VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

MID-TERM REPORT

of

TIME SERIES ANALYSIS COURSE

Lecturer: Trinh Tuan Phong

Group 5:

An Minh Tri - BI12-447

Nguyen Cong Quoc - BI12-375

Nguyen Son – BI12-389

Vu Hung Son - BI12-390

Nguyen Duc Phuong - BI12-357

Hanoi, June 6, 2024

I. Introduction

In this mid-term project for the course, we will try to build predictive models using ARIMA, LSTM, and GRU to predict the price of the two stocks: HPG (Hoa Phat Group) and MWG (Mobile World Investment Corporation) between January 2, 2018, to March 31, 2023.

These companies are from different sectors (manufacturing and retail electronics) and are listed on the Ho Chi Minh Stock Exchange (HOSE).

After comparing the performance of these models, we will know which model is better for forecasting stock prices for these companies.

II. Descriptive Statistics and Data Visualization

1. Descriptive Statistics

For the analysis of the stock prices of Hoa Phat Group (HPG) and Mobile World Investment Corporation (MWG), we conducted basic descriptive statistics to summarize the central tendency, dispersion, and shape of the distribution of the dataset's closing prices.

HPG Stock Prices (January 2, 2018 - March 31, 2023):

- **Count:** 1557
- **Mean:** 19,974.49
- **Standard Deviation:** 9,174.28
- **Minimum:** 7,411.80
- **25th Percentile:** 12,317.80
- **Median (50th Percentile):** 17,727.00
- **75th Percentile:** 25,136.00
- **Maximum:** 43,895.80
- **Range:** 36,484.00

MWG Stock Prices (January 2, 2018 - March 31, 2023):

- **Count:** 1557
- **Mean:** 42,164.64
- **Standard Deviation:** 14,309.89
- **Minimum:** 19,198.00
- **25th Percentile:** 28,747.00

- **Median (50th Percentile):** 39,600.00
- **75th Percentile:** 49,624.00
- **Maximum:** 79,582.00
- **Range:** 60,384.00

2. Data Visualization

The following plots illustrate the time series data of the closing prices and the daily percentage changes in prices for both HPG and MWG stocks.

Figure 1: Closing Prices of HPG and MWG

The time series plot shows the closing prices of HPG and MWG from January 2018 to March 2023. Both stocks experienced significant fluctuations during this period, with MWG generally having higher prices compared to HPG. Notable peaks and troughs can be observed, particularly around early 2022.

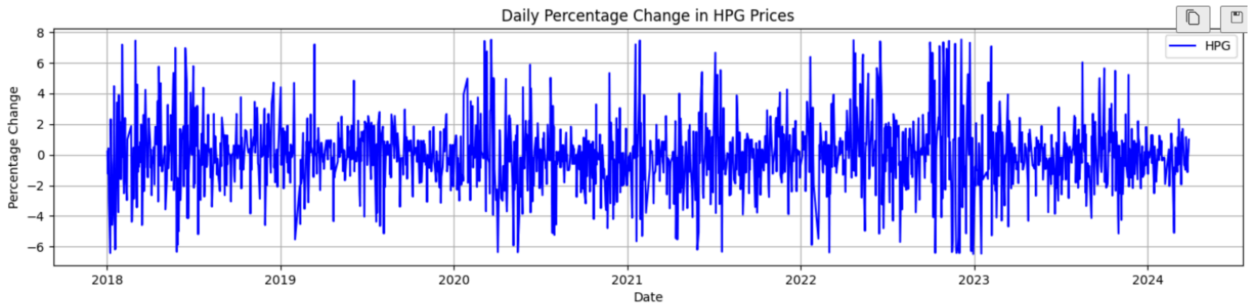

Figure 2: Daily Percentage Change in HPG Prices

This plot depicts the daily percentage change in the closing prices of HPG. The percentage changes indicate high volatility, especially noticeable in early 2018 and throughout 2020.

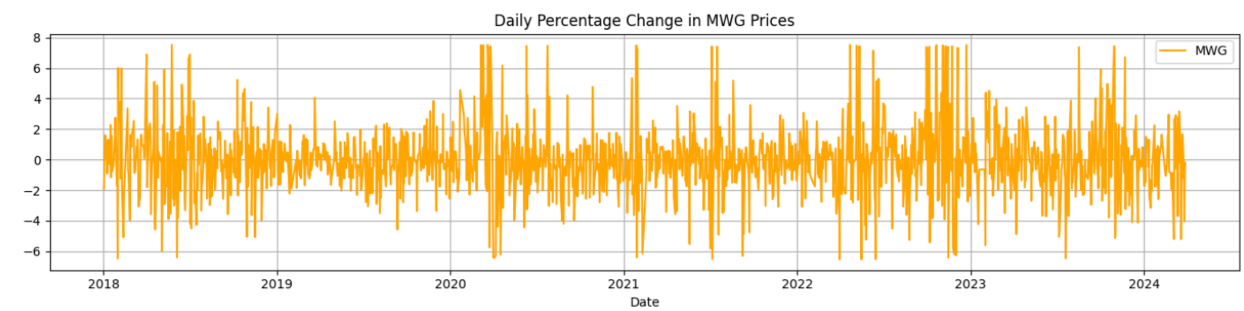

Figure 3: Daily Percentage Change in MWG Prices

Similar to HPG, the MWG stock shows considerable daily percentage changes, indicating volatility. Significant spikes in both positive and negative directions can be observed, particularly in 2020 and early 2023.

3. Remarks on Price Changes

HPG (Hoa Phat Group)

- Price Trend: The price of HPG shows significant fluctuations over the observed period. There are periods of both decline and growth.

- Volatility: The standard deviation of 9,174.28 indicates substantial volatility in the stock prices.

- Growth and Decline: The stock saw significant growth phases around 2021 but experienced notable declines afterward.

- Recovery: The stock showed a recovery trend towards the end of the observed period.

MWG (Mobile World Investment Corporation)

- Price Trend: MWG shows an overall increasing trend with notable peaks around mid-2021 and some volatility thereafter.

- Volatility: The higher standard deviation of 14,309.89 reflects greater volatility compared to HPG.

- Significant Peaks: The stock price peaked at 79,582.00, showing substantial growth from its minimum value.

- Consistent Growth: Despite fluctuations, MWG maintained a relatively consistent upward trend over the period.

Conclusion:

HPG shows considerable volatility with significant periods of both decline and growth. MWG demonstrates a more consistent upward trend with higher overall volatility.

III. Model Implementation

1. ARIMA Model

a. Data preprocessing

There is no need to preprocess the dataset beforehand when training an ARIMA model. Although the dataset needs to be stationary for the model to work, the differencing is part of the model fitting process and not a separate preprocessing step that you need to apply manually to the training data. We only need to statistically test the data to see if it is stationary or not to determine the d value (the order of differencing), plot the Partial Autocorrelation (PACF) plot to determine the p-value (the order of the Auto Regressive (AR) term) and plot the Autocorrelation (ACF) plot to determine the q value (the order of the Moving Average (MA) term).

b. Hyper-parameters

There are 3 hyper-parameters we need to determine before building an ARIMA model, the p, d, and q values. We applied the traditional method to find suitable p, d, and q values on the training dataset:

