Using Forecasting Models in Cryptocurrency Price Prediction

HOANG NHAT MINH[1], LE NGUYEN NHAT MINH[2], AND TRAN THI LUYEN[3]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21522336@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21522338@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21521107@gm.uit.edu.vn)

ABSTRACT Predicting cryptocurrency price has become a central focus for investors and traders in the rapidly evolving landscape of digital assets. Cryptocurrencies like Bitcoin, Ethereum, and other altcoins have captured global attention, with their prices subject to fluctuations influenced by various factors, from market sentiment to technological developments. With the unprecedented volatility of these markets, the need for accurate forecasting models has never been more crucial. In this research, we conduct an in-depth analysis using various models including Linear regression (LR), Autoregressive Integrated Moving Average (ARIMA), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Seasonal Autoregressive Integrated Moving Average with eXogenous factors (SARIMAX), Neural Basis Expansion Analysis Time Series (NBEATS), Random Forest (RF) to predict prices of Bitcoin, Ethereum and BNB.

INDEX TERMS Cryptocurrency price prediction, Bitcoin, Ethereum, BNB, Linear Regression, ARIMA, RNN, GRU, LSTM, SARIMAX, NBEATS, Random Forest.

I. INTRODUCTION

Cryptocurrency, a digital form of currency, has transformed the traditional concept of money. Unlike cash, cryptocurrencies exist purely in electronic form, utilizing encryption techniques for secure transactions. Bitcoin, the pioneering cryptocurrency introduced in 2009 by Satoshi Nakamoto, paved the way for a myriad of digital currencies that followed. The decentralized nature of cryptocurrencies means they operate without a central authority, using blockchain technology to record and verify transactions across a network of computers. This decentralization provides a level of transparency and security previously unseen in financial systems. As interest and adoption continue to grow, cryptocurrencies are reshaping the landscape of finance, offering individuals around the globe new avenues for transactions, investments, and financial freedom.

By using forecasting models and machine learning algorithms to analyze historical data of Bitcoin, Ethereum, and BNB, the research objective is to provide valuable insights for investors and traders to forecast the cryptocurrency price movement. To evaluate the performance of each predictive model, we use three evaluation metrics Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE).

II. RELATED WORKS

Mahir Iqbal and his team [1] applied a machine learning based time series analysis in predicting the price of Bitcoin. They used ARIMA, FBProphet, and XG Boosting for time series analysis and evaluated these models by using three parameters RMSE, MAE, and R Squared (R^2). After conducting the analysis, they concluded that ARIMA is the best of three with RMSE score and MAE score are 322.4 and 227.3 respectively.

Harsha Nanda Gudavalli and Khetan Venkata Ratnam Kancherla [2] identified four machine learning algorithms RF, Gradient Boosting (GB), LSTM, and GRU to make long-term predictions of cryptocurrency prices. Overall, LSTM is the most accurate model based on RMSE score of 0.01083, MSE score of 0.00011, and R Squared score of 0.80618.

Junyi Zhu [3] applied LR for Bitcoin price prediction with the accuracy rate for training RSS and testing RSS are 94.6% and 96.99% respectively.

Namrata Hemraj Gawali [4] developed predictive models to predict the price of Bitcoin using LSTM, Seasonal Autoregressive Integrated Moving Average (SARIMA), SARIMAX, and RNN. In conclusion, LSTM outperformed all other models with R Squared score of 0.9519, MAE score of 2476.68, MSE score of 10958440.29 and RMSE score of

3310.35.

Alikhan Bulatov [5] used N-BEATS trained on Bitcoin daily, hourly, and up-to-the-minute data in comparison with LSTM and ARIMA. In this analysis, it is concluded that N-BEATS is the best model for forecasting the Bitcoin price with the RMSE score in daily, hourly, and minute data are 308.859, 59.303, and 13.678 respectively.

III. MATERIALS

A. DATASET

This research focuses on analyzing cryptocurrency trends through three data sets of three cryptocurrencies:

- BTC/USD
- ETH/USD
- BNB/USD

Those data sets are extracted from the website *investing.com* with the selected period from March 1, 2019, to June 1, 2024. Each data set contains 1920 data rows and has 6 attributes:

- Price: The last transacted price in a day.
- Open: The opening price in a day.
- High: The highest price in a day.
- Low: The lowest price in a day.
- Vol: The total amount traded in a day.
- Change %: The percentage change in the closing price compared to the previous day's closing price.

