The Relationship between Bitcoin, Nasdaq and U.S. Dollar Index

HO QUANG LAM[1], LE THI LE TRUC[2], AND NGUYEN THANH DAT[3]
[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21521049@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21521586@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21521938@gm.uit.edu.vn)

ABSTRACT This paper investigates the long-run interaction between Bitcoin, Nasdaq and U.S. Dollar Index by applying weekly data from a January 3, 2013 until March 1, 2024.This study uses Linear Regression (LR), Long Short-Term Memory (LSTM), Autoregressive Integrated Moving Average (ARIMA), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Gauss Newton Method Non-Linear (GNM), Random Forest, RESCNN methods to examine the long-run association between the variables.

INDEX TERMS Keywords: Bitcoin, Nasdaq, U.S. Dollar Index

I. INTRODUCTION

The development of money has been influenced by the evolving demands of human cultures and technological advancements. Over time, paper cash emerged to address the requirements of growing economies, and the transition from physical goods to plastic cards promoted faster transactions. With the advent of the electronic era, electronic cash systems were developed, enabling seamless and rapid transactions. However, the decentralized nature of Bitcoin, based on blockchain technology, is currently challenging well-established financial institutions. In recent years, the number of cryptocurrencies has exponentially increased, with Bitcoin being the dominant player. Understanding the relationship between Bitcoin and traditional financial indicators, such as the U.S. Dollar Index and the Nasdaq stock market index, is crucial for grasping its valuation and integration into the global financial system.

In the course of this research, we use Linear Regression (LR), Long Short-Term Memory (LSTM), Autoregressive Integrated Moving Average (ARIMA), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Gauss Newton Method Non-Linear (GNM), Random Forest, RESCNN methods to examine the long-run association between the variables.

II. RELATED WORKS

Dwyer (2015) [1] delivers an influential paper demonstrating that BTC has higher average monthly volatility than gold or a group of international currencies. Urquhart (2016) [2], Nadarajah and Chu (2017) [3], and Bariviera (2017) [4] all corroborate this result by demonstrating BTC's inefficient returns.

Several studies have explored the relationship between Bitcoin, gold, and traditional currencies, such as the US dollar. Dyhrberg (2016) [5] suggests that Bitcoin can be a useful tool for risk management, especially for risk-averse investors who anticipate negative market shocks. Baur et al. (2018) [6] argue that Bitcoin exhibits different return, volatility, and correlation characteristics compared to gold and the US dollar, indicating its unique nature as an asset.

Dirican and Canoz (2017) [7] employed the ARDL boundary test approach to find a cointegration relationship between Bitcoin prices and the NASDAQ index, revealing hidden links underneath apparent discrepancies.

Studies have also examined the relationship between Bitcoin and equity markets, particularly during periods of uncertainty. The COVID-19 pandemic has acted as a catalyst for further research in this area. Quantile regression analysis conducted by Nguyen (2022) [8] revealed that during periods of high uncertainty, such as the COVID-19 crisis, the returns of the S&P 500 had a significant impact on Bitcoin returns. Additionally, stock market shocks had an effect on Bitcoin volatility during these years. This indicates that during times of heightened uncertainty, there is a stronger connection between the stock market and Bitcoin.

Several significant findings have emerged from studies examining the relationship between Bitcoin and the stock market. Wang et al. (2019) [9] found that the S&P 500 and Dow Jones indexes have a positive influence on Bitcoin, suggesting a favorable association between the cryptocurrency and the stock market. Maghyereh and Abdoh (2021) [10] discovered that Bitcoin and the US stock market exhibit positive co-movement at specific frequencies and time periods, indicating a potential interdependence between the two.

Additionally, Bouri et al. (2022) [11] demonstrated that the co-movement between US equities and Bitcoin changes over time and frequency, highlighting the dynamic nature of their interaction.

III. MATERIALS

A. DATASET

We get data on cryptocurrency prices from the Investing.com website with three datasets contains historical price data for three popular cryptocurrencies: Bitcoin, Nasdaq, U.S. Dollar Index and covers the time period from January 03, 2013 to June 1, 2024. Each dataset consists of 2023 rows and 7 columns include Date, Price, Open, High, Low, Vol., Change

Date: This column represents the date of the recorded data point. It provides the chronological information for each observation in the dataset.

Price: This column represents the closing price of the asset or security being analyzed (e.g., Bitcoin, stock, commodity) on a specific date. It indicates the value of the asset at the end of the trading day.

Open: This column represents the opening price of the asset on a specific date. It indicates the value of the asset at the beginning of the trading day.

High: This column represents the highest price reached by the asset during the trading day on a specific date. It provides insight into the peak value of the asset during that period.

Low: This column represents the lowest price reached by the asset during the trading day on a specific date. It provides insight into the lowest value of the asset during that period.

Vol. (Volume): This column represents the trading volume of the asset on a specific date. It indicates the total number of shares, contracts, or units of the asset that were traded during the trading day.