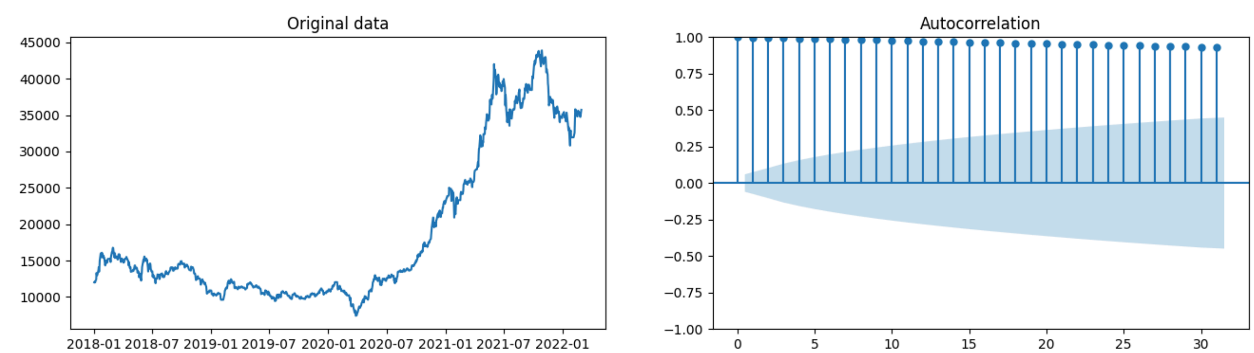

Figure 4: HPG Original

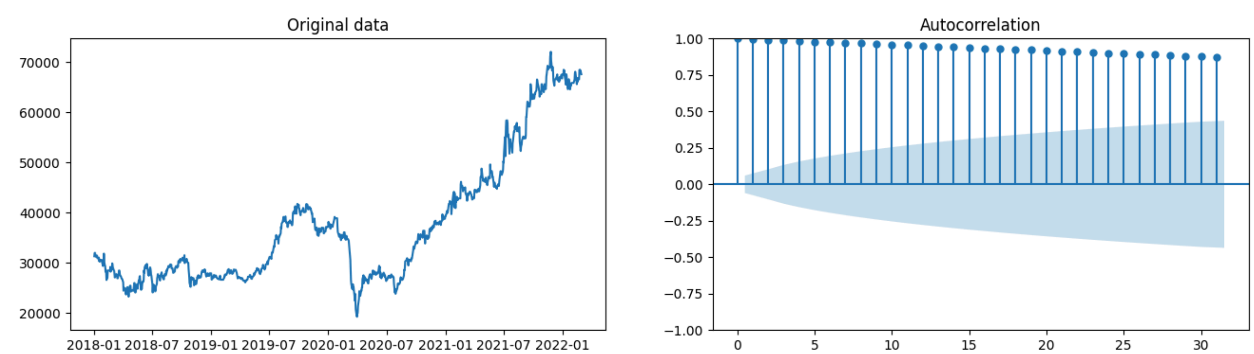

Figure 5: MWG Original

d = 1 for both datasets, because the original data is not stationary. The p-value after testing the dataset using the Augmented Dickey-Fuller (ADF) test is larger than 0.05. After first-order differencing, the dataset is randomly distributed around the mean of 0, the autocorrelation plot points hover around the significance line except lag 0 instead of getting close to 1 at all lags as before, also the p-value for ADF is lower than 0.05. After second-order differencing, the immediate lag went far into the negative side compared to first-order differencing which might indicate that the series has been over-differenced.

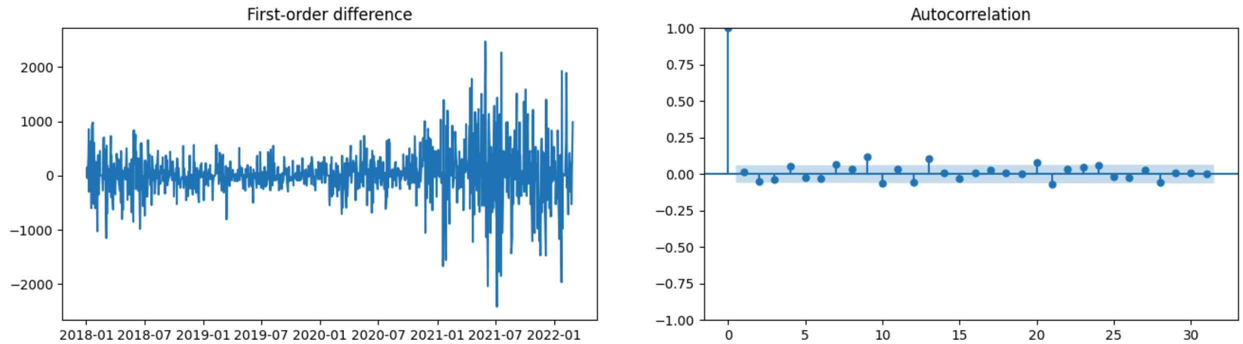

Figure 6: HPG First-order Differencing (d=1)

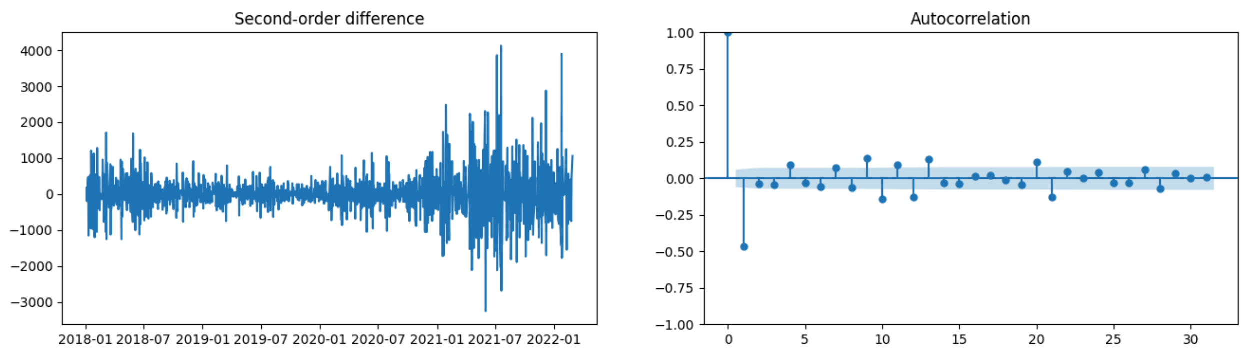

Figure 7: HPG Second-order Differencing (d=2)

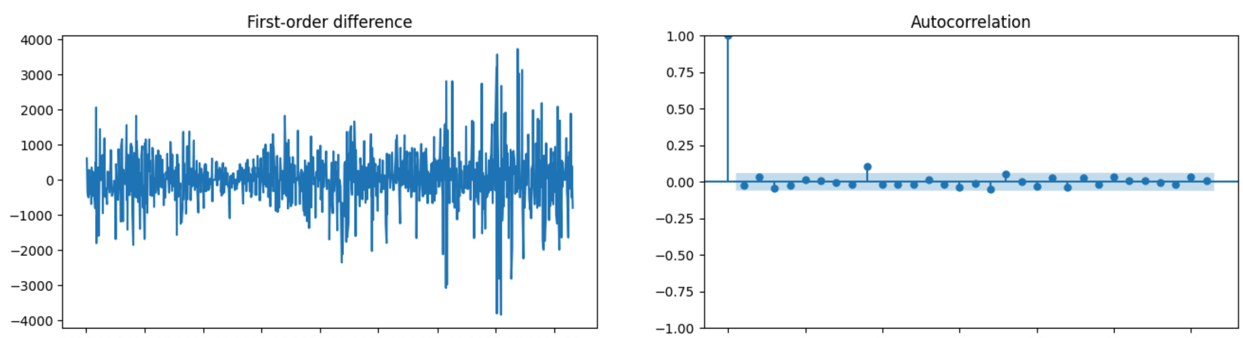

Figure 8: MWG First-order Differencing (d=1)

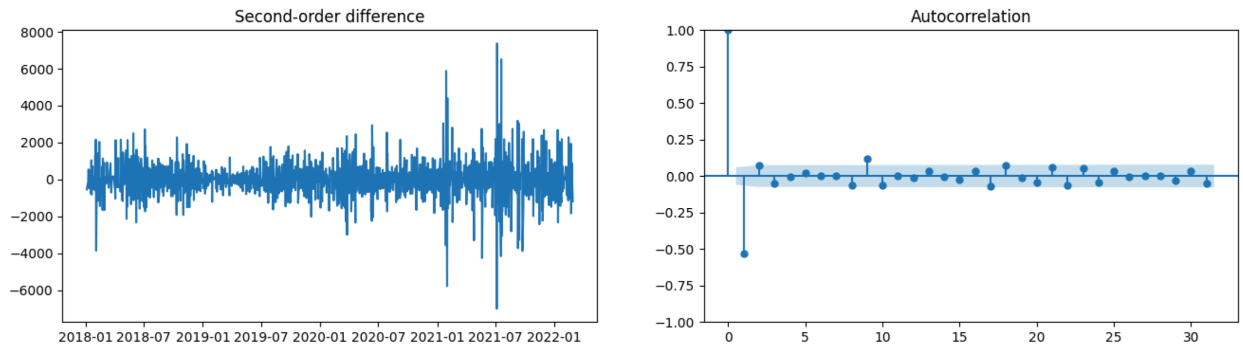

Figure 9: MWG Second-order Differencing (d=2)

p = 2 or 4 for HPG dataset and p = 3 or 9. for MWG dataset. To determine the value p or the order of the Auto Regressive (AR) term, we plot the Partial Autocorrelation (PACF) plot. The lag where the PACF cuts off after a significant spike will be chosen.

