.94% | 3.51% | 2.69% |
| | **MAE** | 41.14 | 44.58 | 59.18 | 1.22 | 0.86 | 1.37 | 38.83 | 35.8 | 27.68 |
| **NNAR** | **RMSE** | 66.39 | 79.11 | 66.84 | 1.44 | 0.94 | 1.1 | 40.0 | 29.43 | 38.39 |
| | **MAPE** | 3.04% | 3.34% | 2..75% | 5.31% | 3.47% | 3.99% | 3.32% | 2.33% | 3.05% |
| | **MAE** | 55.7 | 84.72 | 52.92 | 11.19 | 0.8 | 0.93 | 32.69 | 23.82 | 31.36 |
| **MRNN** | **RMSE** | 26.54 | 27.55 | 35.34 | 0.54 | 0.75 | 0.64 | 20.36 | 17.97 | 20.74 |
| | **MAPE** | 1.17% | 1.07% | 1.47% | 1.88% | 2.69% | 2.14% | 1.69% | 1.36% | 1.62% |
| | **MAE** | 21.16 | 20.85 | 28.8 | 0.4 | 0.61 | 0.49 | 0.49 | 16.39 | 16.5 |

*Table 3 Evaluation results of the set of TRAIN*

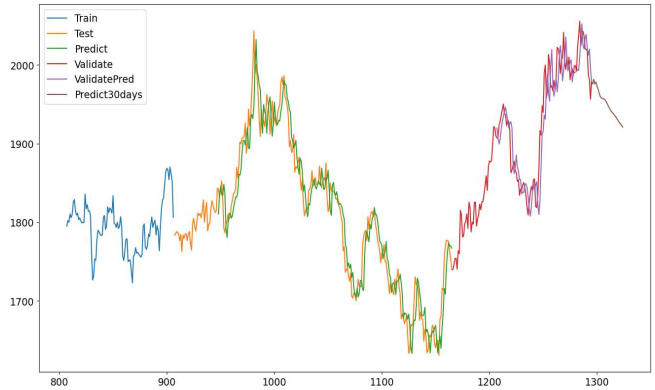

*Figure 11 Predictive Gold price with  the RNN model with rate of 7-2-1*

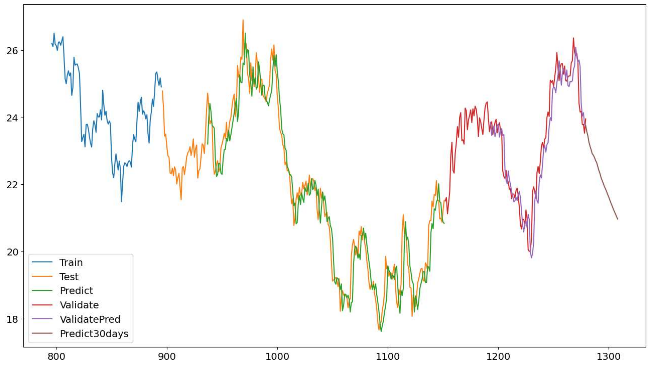

*Figure 12 Predictive Silver price with the RNN model with rate of 7-2-1*

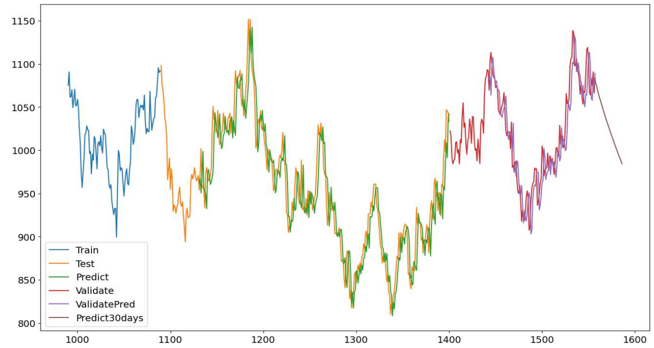

*Figure 13 Predictive Platinum price with the RNN model with rate of 7-2-1*

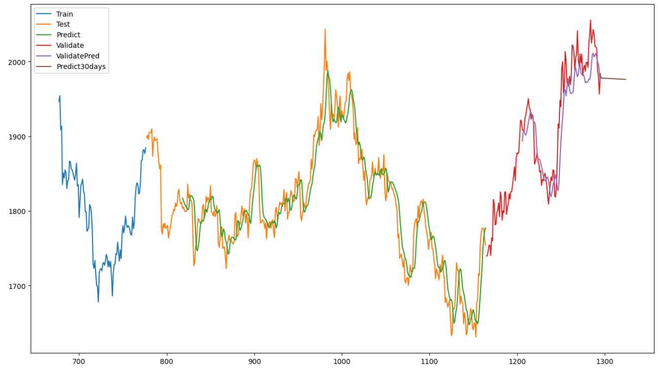

*Figure 14 Predictive Gold  price with the MRNN model with rate of 6-3-1*

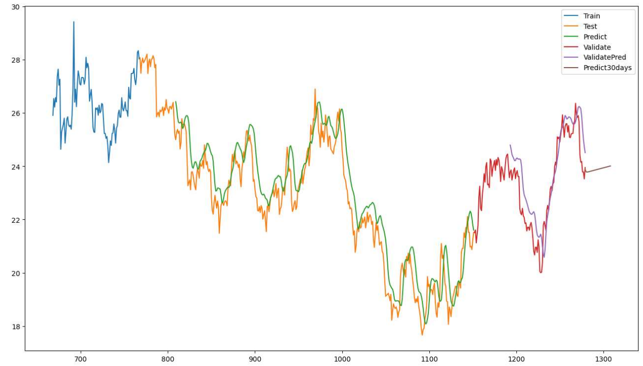

*Figure 15 Predictive Silver price with the MRNN model with rate of 6-3-1*

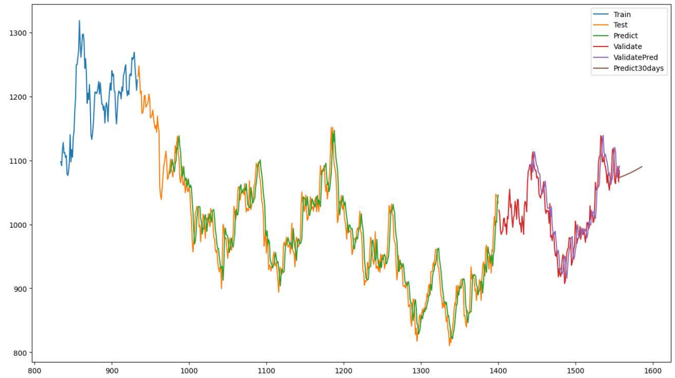

*Figure 16 Predictive Platinum price with the MRNN model with rate of 6-3-1*

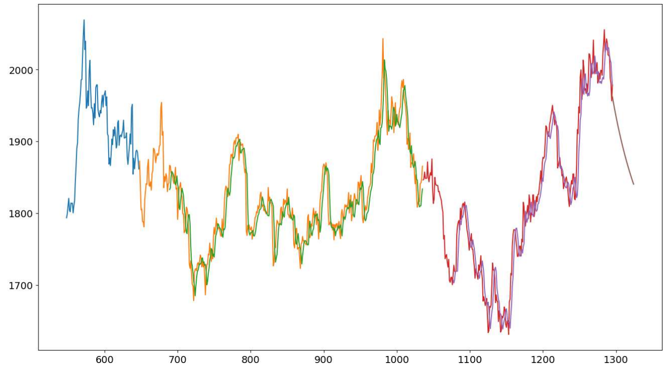

*Figure 17 Predictive Gold price with the GRU model with rate of 5-3-2*

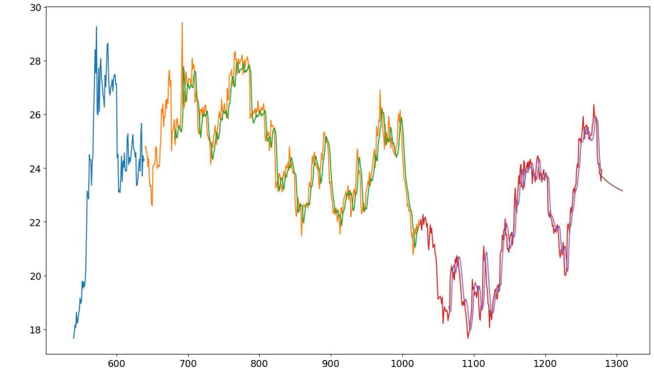

*Figure 18 Predictive Silver price with the GRU model with rate of 5-3-2*

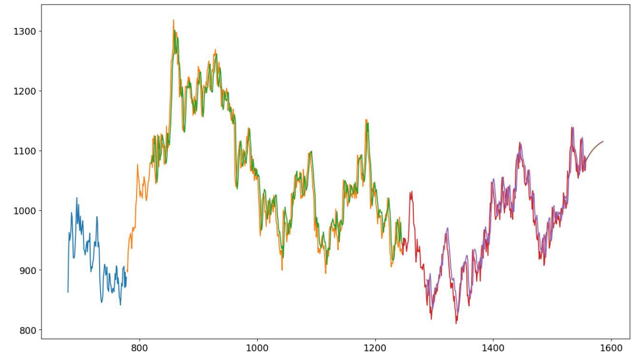

*Figure 19 Predictive Platinum price with the GRU model with rate of 5-3-2*

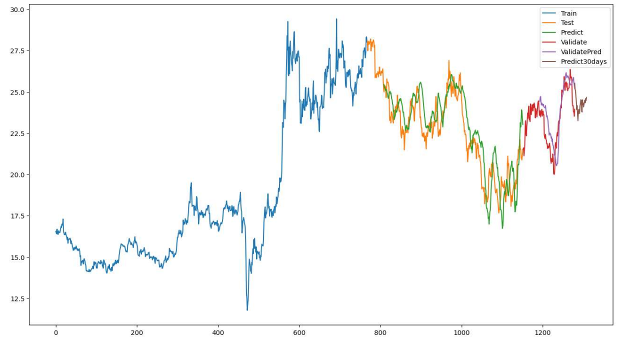