Change: This column represents the percentage change in the price of the asset compared to the previous trading day's closing price. It indicates the percentage increase or decrease in value between consecutive trading days.

B. DESCRIPTIVE STATISTICS

TABLE 1. US DOLLAR, NASDAQ, BITCOIN's Descriptive Statistics

	US DOLLAR	NASDAQ	BITCOIN
Mean	94.77986986	8291.193794	13062.80386
Standard Error	0.133999242	71.12929725	255.8327202
Median	95.754	7408.085	6636
Mode	104.182	13721.03	238.9
Standard Deviation	7.144811411	3743.591069	16218.66456
Sample Variance	51.0483301	14014474.09	263045080
Kurtosis	-0.037847286	-1.122949701	0.696153059
Skewness	-0.421644463	0.501032222	1.323476527
Range	34.921	13108.58	67493.6
Minimum	79.126	3166.36	34.3
Maximum	114.047	16274.94	67527.9
Sum	269459.17	22966606.81	52499408.7
Count	2843	2770	4019

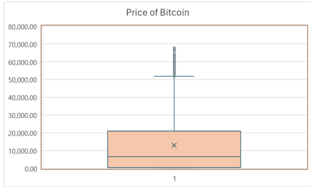

FIGURE 1. Bitcoin stock price's boxplot

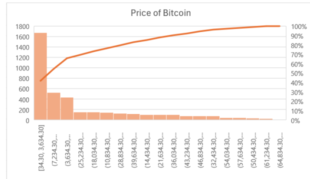

FIGURE 2. Bitcoin stock price's histogram

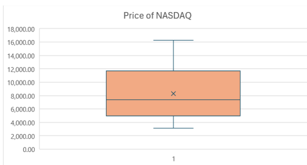

FIGURE 3. NASDAQ stock price's boxplot

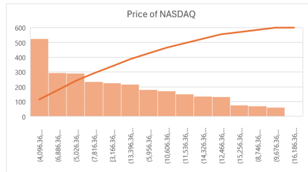

FIGURE 4. NASDAQ stock price's histogram

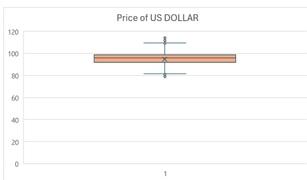

FIGURE 5. US DOLLAR stock price's boxplot

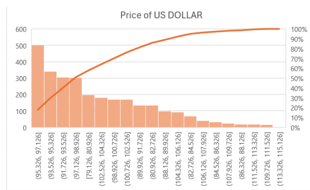

FIGURE 6. US DOLLAR stock price's histogram

IV. METHODOLOGY

A. LINEAR REGRESSION

Simple linear regression [12] estimates how much Y will change when X changes by a certain amount. With the correlation coefficient, the variables X and Y are inter- changeable. With regression, we are trying to predict the Y variable from X using a linear relationship (i.e., a line):

A simple linear regression model has the form:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Where:

- Y is the dependent variable
- X_1 is the independent variable.
- β_0 is the intercept term.
- β_1 is the regression coefficient for the independent variable.
- ε is the error term.

When there are multiple predictors, the equation is simply extended to accommodate them: A multiple linear regression model has the form:

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

Where:

- Y is the dependent variable
- $X_1, X_2, \ldots, X_k$ are the independent 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. ARIMA

ARIMA [13] stands for AutoRegressive (AR) Integrated (I) Moving Average (MA) and represents a cornerstone in time series forecasting. It is a statistical method that has gained immense popularity due to its efficacy in handling various standard temporal structures present in time series data.

AR(p): Autoregression - is the process of finding the relationship between current data and p previous data (lag)

$$Y = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \cdots + \beta_k X_{t-k} + \varepsilon_t$$

Where:
• Y is current observed value.
• $X_{t-1}, X_{t-2}, \ldots, X_{t-k}$ are past observed values.
• β_0 is the intercept term.
• $\beta_1, ..., \beta_k$ are regression analysis parameters.
• ε_t random forecasting error of the current period. The expected mean value is 0.

I(d): Integrated - Compare the difference between d observations (difference between the current value and d previous values)
• First Difference I(1): z(t) = y(t) - y(t - 1)
• Second Difference I(2): h(t) = z(t) - z(t - 1)

MA(q): Moving Average: is the process of finding a relationship between current data and q past errors

$$y_t = \beta_0 + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \ldots + \beta_q \varepsilon_{t-q}$$

Where:
• y(t) is current observed value.
• $\varepsilon_(t-1), \varepsilon_(t-2), \ldots, \varepsilon_(t-k)$ are forecast error.
• β_0 is the intercept term.
• $\beta_1, ..., \beta_k$ mean values of y(t) and moving average coefficients.
• $\varepsilon_(t)$ random forecasting error of the current period. The expected mean value is 0.
• q is the number of past errors used in the moving average.

C. RNN

Recurrent Neural Networks (RNNs) [14] is one of the neural networks of the network for professional data series processing.