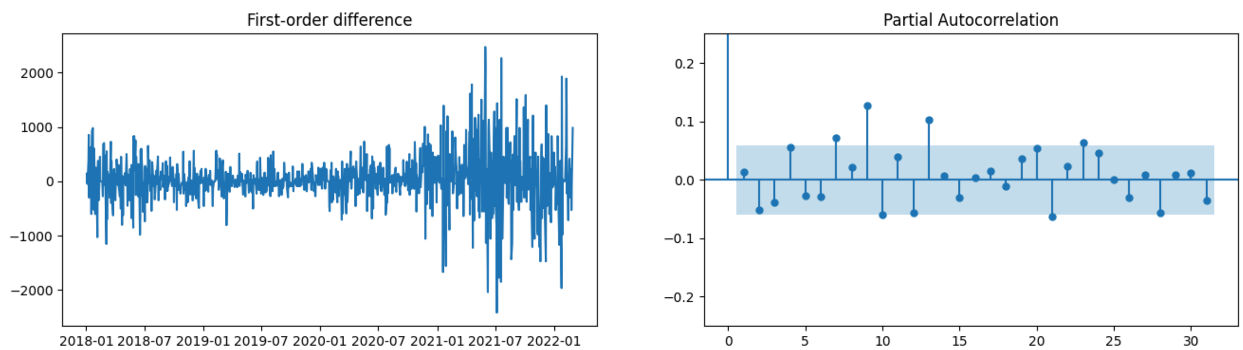

Figure 10: HPG Partial Autocorrelation (PACF)

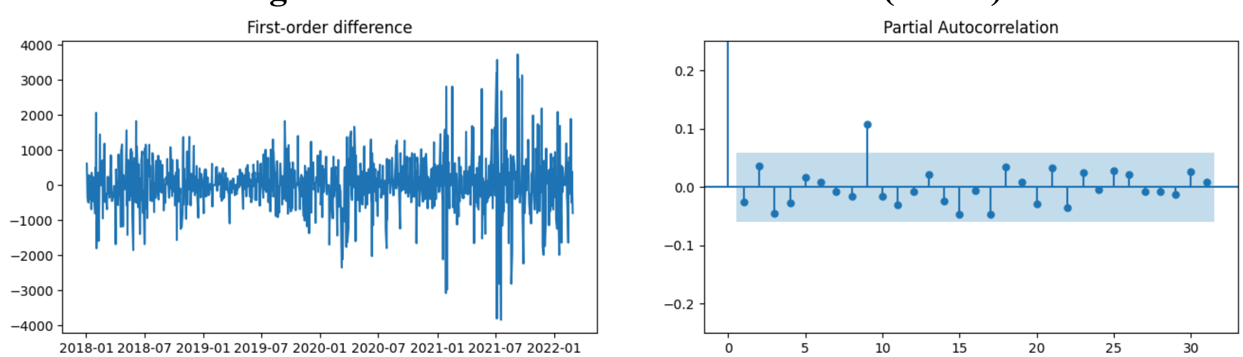

Figure 11: MWG Partial Autocorrelation (PACF)

q = 2 or 4 for HPG dataset and q = 3 or 9 for MWG dataset. To determine the value p or the order of the Auto Regressive (AR) term, we plot the Autocorrelation (ACF) plot. The lag where the ACF cuts off after a significant spike will be chosen.

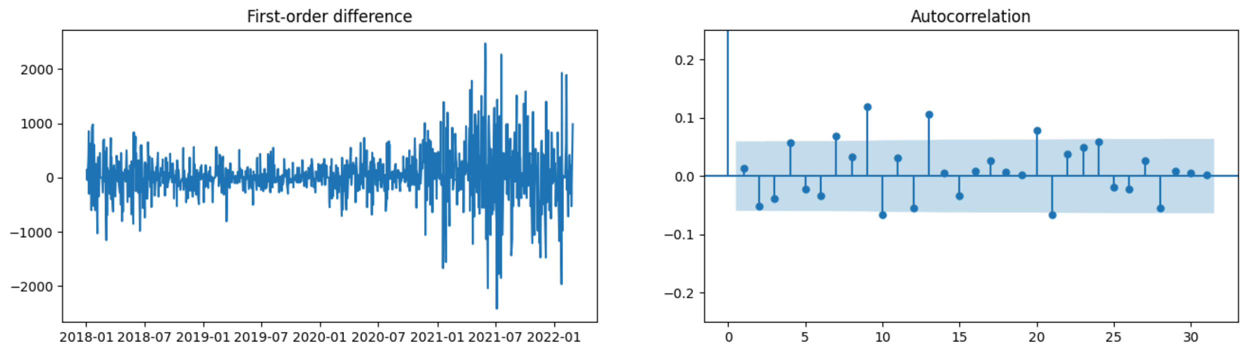

Figure 11: HPG Autocorrelation (ACF)

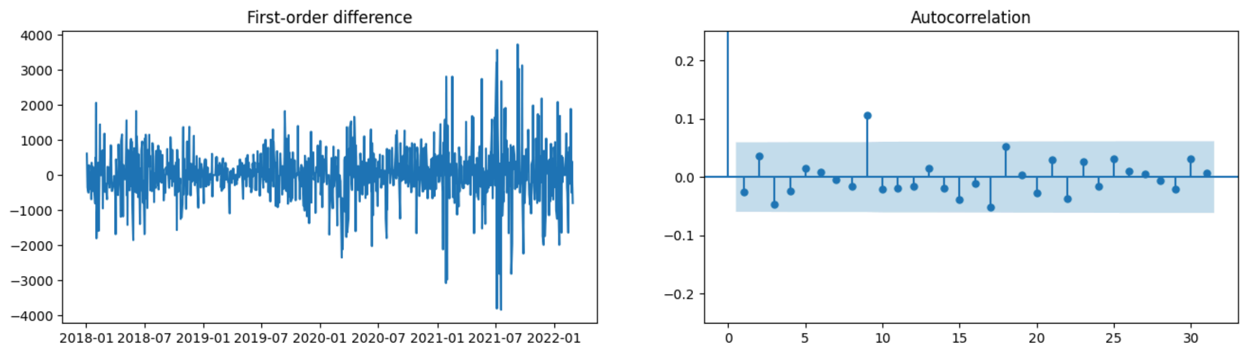

Figure 12: MWG Autocorrelation (ACF)