B. DESCRIPTIVE STATISTICS

TABLE 1. BTC, ETH, BNB's Descriptive Statistics

	BTC	ETH	BNB
Count	1920	1920	1920
Mean	27916.88	1580.30	229.54
Std	17918.78	1205.85	184.43
Min	3715	107.58	9.28
25%	10434.10	268.88	27.03
50%	25933.90	1622.45	247.55
75%	41492.60	2333.36	332.12
Max	73060.80	4808.09	676.12

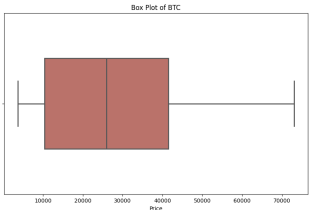

FIGURE 1. Bitcoin price's boxplot

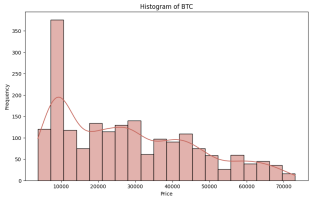

FIGURE 2. Bitcoin price's histogram

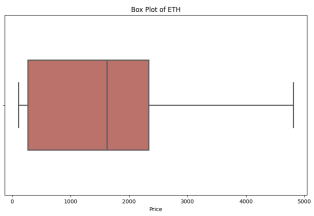

FIGURE 3. Ethereum price's boxplot

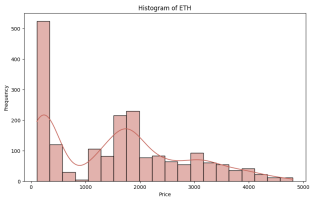

FIGURE 4. Ethereum price's histogram

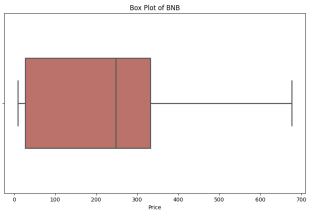

FIGURE 5. BNB price's boxplot

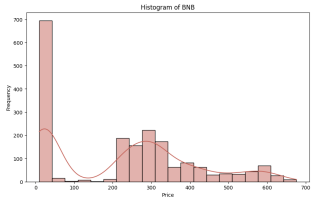

FIGURE 6. BNB price's histogram

IV. METHODOLOGY

A. LINEAR REGRESSION

Linear regression [**?**] is an algorithm that provides a linear relationship between an independent variable and a dependent variable. A linear regression model with more than one independent variable is called a multiple linear regression model. A multiple linear regression model has the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k + \varepsilon$$

Where:

- Y is the dependent variable.
- $X_1, ..., X_k$ are the independent (explanatory) variables.
- β_0 is the intercept term.
- $\beta_1, ..., \beta_k$ are the regression coefficients for the independent variables.
- ε is the error term.

B. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)

ARIMA [**?**], which stands for AutoRegressive Integrated Moving Average, is a statistical analysis model designed to analyze time series data. It helps in understanding the data set more comprehensively and predicting future trends. The complete model is represented as follows:

$$y'_t = c + \phi_1 y'_{t-1} + ... + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Where:

- y'_t is the value of the time series at time t.
- c is the constant term.
- $\phi_1, ..., \phi_p$ are the autoregressive coefficients.
- $\theta_1, ..., \theta_q$ are the moving average coefficients.
- ε_t is the white noise at time t.

A non - seasonal ARIMA model is classified as an ARIMA (p, d, q) model, where:

- p is the number of lag observations included in the model.
- d is the number of times that the raw observations are differenced.

- q is the size of the moving average window, also called the order of moving average.

C. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE WITH EXOGENOUS FACTORS (SARIMAX)

SARIMAX [**?**], which stands for Seasonal Autoregressive Integrated Moving Average with eXogenous factors, is an enhanced version of the SARIMA model, incorporating external feature parameters (X) to improve performance. By including exogenous factors, SARIMAX reduces prediction errors, addresses autocorrelation issues, and enhances prediction accuracy. The model captures both seasonal effects and exogenous factors, represented as SARIMAX (p, d, q) * (P, D, Q), with the exogenous factors being optional parameters. These factors provide additional details to support and refine the prediction model. The SARIMAX can be represented as follows:

$$d_t = c + \sum_{n=1}^{p} \alpha_n d_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \sum_{n=1}^{r} \beta_n x_{n_t} + \sum_{n=1}^{P} \phi_n d_{t-sn} + \sum_{n=1}^{Q} \eta_n \epsilon_{t-sn} + \epsilon_t$$

Where:

- P is seasonal autoregressive order.
- D is seasonal differencing order.
- Q is seasonal moving average order.
- s is the length of the seasonal cycle.