*Figure 20 Predictive Silver price with the HNN model with rate of 6-3-1*

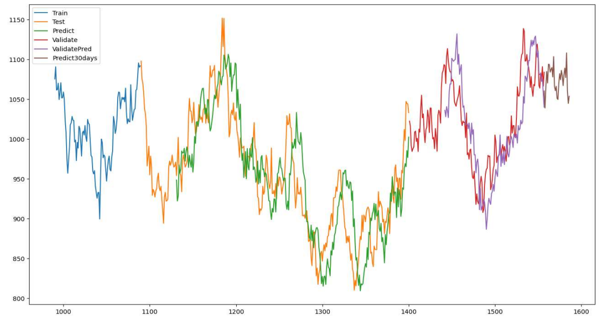

*Figure 21 Predictive Gold price with  the NNAR model with rate of 7-2-1*

## VI. CONCLUSION

From the comparison results of Table of test data based on the RMSE and MAPE measures, it can be seen that the RNN model has the best results in the models at all 3 scales and all three data sets. Besides, although the MRNN at the ratio of 6-3-1 is better than the RNN in the Platinum dataset, overall, the RNN still gives the best results in most cases. Along with that, it can also be seen that GRU, HNN, and NNAR models also give very good results.

Thereby, it can be seen that the RNN and GRU models when reasonably built give very good results and are suitable for the precious metal price forecasting problem. Choosing the best model will depend on the purpose, when considering the MAE measure, the team will focus on finding the models that have the best ability to predict the next day's binary trend (up/down) without regard to focus on the actual increase or decrease. In contrast, considering the RMSE or MAPE measure, it means that we will be interested in the increase or decrease, the difference between the predicted and actual value of the stock that most related studies consider empirical results. on this aspect.

## VII. FUTURE DEVENLOPMENT DIECTION

During the experiment, despite some difficulties in finding suitable layered architectures for deep learning models as well as evaluating the experimental results of a large number of models spanning from statistical , machine learning to deep learning, but in general, the experimental results have exceeded the initial expectations of the study. With the stock data set with many unexpected fluctuations, especially