After testing the model, the value for p and q when going above 4 for both datasets will make the ARIMA model not able to converge properly using Maximum Likelihood optimization. So we decided to try p = 2 and q = 2 for the HPG dataset and p = 3 and q = 3 for the MWG dataset. We also implemented Auto ARIMA which uses a more systematic and automated approach to search through a range of possible models, evaluating each combination of p, d, and q based on specific criteria (like AIC, BIC) to find the best model. These criteria, namely the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), balance model fit and complexity, potentially leading to different p, d, and q values than those suggested by visual methods. After evaluating the models using AIC and BIC we decided to use the best model suggested by Auto ARIMA and conclude the final p, d, and q values for each model corresponding to its dataset**. Order(p, d, q) = (2, 1, 3) for HPG and (2, 1, 2) for MWG.**

c. Model Architecture

SARIMAX Results

Dep. Variable:	Price	No. Observations:	1085
Model:	ARIMA(2, 1, 3)	Log Likelihood	-8147.515
Date:	Sat, 01 Jun 2024	AIC	16307.030
Time:	20:56:24	BIC	16336.960
Sample:	01-02-2018	HQIC	16318.361
	- 02-28-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.7491	0.019	-40.426	0.000	-0.785	-0.713
ar.L2	-0.9456	0.018	-51.747	0.000	-0.981	-0.910
ma.L1	0.7686	0.024	31.891	0.000	0.721	0.816
ma.L2	0.9430	0.026	36.048	0.000	0.892	0.994
ma.L3	-0.0190	0.019	-1.006	0.314	-0.056	0.018
sigma2	1.979e+05	4592.856	43.090	0.000	1.89e+05	2.07e+05

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	1293.20
Prob(Q):	0.96	Prob(JB):	0.00
Heteroskedasticity (H):	4.97	Skew:	0.20
Prob(H) (two-sided):	0.00	Kurtosis:	8.34

Figure 13: ARIMA Model Architecture for HPG Dataset

SARIMAX Results

Dep. Variable:	Price	No. Observations:	1085
Model:	ARIMA(2, 1, 2)	Log Likelihood	-8686.462
Date:	Sat, 01 Jun 2024	AIC	17382.925
Time:	20:58:14	BIC	17407.867
Sample:	01-02-2018	HQIC	17392.367
	- 02-28-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.1626	0.042	-3.859	0.000	-0.245	-0.080
ar.L2	-0.9047	0.045	-20.280	0.000	-0.992	-0.817
ma.L1	0.1744	0.034	5.064	0.000	0.107	0.242
ma.L2	0.9434	0.037	25.445	0.000	0.871	1.016
sigma2	5.409e+05	1.33e+04	40.759	0.000	5.15e+05	5.67e+05

Ljung-Box (L1) (Q):	1.42	Jarque-Bera (JB):	983.98
Prob(Q):	0.23	Prob(JB):	0.00
Heteroskedasticity (H):	3.01	Skew:	0.01
Prob(H) (two-sided):	0.00	Kurtosis:	7.67

Figure 14: ARIMA Model Architecture for MWG Dataset

2. LSTM Model

a. Data Preprocessing

For this project, we employ two scaling techniques: MinMaxScaler and StandardScaler. These techniques ensure that the data is transformed to a suitable range and distribution, which is essential for the performance and convergence of the LSTM network.

- MinMaxScaler: scales the data to a specified range, typically between 0 and 1. This scaling method preserves the relationships in the data and is particularly useful for neural networks, which can be sensitive to the scale of input data.
- StandardScaler: standardizes the data to have a mean of 0 and a standard deviation of 1. This normalization is beneficial when the features have different units or magnitudes, ensuring that the LSTM model treats each feature equally.

b. Hyper-parameters

There are some hyper-parameters we need to determine before building an LSTM model, the number of inputs which indicates the input sequence length that the model considers as past data we denote as n_input, the number of LSTM layers, the number of units in the LSTM layer, the learning rate, the batch size and the number of epochs.

- By setting n_input to 20 for both datasets, we balance the trade-off between capturing relevant temporal patterns and maintaining computational efficiency. This length is chosen based on domain knowledge and empirical analysis, ensuring that the model has enough context to understand and predict future values without overfitting to noise or irrelevant fluctuations.
- We only used 1 LSTM layer with 64 units as the first layer for HPG and 1 LSTM layer with 16 units for MWG as they proved to perform best to our knowledge for their corresponding dataset. The ReLU (Rectified Linear Unit) activation function is chosen for its ability to mitigate the vanishing gradient problem and enhance the training speed.
- The Adam optimizer with a learning rate of 0.01 is used for both datasets. We also implemented the ReduceLROnPlateau callback which reduces the learning rate by a factor of 0.2 if the validation loss does not improve for 5 consecutive epochs, with a minimum learning rate set to 0.001. This helps in fine-tuning the learning process and prevents the model from getting stuck in local minima.
- The batch size and number of epochs are the same across both datasets which are 32 and 30 epochs.

c. Model Architecture

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_5 (LSTM)               (None, 64)                16896

 dense_5 (Dense)             (None, 1)                 65


=================================================================
Total params: 16961 (66.25 KB)
Trainable params: 16961 (66.25 KB)
Non-trainable params: 0 (0.00 Byte)

_________________________________________________________________
```

Figure 15: LSTM Model Architecture for HPG Dataset

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_3 (LSTM)               (None, 16)                1152

 dense_3 (Dense)             (None, 1)                 17


=================================================================
Total params: 1169 (4.57 KB)
Trainable params: 1169 (4.57 KB)
Non-trainable params: 0 (0.00 Byte)

_________________________________________________________________
```

Figure 16: LSTM Model Architecture for MWG Dataset

The Dense layer is 1 because we want to predict 1 output which is the next day of the past data.

3. GRU Model

a. Data preprocessing:

For this project, we employ two scaling techniques: MinMaxScaler and StandardScaler. These techniques ensure that the data is transformed to a suitable range and distribution, which is essential for the performance and convergence of the GRU network.

- MinMaxScaler: scales the data to a specified range, typically between 0 and 1. This scaling method preserves the relationships in the data and is particularly useful for neural networks, which can be sensitive to the scale of input data.
- We didn't use StandardScaler because standardization might not provide additional benefits for GRU models and could complicate the preprocessing pipeline without significant performance gains.

b. Hyper-parameters:

There are some hyper-parameters we need to determine before building a GRU model, the number of inputs which indicates the input sequence length that the model considers as past data we denote it as n_input, the number of GRU layers, the number of units in the GRU layer, the learning rate, the batch size and the number of epochs.

- By setting n_input to 20 for both datasets, we balance the trade-off between capturing relevant temporal patterns and maintaining computational efficiency. This length is chosen based on domain knowledge and empirical analysis, ensuring that the model has enough context to understand and predict future values without overfitting to noise or irrelevant fluctuations.
- We only used 1 GRU layer with 64 units as the first layer for both dataset as they proved to perform best to our knowledge for their corresponding dataset. The ReLU (Rectified Linear Unit) activation function is chosen for its ability to mitigate the vanishing gradient problem and enhance the training speed.
- The Adam optimizer with a learning rate of 0.01 is used for both datasets. We also implemented the ReduceLROnPlateau callback which reduces the learning rate by a factor of 0.2 if the validation loss does not improve for 5 consecutive epochs, with a minimum learning rate set to 0.001. This helps in fine-tuning the learning process and prevents the model from getting stuck in local minima.
- The batch size and number of epochs are the same across both datasets which are 32 and 30 epochs.

- The hyper-parameters for the GRU model isn't that different compared to LSTM because they are very similar in terms of use case and architecture, GRU was invented after LSTM, needs less computation resources to train but isn't as accurate as LSTM when the input sequence is too long to remember/process.

c. Model architecture:

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

 dense (Dense)               (None, 1)                 65

=================================================================
Total params: 12929 (50.50 KB)
Trainable params: 12929 (50.50 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 17: GRU Model Architecture

The Dense layer is 1 because we want to predict 1 output which is the next day of the past data.

IV. Model Evaluation

1. ARMIA Model

a. HPG Dataset

The ARIMA model, after being fine-tuned with parameters (p,d,q) = (2,1,3) was tested on the HPG dataset. The performance of the model on the test data was not satisfactory. The predicted prices were flat, which suggests that the model did not capture the dynamic trends in the data. This is likely due to the lack of strong seasonality or trends in the dataset, as mentioned by Rob Hyndman. The inability to predict varying future prices indicates that the ARIMA model might not be suitable for the HPG dataset, where high volatility and non-stationary characteristics are prevalent.