A recurrent neural network (RNN) is essentially designed to handle sequential data by utilizing the concept of recurrent neural network architecture where information flows only in one direction from input to output. It has a type of feedback loop method that allows for information to be passed from one step of the network to the next to maintain a kind of memory usage. An RNN has sequence input data with unspecified size. This structure is represented and expressed in Figure 7

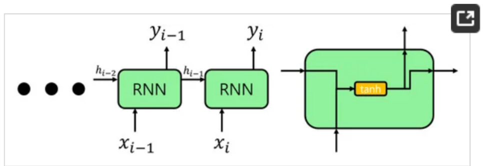

FIGURE 7. Recurrent neural network (RNN)

Fomula:

$$h_t = tanh(W_{xh}xt + W_{hh}x_{(t-1)}) + b_h$$

$$y_{xh} = W_{hy}h_t + b_y$$

Where:
• h_t: Hidden status at the present time t
• W_x: Input mapping weight matrix h_x enter hidden state
• h_x: Input at the current time
• W_h: The weight matrix maps the previous hidden state h_{t-1} enter a new hidden state h_t
• h_{t-1}: Hidden status at the previous time $(t-1)$

D. GRU

The Gated Recurrent Unit (GRU) is an improved version of RNN.

GRU uses a reset gate and an update gate to solve the vanishing gradient problem. These gates decide what information to be sent to the output. They can keep the information from long back without diminishing it as the training continues. We can visualize the architecture of GRU below:

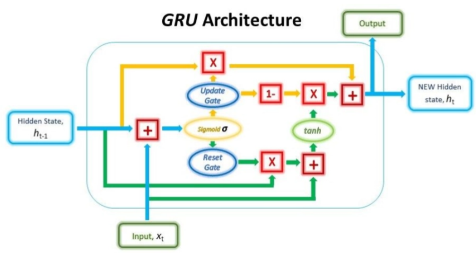

FIGURE 8. Gated Recurrent Unit (GRU)

The reset gate determines the information of the past that it needs to forget.

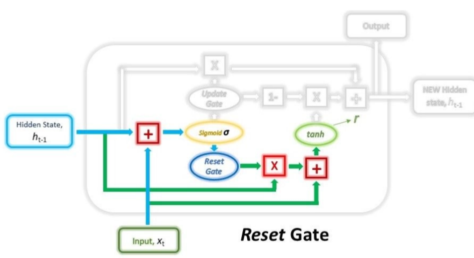

FIGURE 9. Reset Gate - Gated Recurrent Unit (GRU)

Update gate is responsible for long-term memory. It determines the amount of information on the previous steps that must be passed further.

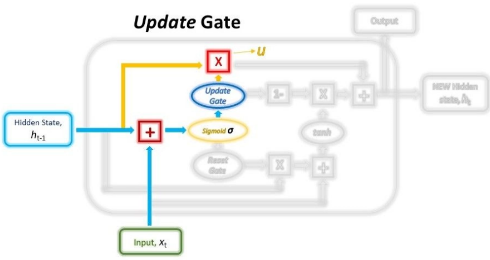

FIGURE 10. Update Gate - Gated Recurrent Unit (GRU)

This is how GRU solves the vanishing gradient problem. It keeps the relevant information and passes down the next step. It can perform excellently if trained correctly.

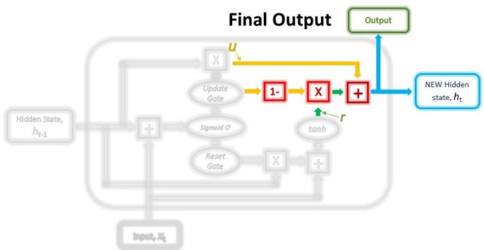

FIGURE 11. Final Output - Gated Recurrent Unit (GRU)

E. LSTM

Long Short-Term Memory is an improved version of recurrent neural network

A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network to learn long-term dependencies. LSTMs model addresses this problem by introducing a memory cell, which is a container that can hold information for an extended period.

LSTM architectures are capable of learning long-term dependencies in sequential data, which makes them well-suited for tasks such as language translation, speech recognition, and time series forecasting.

In the t state of the LSTM model:

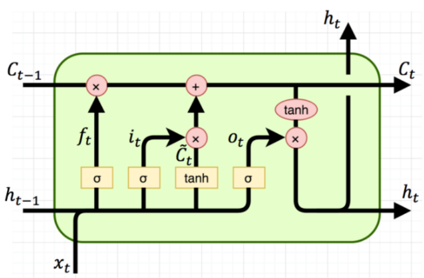

FIGURE 12. Long Short-Term Memory (LSTM)

Forget Gate: $f_t = (W_f[h_{t-1}, x_t] + b_f$
Input Gate: $i_t = (W_i[h_{t-1}, x_t] + b_i$
Output Gate: $o_t = (W_o[h_{t-1}, x_t] + b_o$
Temporary cell state:

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c$$

Current cell state: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$

F. GAUSS-NEWTON

The Gauss-Newton method is an iterative method for finding the experience of linear nonlinear optimization problems. It is based on the Jacobian objective function of the calculation matrix and uses this matrix to update the parameter vector in the direction of minimization.