D. RANDOM FOREST

Random Forest [**?**] is an ensemble learning method used for classification, regression, and various other tasks. It works by constructing numerous decision trees during training and then outputs either the mode of the classes for classification tasks or the mean prediction for regression tasks, based on the individual trees' results.

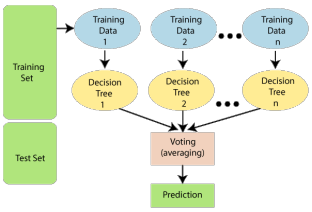

FIGURE 7. The structure of Random Forest model

Each tree in a random forest samples subsets of the training data randomly through a process called bootstrap aggregating. The model is then fitted to these smaller data sets, and the predictions are aggregated. This method allows multiple instances of the same data to be used repeatedly through replacement sampling. As a result, the trees are trained on different sets of data and utilize different features for decision-making.

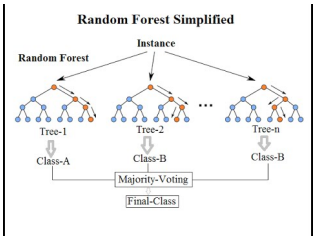

FIGURE 8. Overview of Random Forest model

The following steps explain the working Random Forest algorithm:

1) From the original dataset, select random samples of data .
2) Train a decision tree model on each sample.
3) For each tree, select a random subset of features to consider at each split.
4) Combine the individual decision trees into a final model by taking a majority vote for classification or averaging for regression.

E. RECURRENT NEURAL NETWORK (RNN)

Recurrent Neural Networks [**?**] (RNNs) are a type of artificial neural network specifically designed for processing sequential data. Unlike traditional feedforward neural networks, RNNs have cyclic connections that enable them to retain a memory of previous inputs.

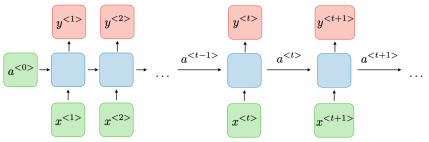

FIGURE 9. Overview of Recurrent Neural Network

For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

and

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Where:

- $W_{ax}, W_{as}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally.
- g_1, g_2 are activation functions.

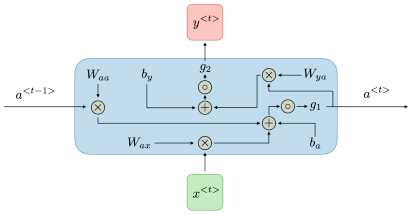

FIGURE 10. Architecture of a traditional RNN

For a recurrent neural network, the loss function $\mathcal{L}$ over all time steps is defined based on the loss at each individual time

step:

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>})$$

Backpropagation is done at each point in time. At timestep *T*, the derivative of the loss $\mathcal{L}$ with respect to weight matrix *W* is expressed as follows:

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}^{(T)}}{\partial W}\Big|_{(t)}$$

F. LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory [**?**] (LSTM) networks are a type of Recurrent Neural Network (RNN) designed to capture long-term dependencies in sequential data. LSTMs can process and analyze various forms of sequential data, including time series, text, and speech. They utilize a memory cell and gates to regulate the flow of information, enabling them to selectively retain or discard information as necessary. This mechanism helps them avoid the vanishing gradient problem that affects traditional RNNs.

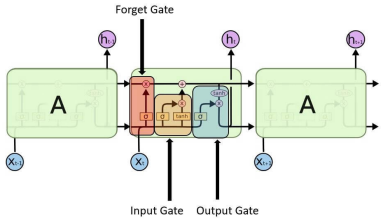

FIGURE 11. Architecture of LSTM

LSTM have 4 different components: cell state (memory cell), forget gate, input gate, output gate.