Metric Results:

- R-squared: -4.81538007423587
- MAPE: 76.84361372688339
- RMSE: 15308.108440283899

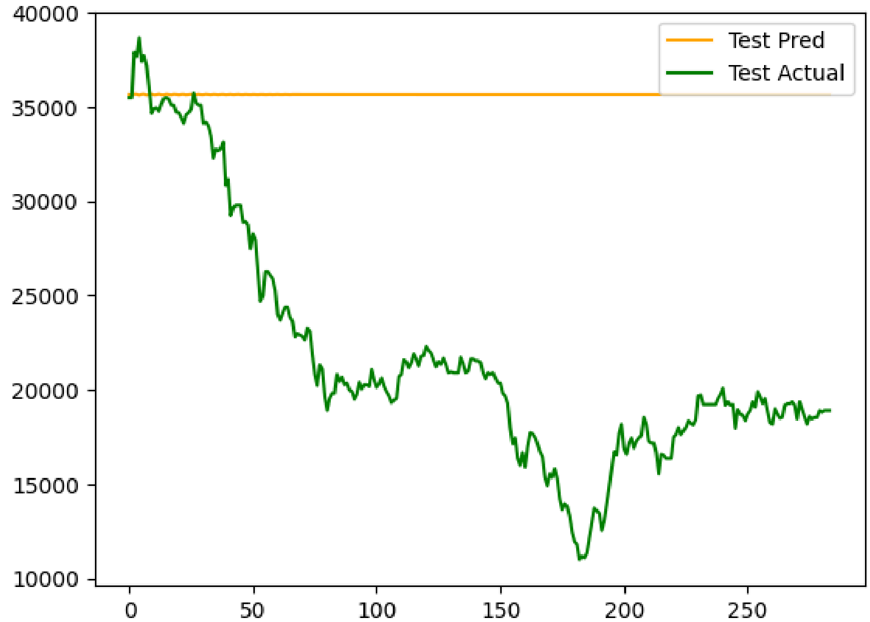

Figure 18: ARIMA Model Predict HPG Dataset

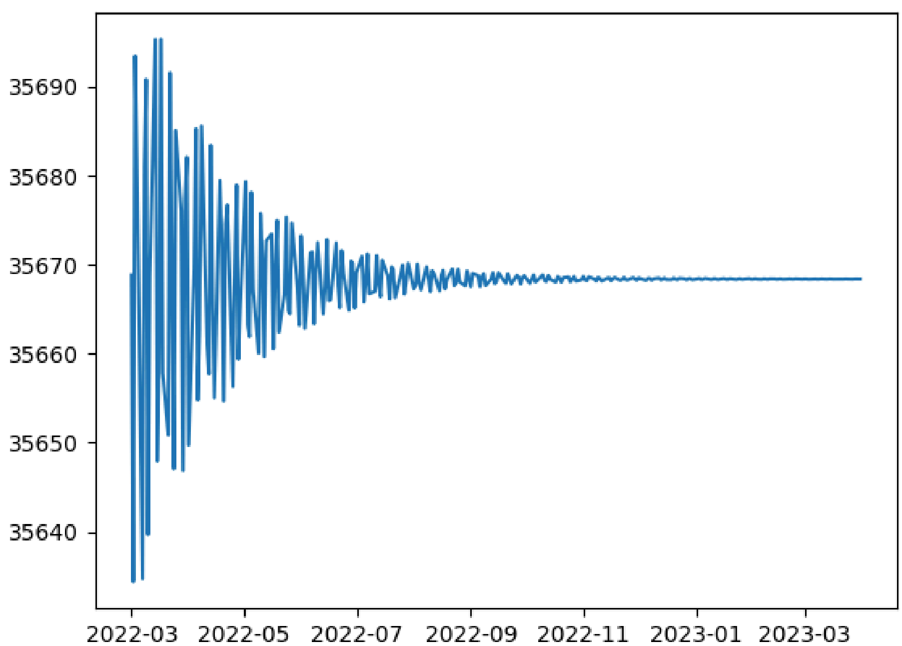

Figure 19: ARIMA Model Predict HPG Dataset in detailed

Metric Results:

- R-squared: 0.9912297292312797
- MAPE: 2.1346349671789904
- RMSE: 594.4823689030817

Figure 20: ARIMA Model Predict HPG Dataset when fed with data

b. MWG Dataset

For the MWG dataset, the ARIMA model with parameters (p,d,q) = (2,1,2) also demonstrated flat predictions on the test data. This flat forecast suggests that the model could not effectively learn from the historical data to predict future values with varying trends. Despite MWG exhibiting a more consistent upward trend compared to HPG, the ARIMA model struggled with the inherent volatility and complexity of the stock price movements.

Metric Results:

- R-squared: -0.5344363910087773
- MAPE: 27.8726630936171
- RMSE: 16399.787193679385