This method has the advantage of converging faster than the classical gradient descent method, especially when the objective function is nearly square. However, it also has the disadvantage of having to calculate the Jacobian matrix, which can be computationally expensive, especially with complex functions.

Function focus:

$f(\mathbf{p}) = \mathbf{y}\text{train} - \mathbf{f}(\mathbf{p}, \mathbf{x}\text{train})$

Where:

- $\mathbf{p} = [a, b, c]$ is the parameter vector
- $\mathbf{x}_{\text{train}}$ is the input data
- $\mathbf{y}_{\text{train}}$ is the output data

Jacobian matrix:

$\mathbf{J}(\mathbf{p}, \mathbf{x}) = \begin{bmatrix} \exp(b\mathbf{x}) & a\mathbf{x}\exp(b\mathbf{x}) & 1 \end{bmatrix}$

G. RANDOM FOREST

The Random forest model will apply both ensemble learning and booster sampling methods. The order of the process of creating a forest model is as follows:

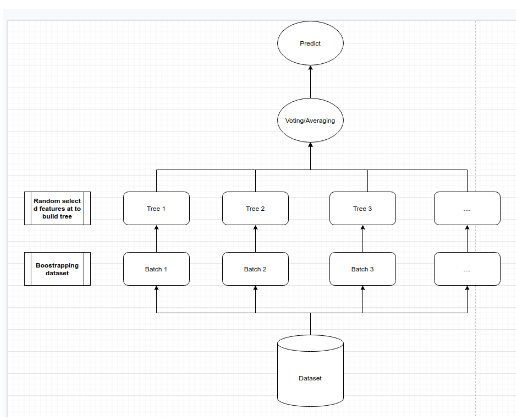

FIGURE 13. Random Forest

H. RESCNN

Residual convolutional neural network (Res-CNN) is an architectural convolutional neural network designed to solve problems that arise when the network is too deep. Different from traditional Convolutional Neural Networks (CNN), Res-CNN adds collaborative blocks (residual blocks) to enhance the ability to learn deep privileges.

Specifically, Res-CNN starts with the first layer to receive data images. Next, the network uses convolutional layers to extract special input image words. However, unlike CNN, Res-CNN uses block communities consisting of 2 or 3 fast active layers, accompanied by hopping connections (skip connections). This connection helps transmit information from the input directly to the output of the block, avoiding information loss when the network is too deep.

Next, the network uses pooling layers to reduce the feature map's size, which reduces the number of parameters to learn. Finally, the fully connected layers will be used to perform classification based on the extracted features. The training process of Res-CNN uses a backpropagation algorithm to update the network parameters.

By combining slow distribution layers with residual blocks, Res-CNN can better learn special depths, thereby improving the performance of distributed CNN systems.

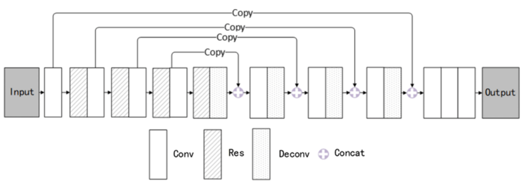

FIGURE 14. ResCNN

V. RESULT

A. EVALUATION METHODS

Mean Percentage Absolute Error (MAPE): measures the average percentage error between the predicted value and the actual value. The smaller the MAPE value, the better the model. MAPE is calculated by:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$$

Root Mean Squared Error (RMSE): measures the average error of the model compared to the actual data. The smaller the RMSE value, the better the image. RMSE is calculated by:

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

Mean Absolute Error (MAE): MAE calculates the average absolute value of the error. The smaller the MAE value, the better the model.

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

Where:

- n is the number of observations in the dataset.
- y_i is the true value.
- $\hat{y_i}$ is the predicted value.

B. BITCOIN DATASET

TABLE 2. BITCOIN Dataset's Evaluation

BITCOIN Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MAE
Linear Regression	6:4	25973.6	52.6	20231.9
	7:3	27883.78	55.9	23588.47
	8:2	**13118.3**	**35.37**	**10752.59**
ARIMA	6:4	29294.9	64.68	23753.45
	7:3	300699.74	848.48	259993.78
	8:2	**18089.92**	**63.35**	**16223.77**
GRU	6:4	1572.07	2.94	1039.9
	7:3	1876.75	3.95	1431.3
	8:2	**1042.3**	**2.01**	**661.66**
RNN	6:4	4017.1	7.38	2923.14
	7:3	**1357.61**	**2.35**	**894.73**
	8:2	1147:87	2.63	806.97
LSTM	6:4	3165.17	4.78	1956.52
	7:3	3566.02	5.67	2519.39
	8:2	**1047.35**	**3.11**	**1025.77**
GAUSS NEWTON	**6:4**	**17074.42**	**37.4**	**12552.76**
	7:3	19714.68	49.48	17183.64
	8:2	28057.44	67.41	23937.01
RF	6:4	23693.88	42.84	17965.59
	7:3	13048.55	31.84	10658.68
	8:2	**4201.88**	**13.76**	**3547.93**
RESCNN	6:4	0.14124	15.07	0.0925
	7:3	0.0873	9.05	0.0588
	8:2	**0.0195**	**2.4**	**0.0122**