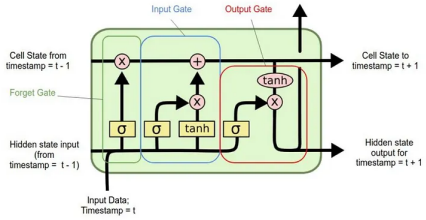

FIGURE 12. LSTM components

- Forget gate:

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

- Input gate:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C)$$

- Output gate:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * tanh(C_t)$$

G. GATED RECURRENT UNIT (GRU)

A Gated Recurrent Unit [**?**] (GRU) is a type of recurrent neural network (RNN) architecture designed to model sequential data by selectively retaining or forgetting information over time. GRUs simplify the RNN structure by utilizing two main gates: the reset gate and the update gate. The reset gate controls how much of the previous hidden state should be forgotten, while the update gate decides how much of the candidate activation vector should be integrated into the new hidden state.

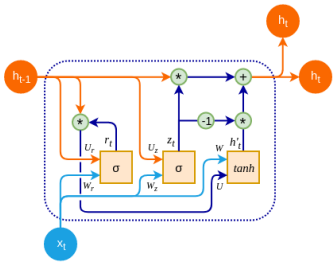

FIGURE 13. Architecture of GRU

A GRU cell has 4 components:

- Reset gate:

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

Where:

  - x_t is the current information.
  - h_{t-1} is the previous information.
  - W_r represents how much current information is weighted.
  - U_r represents how much previous information is weighted.

- Update gate:

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

Where:

  - x_t is the current information.
  - h_{t-1} is the previous information.
  - W_z represents how much current information is weighted.
  - U_z represents how much previous information is weighted.

- New memory content:

$$h'_t = tanh(W_h x_t + r_t.U_h h_{t-1})$$

Where:

  - x_t is the current information.
  - h_{t-1} is the previous information.
  - r_t represents how much previous information should be forgotten.
  - W_h represents how much current information is weighted.
  - U_h represents how much previous information is weighted.

- Final memory content:

$$h_t = z_t.h'_t + (1 - z_t).h_{t-1}$$

Where:

  - z_t represents how much the unit updates its information with the current information.
  - h_{t-1} represents the previous information.

– h'_t represents the current information with some degree of dependency on the previous information.

H. NEURAL BASIS EXPANSION ANALYSIS TIME SERIES (NBEATS)

Neural Basis Expansion Analysis for Time Series [**?**] (NBEATS) is a deep learning architecture tailored for forecasting time series data. It diverges from traditional methods that depend on statistical techniques or basic machine learning models by employing deep neural networks to directly learn intricate temporal patterns from the data.

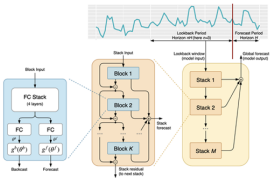

FIGURE 14. Architecture of NBEATS

1) Basic building blocks

These blocks have two fully connected layers in a fork architecture after a stack of fully connected layers. Once the initial stack of layers has processed the input data, the subsequent forecast and backcast layers come into play. These layers use the forward and backward expansion coefficients generated by the initial fully connected network to create the final forecast and backcast outputs.

The forecast layers focus on generating accurate predictions for future data points, while the backcast layers work on estimating the input values themselves, given the constraints on the functional space that the network can use to approximate signals.

2) Stacking blocks

These basic building blocks are stacked together using a technique known as “doubly residual stacking”. This method creates a hierarchical structure with residual connections across different layers, allowing for better interpretability and a more transparent network structure. Each block receives an input and generates a forecast (forward prediction) and a backcast (backward estimation).he backcast output of the previous block is subtracted from its input to create the input for the next block. This residual branch plays a key role in how the network processes the input signal. By handling the input signal step-by-step, it ensures that each block focuses on a specific part of the data, making the overall prediction more accurate and efficient. The partial forecasts, created by individual blocks, each capture different patterns and components of the input data. When move through the hierarchical structure, these partial forecasts are combined to form the final output. This creates a hierarchical decomposition of the forecasting process, where forecasts from the basic building blocks are combined to form the overall prediction.

V. RESULT

A. EVALUATION METHODS

Mean Absolute Percentage Error (MAPE) represents the average of the absolute percentage errors of each entry in a dataset to calculate how accurate the forecasted quantities were in comparison with the actual quantities.

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Where:

- n is the number of forecast values.
- A_t is the actual value of the time series at time t.
- F_t is the forecast value at time t.