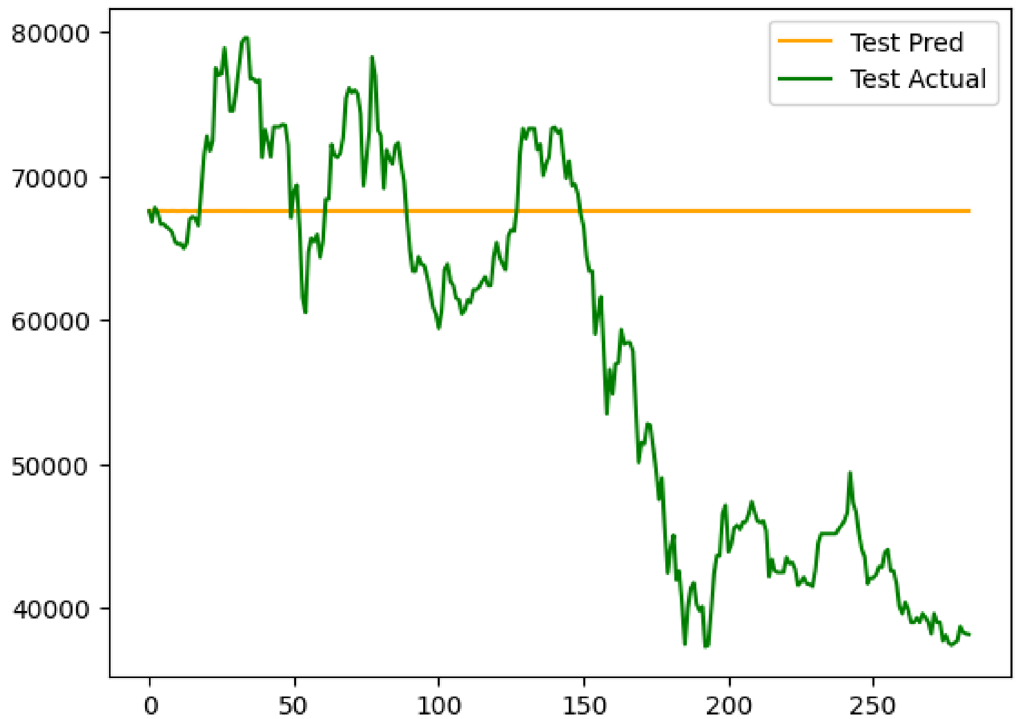

Figure 21: ARIMA Model Predict MWG Dataset

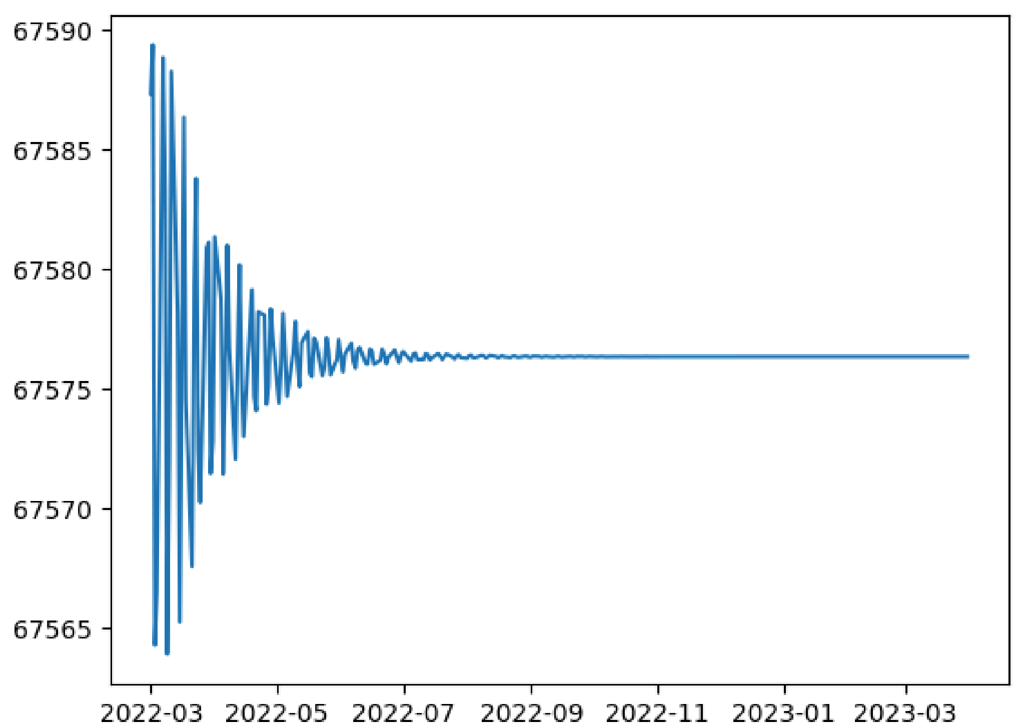

Figure 22: ARIMA Model Predict HPG Dataset in detailed

Metric Results:

- R-squared: 0.9856719231575299
- MAPE: 1.9380135130635707
- RMSE: 1584.738900737968

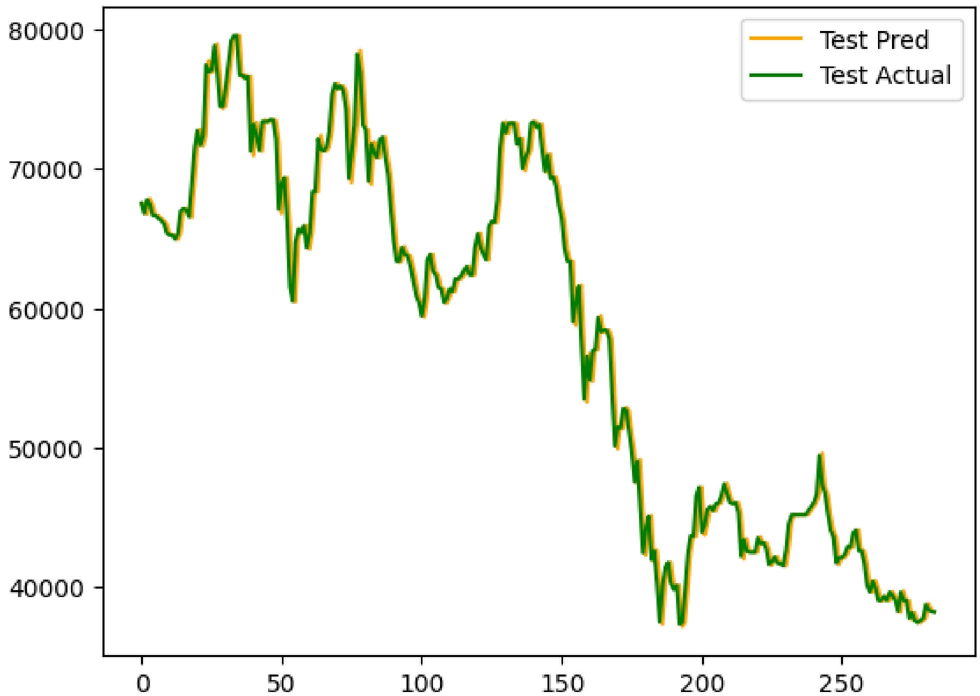

Figure 23: ARIMA Model Predict MWG Dataset when fed with test data

c. Conclusion for ARIMA Model

Why ARIMA outputs a flat prediction on test data while fitted on train data: According to Rob Hyndman (the author of the forecast package), “A point forecast is (usually) the mean of the distribution of a future observation in the time series, conditional on the past observations of the time series. It is possible, even likely in some circumstances, that the future observations will have the same mean and then the forecast function is flat. A random walk model will return a flat forecast function (equal to the last observed value of the series). An ETS(A, N, N) model will return a flat forecast function. An iid model will return a flat forecast function (equal to the mean of the observed data). This is not a bug. It is telling you something about the time series – namely that there is no trend, no seasonality, and insufficient temporal dynamics to allow the future observations to have different conditional means.” Basically means that when the historical data doesn't have strong seasonality the forecasting model may find it difficult to predict the future. Therefore it simply takes the average of your previous values. The flat predictions suggest that ARIMA might not be the best choice for forecasting stock prices in this context.

2. LSTM Model

a. HPG Dataset

The LSTM model, after being trained with 20 input sequences, 64 units in the LSTM layer, and using the Adam optimizer with a learning rate of 0.01, showed a significant improvement over ARIMA. The model was able to capture the trends and fluctuations in the training and validation sets. However, when applied to the test set, the predictions still showed some deviation from the actual prices but were much closer compared to ARIMA. The LSTM model managed to capture the overall trend and some of the volatility in the HPG stock prices, indicating a better understanding of the temporal dependencies in the data.

Metric Results:

- R-squared: 0.9853048712610946
- MAPE: 2.8790802036592646
- RMSE: 769.5187108910271

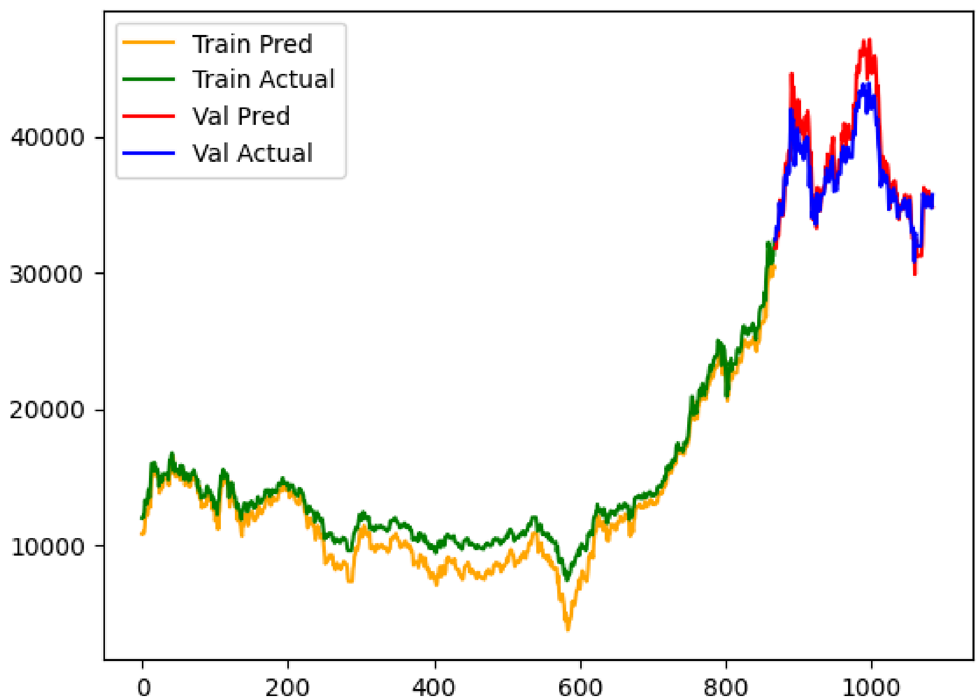

Figure 24: LSTM Model Predict HPG Train + Validation Data

Figure 25: LSTM Model Predict HPG Test Data

b. MWG Dataset

For the MWG dataset, the LSTM model trained with similar parameters (1 LSTM layer with 16 units) performed well on the training and validation sets, showing good alignment with actual prices. On the test set, the model's predictions were relatively close to the actual stock prices, capturing the upward trend and the volatility. This performance demonstrates the LSTM model's capability to handle complex time series data, making it a more suitable choice for MWG stock price prediction compared to ARIMA.