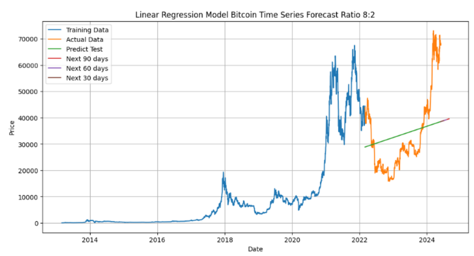

FIGURE 15. Linear model's result with 8:2 splitting proportion

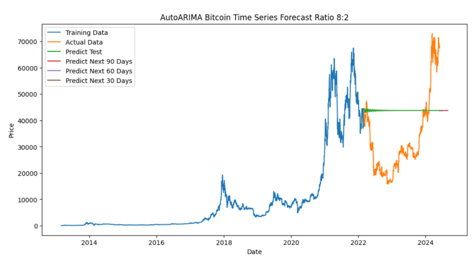

FIGURE 16. ARIMA model's result with 8:2 splitting proportion

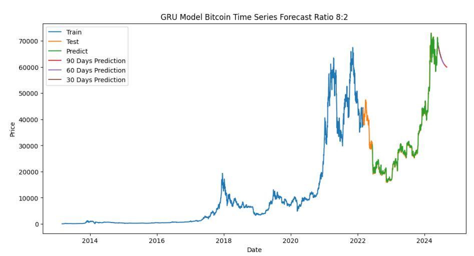

FIGURE 17. GRU model's result with 8:2 splitting proportion

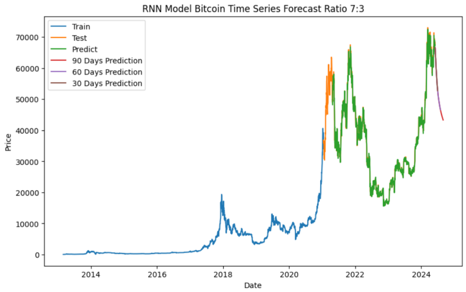

FIGURE 18. RNN model's result with 7:3 splitting proportion

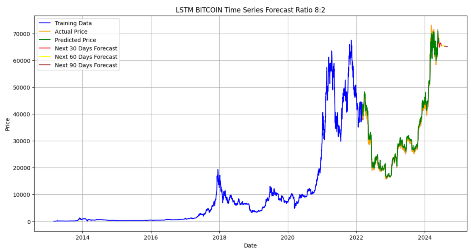

FIGURE 19. LSTM model's result with 8:2 splitting proportion

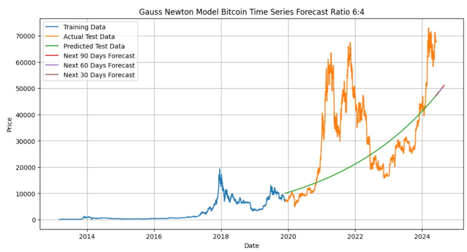

FIGURE 20. Gauss Newton model's result with 6:4 splitting proportion

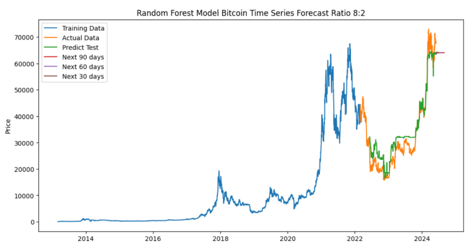

FIGURE 21. Random Forest model's result with 8:2 splitting proportion

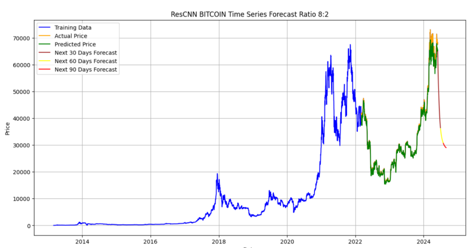

FIGURE 22. ResCNN model's result with 8:2 splitting proportion

C. NASDAQ DATASET

TABLE 3. NASDAQ Dataset's Evaluation

NASDAQ Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MAE
Linear Regression	6:4	3431.4	21.86	2961.64
	7:3	2835.49	16.4	2346.44
	8:2	**1575.34**	**11.3**	**1338.1**
ARIMA	6:4	2881.99	17.33	2369.8
	7:3	2771.89	18.31	2286.68
	8:2	**2200.68**	**15.62**	**1870.2**
GRU	6:4	190.66	1.18	151.48
	7:3	196.34	1.17	153.22
	8:2	**185.97**	**1.16**	**150.58**
RNN	6:4	481.46	1.95	405.96
	7:3	234.83	1.39	181.46
	8:2	**185.64**	**1.17**	**150.21**
LSTM	6:4	288.39	1.82	232.1
	7:3	728.29	4.89	674.76
	8:2	**264.13**	**1.65**	**264.13**
GAUSS NEWTON	**6:4**	**2236.79**	**13.35**	**1752.5**
	7:3	5481.8	35.86	4723.69
	8:2	10335.16	74.83	9487.95
RF	6:4	4918.66	33.55	4563.65
	7:3	1607.76	8.31	1220.17
	8:2	**379.65**	**2.12**	**276.25**
ResCNN	**6:4**	**0.0186**	**2.12**	**0.0145**
	7:3	0.0211	2.27	0.0175
	8:2	0.0215	2.63	0.0179