Root Mean Squared Error (RMSE) measures the average difference between a statistical model’s predicted values and the actual values. It quantifies how dispersed these residuals are, revealing how tightly the observed data clusters around the predicted values.

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y_i} - y_i)^2}{n}}$$

Where:

- $\hat{y_1}, \hat{y_2}, ..., \hat{y_n}$ are predicted values.
- $y_1, y_2, ..., y_n$ are observed values.
- n is the number of observations.

Mean Absolute Error (MAE) measures the average of the magnitudes of the errors between the predicted and actual values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$$

Where:

- n is the number of observations.
- y_i is the actual value of the i^{th} observation.
- $\hat{y_i}$ is the predicted value of the i^{th} observation.

B. BTC/USD DATASET

BTC/USD Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MAE
LR	7:3	21064.65	74.50	19521.81
	8:2	**13640.22**	**29.98**	**11686.03**
	9:1	19141.51	25.41	15704.47
ARIMA	7:3	22094.05	36.85	16069.35
	8:2	**21759.27**	**28.62**	**15439.40**
	9:1	21255.13	28.96	17740.84
RNN	7:3	1498.40	3.38	1205.54
	8:2	**1468.04**	**1.89**	**953.16**
	9:1	2215.84	2.49	1656.83
GRU	**7:3**	**1172.71**	**1.89**	**761.59**
	8:2	1690.30	2.36	1186.07
	9:1	2846.96	3.48	2327.89
LSTM	7:3	2375.04	4.42	1838.50
	8:2	**1652.29**	**2.13**	**1098.40**
	9:1	2504.09	2.95	1967.71
SARIMAX	7:3	18939.08	29.80	13272.13
	8:2	**18183.71**	**23.68**	**12728.52**
	9:1	19199.60	25.88	15909.43
NBEATS	**7:3**	**1359.61**	**2.61**	**954.51**
	8:2	1417.10	2.26	989.33
	9:1	1800.29	2.37	1337.98
RF	**7:3**	**1890.12**	**3.18**	**1211.36**
	8:2	2271.28	2.77	1390.68
	9:1	3063.31	3.51	2180.00

TABLE 2. BTC/USD Dataset's Evaluation

FIGURE 15. LR model's result with 8:2 splitting proportion

FIGURE 16. ARIMA model's result with 8:2 splitting proportion

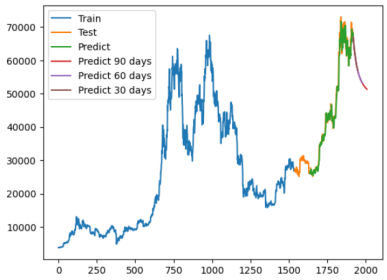

FIGURE 17. RNN model's result with 8:2 splitting proportion

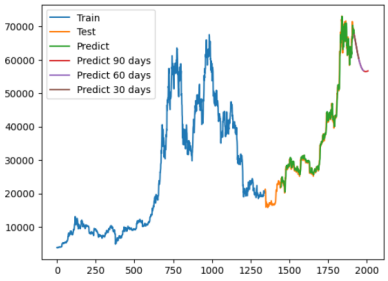

FIGURE 18. GRU model's result with 7:3 splitting proportion

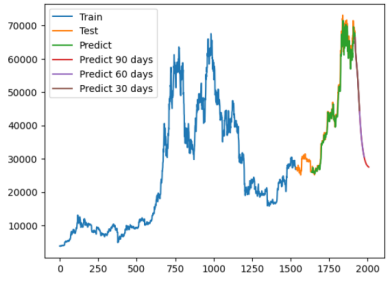

FIGURE 19. LSTM model's result with 8:2 splitting proportion

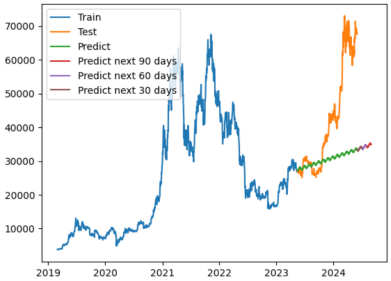

FIGURE 20. SARIMAX model's result with 8:2 splitting proportion

FIGURE 21. NBEATS model's result with 7:3 splitting proportion

FIGURE 22. RF model's result with 7:3 splitting proportion

VI. CONCLUSION

A. SUMMARY

In the endeavor to forecast cryptocurrency prices, various methodologies have been explored, ranging from traditional statistical models to advanced machine learning algorithms. Among the models tested, including Linear Regression (LR), Autoregressive Integrated Moving Average (ARIMA), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Seasonal Autoregressive Integrated Moving Average with eXogenous factors (SARIMAX), Neural Basis Expansion Analysis Time Series (NBEATS), and Random Forest (RF), it is evident that SARIMAX, NBEATS, and RF are the most promising and effective for predicting cryptocurrency prices.