under the influence of the Covid epidemic, accurately forecasting future values can face many disadvantages, however, if choosing hyperparameters, building model architecture as well as choosing a reasonable model where RNN and GRU have given very good results, the team hopes to be able to improve and promote the current advantages in terms of diversity of models. Figures are used to obtain the most objective and comprehensive results.

The addition of the MAE measure also gives the experiment another way to evaluate the model's effectiveness through the data trend matching rate, which is quite new compared to other previous research papers. In the future, the team will focus on improving the performance of the models by refining the hyperparameter sets as well as the layer architecture of deep learning models such as HNN, RNN, MRNN, NNAR, and GRU to take advantage of Get the power of deep learning models.

The group will also experiment more with new models and more data sets, especially with banks that experience a lot of exchange rate fluctuations due to unexpected factors (New Year's Day, politics, inflation) to increase practicality for the training model.

## VIII. ACKNOWLEDGMENT

First and foremost, I would like to express my deepest gratitude to **Assoc. Prof. Dr. Nguyen Dinh Thuan, Mr. Nguyen Minh Nhut, Ms. Nguyen Thi Viet Huong** 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. Nhut** and **Ms. Huong**, who teaches and supports our team throughout our course.

I would like to take this opportunity to thank **Assoc. Prof. Dr. Nguyen Dinh Thuan** 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. Nguyen Dinh Thuan, Mr. Nguyen Minh Nhut** and **Ms. Nguyen Thi Viet Huong** 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.