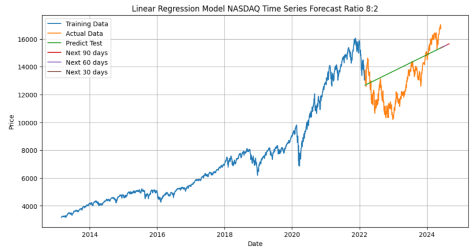

FIGURE 23. Linear model's result with 8:2 splitting proportion

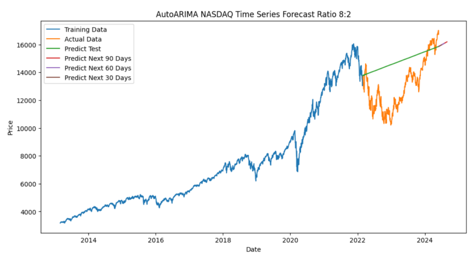

FIGURE 24. ARIMA model's result with 8:2 splitting proportion

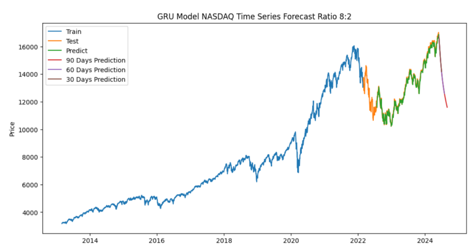

FIGURE 25. GRU model's result with 8:2 splitting proportion

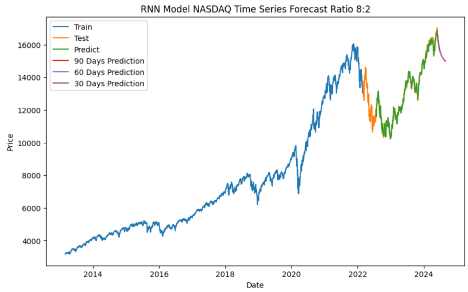

FIGURE 26. RNN model's result with 8:2 splitting proportion

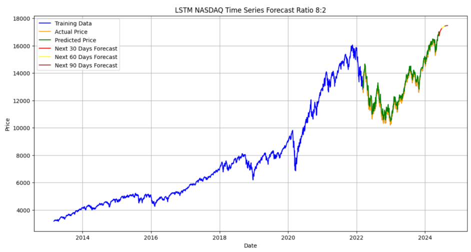

FIGURE 27. LSTM model's result with 8:2 splitting proportion

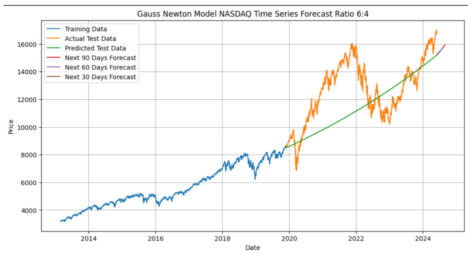

FIGURE 28. Gauss Newton model's result with 6:4 splitting proportion

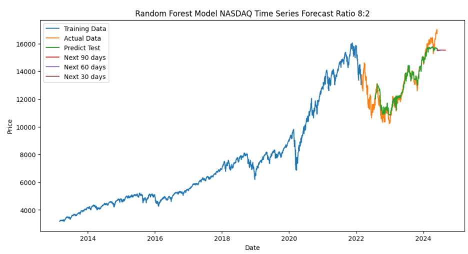

FIGURE 29. RF model's result with 8:2 splitting proportion

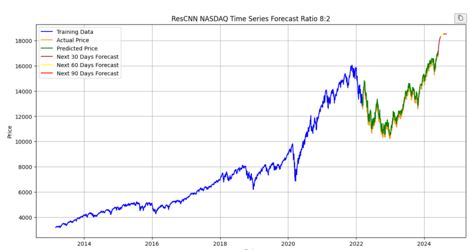

FIGURE 30. ResCNN model's result with 6:4 splitting proportion

D. USD DATASET

TABLE 4. USD Dataset's Evaluation

USD Dataset's Evaluation				
Model	Training:Testing	RMSE	MAPE (%)	MAE
Linear Regression	6:4	7.27	6.57	6.34
	7:3	**5.16**	**4.16**	**4.04**
	8:2	5.59	4.4	4.67
ARIMA	**6:4**	**5.67**	**4.86**	**4.68**
	7:3	11.49	9.54	9.9
	8:2	7.38	6.42	6.76
GRU	6:4	0.53	0.39	0.39
	7:3	0.63	0.48	0.49
	8:2	**0.55**	**0.38**	**0.41**
RNN	6:4	0.67	0.5	0.51
	7:3	0.75	0.54	0.56
	8:2	**0.64**	**0.45**	**0.48**
LSTM	6:4	0.58	0.44	0.57
	7:3	**0.52**	**0.39**	**0.38**
	8:2	0.53	0.38	0.4
GAUSS NEWTON	**6:4**	**7.28**	**6.58**	**6.34**
	7:3	9.79	7.95	8.27
	8:2	11.85	10.94	11.47
Random Forest	6:4	2.56	1.37	1.45
	7:3	**3.1**	**1.83**	**1.94**
	8:2	3.74	2.53	2.7
ResCNN	6:4	0.02	1.89	0.01
	7:3	**0.02**	**1.85**	**0.01**
	8:2	0.03	3.15	0.02