The complexities of cryptocurrency price forecasting, driven by the volatility and unpredictability of financial markets, require models that can detect subtle patterns and relationships in the data. SARIMAX excels in managing complex relationships, offering robust predictions. NBEATS, with its capacity to capture sequential dependencies, shows remarkable performance in forecasting cryptocurrency prices. The implementation of ensemble learning through Random Forest (RF) enhances predictive capabilities, providing collective insights that surpass those of individual models.

Evaluation metrics, such as RMSE, MAPE, and MAE, reveal that SARIMAX, NBEATS, and RF consistently achieve superior performance in various aspects of forecasting accuracy. Their ability to handle the inherent uncertainties of cryptocurrency markets makes them powerful tools for investors and analysts seeking reliable predictions.

B. FUTURE CONSIDERATIONS

In our future research, it is essential to focus on further optimizing the models discussed earlier. This optimization should specifically target:

• Improving model accuracy. Although the aforementioned algorithms have shown promising results in predicting stock prices, there is a need to enhance the model's accuracy to achieve more precise forecasting outcomes.

• Investigating alternative machine learning algorithms or ensemble techniques. Utilizing ensemble techniques, such as combining multiple models or employing various ensemble learning methods, can enhance the robustness and accuracy of the forecasts.

• Exploring new forecasting models. The field of forecasting is continually evolving, with new algorithms and models being developed. It is important to stay informed about these advancements and investigate new forecasting models that may offer better accuracy and performance.

By continuously exploring and integrating new features, data sources, and modeling techniques, we can aim for ongoing optimization of the forecasting models, thereby improving their ability to predict stock prices with greater precision and reliability.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to **Assoc. Prof. Dr. Nguyen Dinh Thuan** and **Mr. Nguyen Minh Nhut** for their invaluable guidance, support, and expertise throughout the research process of "Using Forecasting Models in Cryptocurrency Price Prediction." Their insightful feedback and unwavering encouragement have been instrumental in shaping this study.

Assoc. Prof. Dr. Nguyen Dinh Thuan's profound have provided invaluable insights into the intricacies of cryptocurrency markets and forecasting methodologies. His mentorship has been truly invaluable.

We are also deeply thankful to **Mr. Nguyen Minh Nhut** for his technical assistance, dedication, and relentless pursuit of excellence. His expertise has significantly enhanced the quality and rigor of our research.

This research endeavor stands as a testament to the indispensable guidance and contributions of our esteemed mentors. We extend our sincere gratitude to all involved for their invaluable assistance, unwavering encouragement, and steadfast belief in the significance of our research.

REFERENCES

[1] M. Iqbal, M. S. Iqbal, F. H. Jaskani, K. Iqbal, and A. Hassan, "Time-Series Prediction of Cryptocurrency Market using Machine Learning Techniques," 07 Jul. 2021. [Online]. Available: https://eudl.eu/pdf/10.4108/eai.7-7-2021.170286

[2] H. N. Gudavalli and K. V. R. Kancherla, "Predicting Cryptocurrency Prices with Machine Learning Algorithms: A Comparative Analysis," Jun. 2023. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1778251/FULLTEXT03.pdf

[3] J. Zhu, "Bitcoin Price Prediction: ARIMA & SARIMA vs Linear Regression," Dec. 2023. [Online]. Available: https://www.researchgate.net/publication/376887838_Bitcoin_Price_Prediction_ARIMA_SARIMA_vs_Linear_Regression

[4] N. H. Gawali, "Bitcoin Price Prediction using Neural Network," 2021. [Online]. Available: https://norma.ncirl.ie/6125/1/namratagawali.pdf

[5] A. Bulatov, "Forecasting Bitcoin Prices Using N-BEATS Deep Learning Architecture," 12 Dec. 2020. [Online]. Available:

https://source.sheridancollege.ca/cgi/viewcontent.cgi?article=1008&
context=fast_sw_mobile_computing_theses