## IX. REFERENCES

[1] D. Satria, “PREDICTING BANKING STOCK PRICES USING RNN, LSTM, AND GRU APPROACH,” Appl. Comput. Sci., vol. 19, pp. 82–94, Mar. 2023, doi:10.35784/acs-2023-06.

[2] S. L. Ho, M. Xie, and T. N. Goh, “A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction,” Computers & Industrial Engineering, vol. 42, no. 2, pp. 371–375, Apr. 2002, doi:10.1016/S0360-8352(02)00036-0.

[3] “Time Series Forecasting Using Tree Based Methods,” J.Stat. Appl. Prob., vol. 10, no. 1, pp. 229–244, Mar. 2021, doi: 10.18576/jsap/100121.

[4] M. Karim, M. Foysal, and S. Das, “Stock Price Prediction Using Bi-LSTM and GRU-Based Hybrid Deep Learning Approach,” Nov. 2022, pp. 701–711. doi: 10.1007/978-981-19-3148-2_60.

[5] J. Chung, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” arXiv.org, Dec. 11, 2014. https://arxiv.org/abs/1412.3555?fbclid=IwAR1Gvx3JKBDVjy

[6] B. C. Mateus, M. Mendes, J. P. S. Farinha, R. Assis, and A. H. Cardoso, “Comparing LSTM and GRU Models to Predict the Condition of a Pulp Paper Press,” Energies, vol. 14, no. 21, p. 6958, Oct. 2021, doi: 10.3390/en14216958.

[7] R. Bevans, “Simple Linear Regression | An Easy Introduction & Examples,” Scribbr, Feb. 19, 2020. https://www.scribbr.com/statistics/simple-linear-regression/ (accessed Jun. 17, 2023).

[8] A. C. Harvey, “ARIMA Models,” in *Time Series and Statistics*, J. Eatwell, M. Milgate, and P. Newman, Eds., in The New Palgrave. London: Palgrave Macmillan UK, 1990, pp. 22–24. doi: 10.1007/978-1-349-20865-4_2.

[9] J. L. Elman, “Finding Structure in Time,” *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, 1990, doi: 10.1207/s15516709cog1402_1.

[10] Alex Graves, “Generating Sequences With Recurrent Neural Networks”, 5 Jun 2014 doi: 10.1016/j.inpa.2020.02.002

[11]Wuyan Li, Hao Wu, Nanyang Zhu, Yongnian Jiang, Jinglu Tan, Ya Guo , “ Prediction of dissolved oxygen in a fishery pond based on gated recurrent unit (GRU)”, Received 17 May 2019, Revised 28 January 2020, Accepted 10 February 2020, Available online 10 February 2020, Version of Record 9 March 2021. doi: 10.1016/j.inpa.2020.02.002

[12] Geng Tang, Xiaoming Xue, Adnan Saeed, Xin Hu, “Short-Term Wind Speed Interval Prediction based on Ensemble GRU Model” DOI 10.1109/TSTE.2019.2926147

[13] https://towardsdatascience.com/vector-autoregressive-for-forecasting-time-series-a60e6f168c70

[14] Vector Autoregression (VAR) - Comprehensive Guide with Examples in Python - Machine Learning Plus

[15] Ordinary Least Squares Method: Concepts & Examples - Data Analytics (vitalflux.com)

[16] Maximum Likelihood Method - an overview | ScienceDirect Topics

[17] (The paper “Random Forests” by Leo Breiman: Random Forests | SpringerLink )

[18] Random Forest Algorithm In Trading Using Python (quantinsti.com)

[19]: RANDOM FOREST ALGORITHM. | Download Scientific Diagram (researchgate.net)

[20] Ensemble Methods in Machine Learning: Bagging Versus Boosting | Pluralsight

[21] Li, S., Dan, Y., Li, X., Hu, T., Dong, R., Cao, Z., & Hu, J. (2020). *Critical Temperature Prediction of Superconductors Based on Atomic Vectors and Deep Learning. Symmetry, 12(2), 262.* doi:10.3390/sym12020262

[22] Oskar J. Triebe, Nikolay Laptev, Ram Rajagopal “AR-Net: A SIMPLE AUTO-REGRESSIVE NEURAL NETWORK FOR TIME-SERIES”, December 2, 2019.

[23] Oskar Triebe, “ AR-Net: A simple Auto-Regressive Neural Network for time-series“, Wed, 27 Nov 2019.

[24] Anyue Jiang, Zhen Qin, Trenton T. Cladouhos, Dave Faulder, and Behnam Jafarpour Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, CA, USA “A Multiscale Recurrent Neural Network Model for Long-Term Prediction of Geothermal Energy Production” February 7-9, 2022.