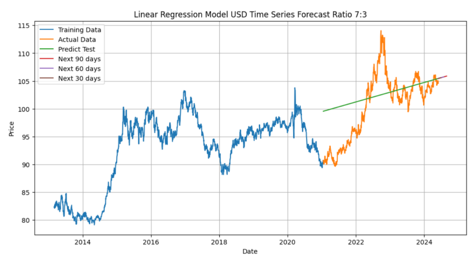

FIGURE 31. Linear model's result with 7:3 splitting proportion

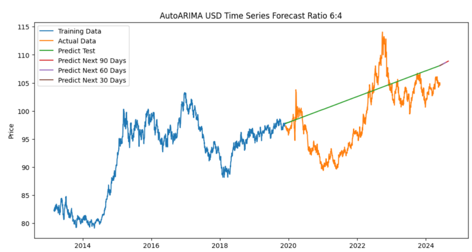

FIGURE 32. ARIMA model's result with 6:4 splitting proportion

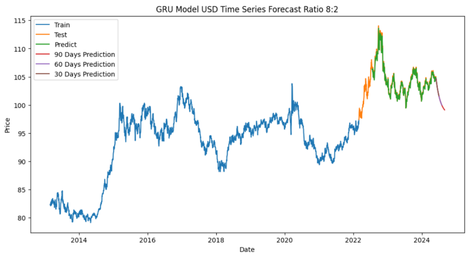

FIGURE 33. GRU model's result with 8:2 splitting proportion

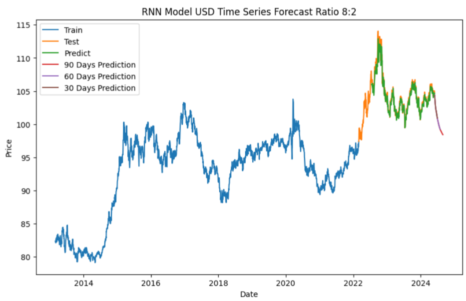

FIGURE 34. RNN model's result with 8:2 splitting proportion

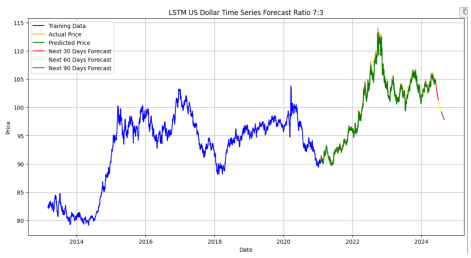

FIGURE 35. LSTM model's result with 7:3 splitting proportion

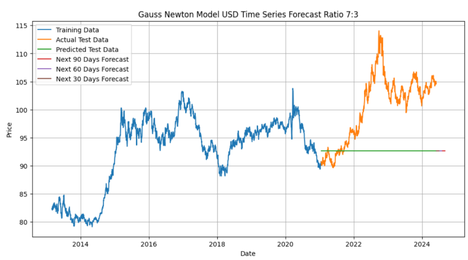

FIGURE 36. Gauss Newton model's result with 6:4 splitting proportion

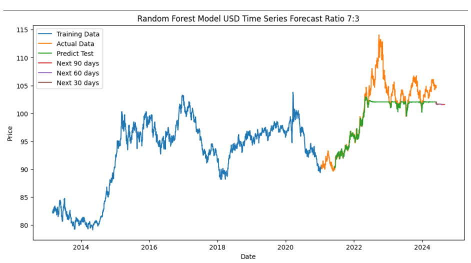

FIGURE 37. Random Forest model's result with 7:3 splitting proportion

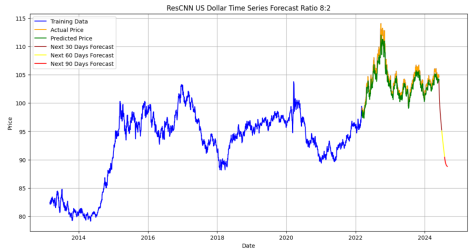

FIGURE 38. ResCNN model's result with 7:3 splitting proportion

VI. CONCLUSION

A. SUMMARY

In the bitcoin, nasdqa, usd price forecasting work, exploration of diverse methods, from traditional statistical models to advanced machine learning algorithms, has been targeted. Among the implemented models, Linear Regression (LR), Autoregressive Integrated Moving Average (ARIMA), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), LSTM, Gauss Newton Method Non -Linear (GNM), Random Forest, RESCNN are the most promising and effective models to predict the price of bitcoin, nasdqa, usd The practice of forecasting the price of bitcoin, nasdqa, usd, stems from the complexity and difficulty of major financial markets, requiring models that can capture subtle patterns and relationships in the data.

As can be shown through the evaluation criteria, including RMSE, MAPE, and MAE, the GNM model, Random Forest, and ResCNN were consistently able to perform superior performance at the expected exact edges. Their adaptability to deal with the inherent vagaries of the stock market positions them as reliable tools for investors and analysts looking for reliable predictions.

B. FUTURE CONSIDERATIONS

In our future research, it is crucial to prioritize further optimization of the previously mentioned models. This optimization effort should specifically focus on:

• Enhancing the accuracy of the model. While the above algorithms have demonstrated promising results in predicting stock prices, there is a need to further improve the model's accuracy to ensure more precise forecasting outcomes.

• Exploring alternative machine learning algorithms or ensemble techniques. Ensemble techniques, such as combining multiple models or using various ensemble learning methods, can also improve the robustness and accuracy of the forecasts.

• Researching new forecasting models. The field of forecasting continuously evolves, with new algorithms and models being researched and developed. It is crucial to stay updated with these approaches and explore new forecasting models that offer improved accuracy and performance.