Metric Results:

- R-squared: 0.8992728198608252
- MAPE: 5.970068713682484
- RMSE: 4201.81571558336

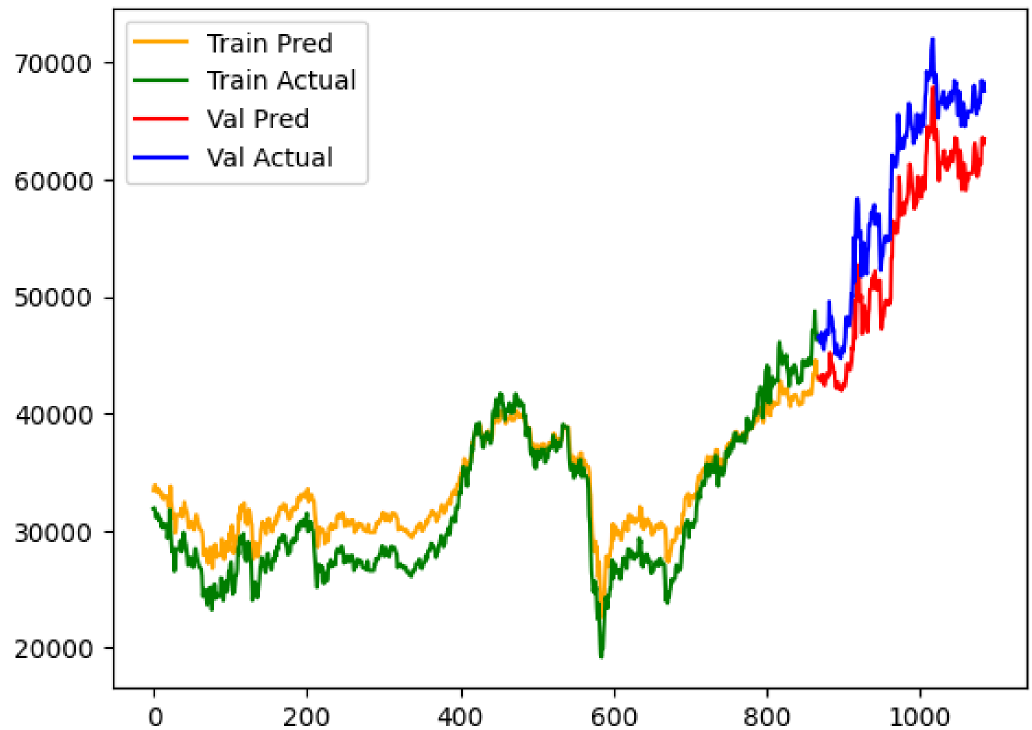

Figure 26: LSTM Model Predict MWG Train + Validation Data

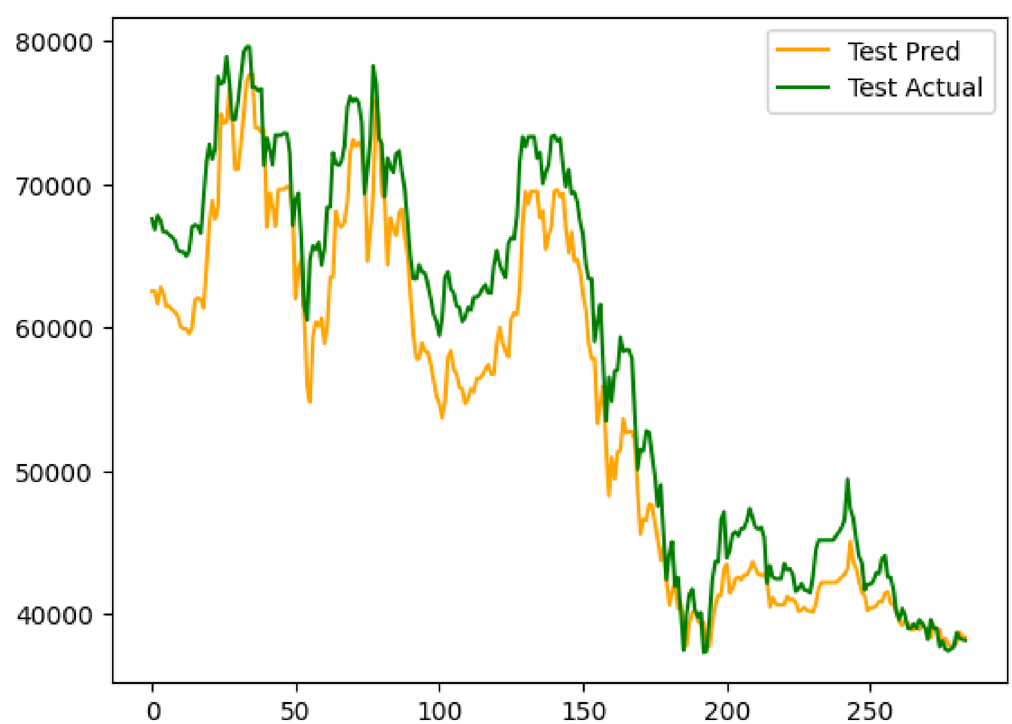

Figure 27: LSTM Model Predict MWG Test Data

c. Conclusion for LSTM

The LSTM model outperformed ARIMA in predicting stock prices for both HPG and MWG. It demonstrated a better capability to capture the temporal patterns and volatility inherent in the stock price data, making it a more robust and reliable model for this purpose.

3. GRU Model

a. HPG Dataset

The GRU model, trained with similar hyper-parameters as the LSTM model (20 input sequences, 64 units in the GRU layer), also showed an improvement over the ARIMA model. The GRU model's predictions on the training and validation sets were accurate, capturing the trends and fluctuations effectively. On the test set, the GRU model's predictions were close to the actual prices, similar to the LSTM model. The GRU model managed to understand the temporal dependencies and volatility in the HPG stock prices, providing reliable predictions.

Metric Results:

- R-squared: 0.9817207992953129
- MAPE: 3.2228727832979254
- RMSE: 858.2445969595135

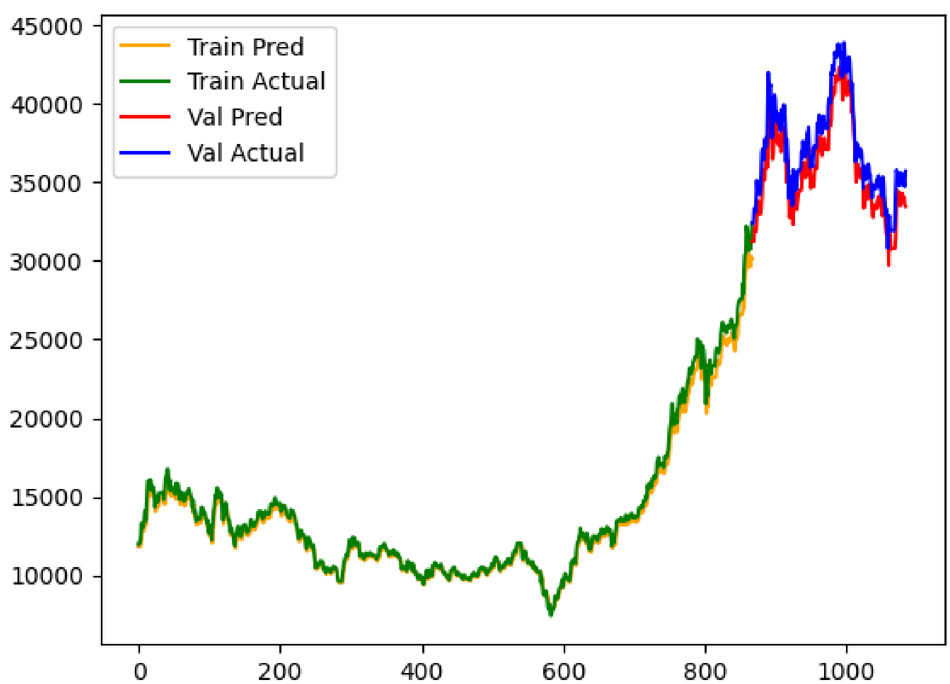

Figure 28: GRU Model Predict HPG Train + Validation Data

Figure 29: GRU Model Predict HPG Test Data

b. MWG Dataset

For the MWG dataset, the GRU model (with 1 GRU layer and 64 units) performed well on both the training and validation sets, aligning closely with the actual stock prices. On the test set, the GRU model's predictions were accurate, capturing the upward trend and the inherent volatility of the MWG stock prices. This indicates that the GRU model is capable of handling the complexities of the stock price data, similar to the LSTM model.