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

ACKNOWLEDGMENT

First and foremost, we would like to express our sincere gratitude to **Assoc. Prof. Dr. Nguyen Dinh Thuan** and **Mr. Nguyen Minh Nhut** for their exceptional guidance, expertise, and invaluable feedback throughout the research process. Their mentorship and unwavering support have been instrumental in shaping the direction and quality of this study. Their profound knowledge, critical insights, and attention to detail have significantly contributed to the success of this research.

This research would not have been possible without the support and contributions of our mentors. We would like to extend our heartfelt thanks to everyone involved for their invaluable assistance, encouragement, and belief in our research. Thank you all for your invaluable assistance and encouragement.

REFERENCES

[1] Dwyer, G.P. (2015), The economics of Bitcoin and similar private digital currencies. Journal of Financial Stability, 17, 81-91.

[2] Urquhart, A. (2016), The inefficiency of Bitcoin. Economics Letters, 148, 80-82

[3] Nadarajah, S., Chu, J. (2017), On the inefficiency of Bitcoin. Economics Letters, 150, 6-9.

[4] Bariviera, A.F. (2017), The inefficiency of Bitcoin revisited: A dynamic approach. Economics Letters, 161, 1-4 Available: https://ieeexplore.ieee.org/document/7046047..

[5] Dyhrberg, A.H. (2016), Bitcoin, gold and the dollar-a GARCH volatility analysis. Finance Research Letters, 16, 85-92.

[6] Baur, D.G., Dimpfl, T., Kuck, K. (2018), Bitcoin, gold and the US dollar-a replication and extension. Finance Research Letters, 25, 103-110.

[7] Dirican, C., Canoz, I. (2017), The cointegration relationship between Bitcoin prices and major world stock indices: An analysis with ARDL model approach. Journal of Economics Finance and Accounting, 4(4), 377-392

[8] Nguyen, K.Q. (2022), The correlation between the stock market and Bitcoin during COVID-19 and other uncertainty periods. Finance Research Letters, 46, 102284.

[9] Shen, D., Urquhart, A., Wang, P. (2019), Does twitter predict Bitcoin? Economics Letters, 174, 118-122.

[10] Maghyereh, A., Abdoh, H. (2021), Time-frequency quantile dependence between Bitcoin and global equity markets. The North American Journal of Economics and Finance, 56, 101355

[11] Bouri, E., Kristoufek, L., Azoury, N. (2022), Bitcoin and SP500: Comovements of high-order moments in the time-frequency domain. PLoS One, 17(11), e0277924

[12] Peter Bruce, Andrew Bruce Peter Gedeck, "Practical Statistics for Data Scientists", 141-149

[13] Jason Brownlee (2023) , "How to Create an ARIMA Model for Time Series Forecasting in Python"

[14] Debasish Kalita (2024) , "A Brief Overview of Recurrent Neural Networks (RNN)"