Metric Results:

- R-squared: 0.9854582981162439
- MAPE: 1.9784996605462388
- RMSE: 1596.5088030953464

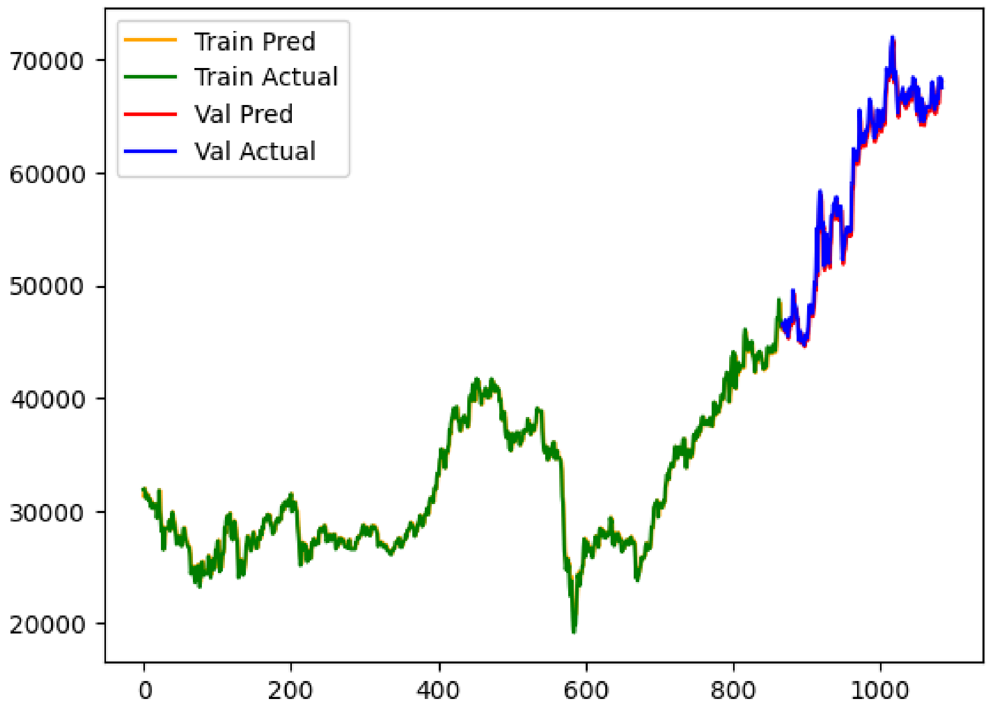

Figure 30: GRU Model Predict MWG Train + Validation Data

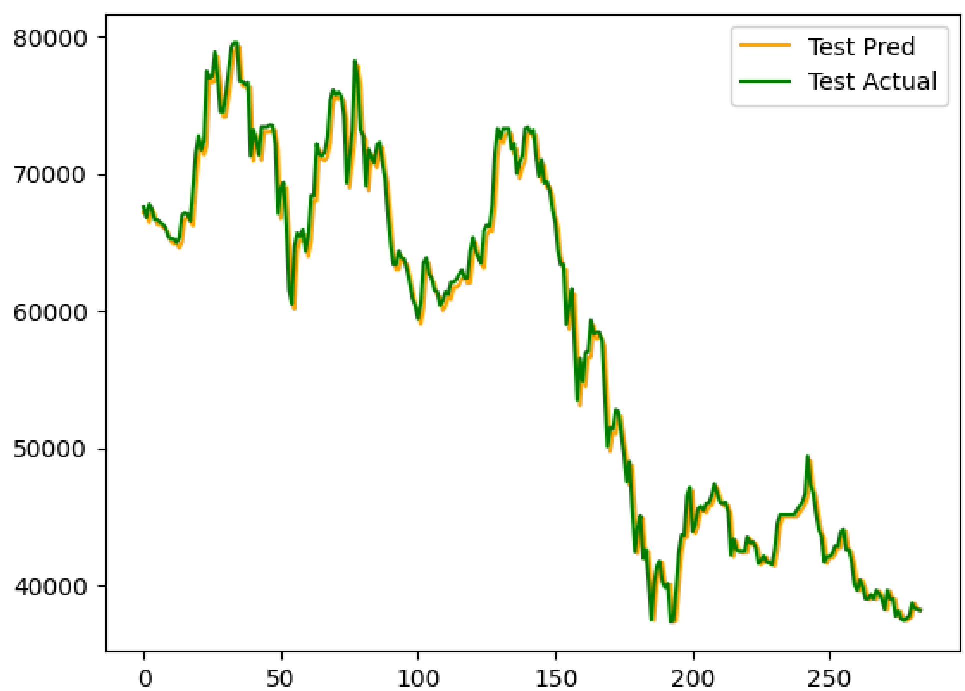

Figure 31: GRU Model Predict MWG Test Data

c. **Conclusion for GRU**

The GRU model's performance was comparable to that of the LSTM model, outperforming ARIMA in predicting stock prices for both HPG and MWG. The GRU model effectively captured the temporal dependencies and volatility, making it a suitable alternative to LSTM for stock price prediction.

4. Performance Comparison between ARIMA and LSTM Model

Table 1: Performance Comparison between ARIMA and LSTM Model

		ARIMA	ARIMA fed with test data	LSTM
HPG Dataset	R-squared	-4.8154	0.9912	0.9853
	MAPE	76.84%	2.13%	2.88%
	RMSE	15,308.11	594.48	769.52
MWG Dataset	R-squared	-0.5344	0.9856	0.8993
	MAPE	27.87%	1.93%	5.97%
	RMSE	16,399.79	1,584.73	4,201.82

The ARIMA model produced flat predictions for both datasets, indicating that it failed to capture the dynamic trends in the stock prices. The negative R-squared values and high MAPE suggest that the model's predictive performance was unsatisfactory, especially for the HPG dataset. This is likely due to the lack of strong seasonality and the high volatility in the stock prices, which ARIMA models struggle to handle.

ARIMA fed with test data approach demonstrates that by focusing on short-term predictions (one day ahead), the ARIMA model can significantly enhance its performance. This method is more effective for time series data with high volatility and non-linear patterns, as it allows the model to adapt to the latest observed data points, leading to more accurate short-term forecasts.

The LSTM model significantly outperformed the ARIMA model. For the HPG dataset, the LSTM model managed to capture the trends and fluctuations, resulting in much closer predictions to the actual prices. The R-squared value of 0.9853 indicates a high level of accuracy, with a low MAPE of 2.88%. Similarly, for the MWG dataset, the LSTM model showed a good alignment with actual prices, capturing the upward trend and inherent volatility more effectively than ARIMA

The comparison between ARIMA and LSTM models clearly shows that the LSTM model is superior in forecasting stock prices for both HPG and MWG. The ARIMA model's inability to capture the dynamic nature of the stock prices resulted in flat and inaccurate predictions. In contrast, the LSTM model demonstrated a better understanding of temporal dependencies and volatility, making it a more robust and reliable choice for stock price prediction in this context.

But while the LSTM model outperforms the modified ARIMA model overall, the ARIMA fed with test data method provides a substantial improvement over the standard ARIMA approach, making it a viable option for specific forecasting scenarios, particularly when computational simplicity and interpretability are important.

Overall, the LSTM model's performance highlights its suitability for handling complex time series data, particularly in scenarios with significant volatility and non-linear patterns.

.