COMPARATIVE ANALYSIS OF STATISTICAL AND MACHINE LEARNING MODELS FOR ENHANCED CRYPTOCURRENCY PRICE PREDICTION WITH TECHNICAL INDICATOR INTEGRATION

LE QUOC KHANH[1],LE GIA KIET[2], AND NGUYEN THI THUY[3]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21520283@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21522255@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 25122662@gm.uit.edu.vn)

ABSTRACT Cryptocurrency price prediction is a challenging yet crucial task in the dynamic realm of financial markets. In this article, we explore the efficacy of various statistical models and machine learning algorithms to forecast cryptocurrency prices. Leveraging a diverse toolkit including Linear Regression, ARIMAX (AutoRegressive Integrated Moving Average with Exogenous Variables), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM) networks, Vector Autoregression (VAR), XGBoost, and LightGBM, we aim to capture the complex patterns inherent in cryptocurrency price movements. Technical indicators serve as the primary features, offering insights into market sentiment and trends. Through rigorous evaluation and comparison of these models, we seek to discern their strengths and weaknesses in accurately predicting cryptocurrency prices, contributing to the advancement of predictive analytics in the volatile domain of digital assets.

INDEX TERMS *Cryptocurrency price, Forecasting, Technical Indicators, Linear regression, ARIMAX, RNN, GRU, LSTM, VAR, XGBoost, LightGBM*

I. INTRODUCTION

Cryptocurrency has become a significant and rapidly growing market, with thousands of different digital currencies available for trading. With the rise of cryptocurrency, there has been an increasing demand for accurate and reliable price prediction models. This paper aims to explore the use of statistical models and machine learning algorithms to predict the price of three popular cryptocurrencies: Bitcoin, Ethereum, and Dogecoin.

The price of cryptocurrencies is influenced by a variety of factors, including market demand, investor sentiment, and global economic conditions. These factors make predicting the price of cryptocurrencies a challenging task. However, with the use of statistical models and machine learning algorithms, it is possible to identify patterns and trends in the data that can be used to make more accurate predictions.

This paper scrutinizes a comprehensive suite of models, encompassing traditional statistical methods and advanced machine learning algorithms. Specifically, we explore the application of Linear Regression, ARIMAX (AutoRegressive Integrated Moving Average with Exogenous Variables), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM) networks, Vector Autoregression (VAR), XGBoost, and LightGBM. We will evaluate the performance of these models using historical price data and assess their ability to accurately predict future price movements.

Technical indicators provide valuable insights into market sentiment and trends, aiding in predicting price movements. Integrating these indicators into machine learning models offers a promising strategy to leverage the abundant cryptocurrency market data. This study employs various stock market evaluation indicators, such as moving averages and momentum indicators, to derive features for machine learning. The goal is to enhance predictive accuracy by enabling algorithms to identify meaningful patterns.

In summary, by using statistical models and machine learning algorithms to predict the price of cryptocurrencies, investors and traders can make more informed decisions and potentially increase their returns. Additionally, these models and algorithms can be used by cryptocurrency exchanges and financial institutions to manage risk and improve their trading strategies.

II. RELATED WORKS

There have been multiple studies conducted on the use of statistical models and machine learning algorithms for cryptocurrency price prediction. In a article by Gouxuan Son (2024) [1], two models that we concerned in three distinct models employed, Xboost and LightGBM, for predicting

Bitcoin prices was conducted. Another paper by Ziyang Yuan (2023) [2] used KNN, XGBoost and LightGBM to predict the price of Gold and Bitcoin Price. Especially, the investigation found that LightGBM is more effective and space-saving. Haydier, Albarwari and Ali compared between VAR and ARIMAX Time Series Models in Forecasting [3]. The results showed that the VAR model is better than the ARIMAX model for their observed data depending on the MSE criterion.

In another article, [4] the authors compared the performance of LSTM and GRU models in predicting Bitcoin prices. Additionally, they approved that the GRU model was able to capture long-term dependencies in the Bitcoin price data, while the LSTM model struggled to do so. Meanwhile, [5] compared and proved that LSTM is also better than RNN.

Recent research emphasizes the pivotal role of feature selection in developing effective and interpretable models for cryptocurrency price prediction. Huang, and Ni (2019) [6] showcased this significance by integrating high-dimensional technical indicators to predict bitcoin returns. Similarly, Mudassir et al. (2020) [7] employed a machine learning approach using such features for time-series forecasting of Bitcoin prices. These studies underscore the increasing acknowledgment of technical indicators as valuable features in enhancing predictive accuracy within cryptocurrency markets.

Based on insights gleaned from numerous prior literature studies, this research aims to predict cryptocurrency prices utilizing a diverse array of predictive models, including Linear Regression, ARIMAX, RNN, GRU, LSTM, VAR, XGBoost, and LightGBM.

III. MATERIALS

A. DATASET

The data is collected from finance.yahoo.com, downloading the daily data of Bitcoin, Ethereum and Dogecoin from 2018-Mar-01 to 2024-Jun-01, including close price, open price, high price, low price, Adjust close and the volume of trading coins with Currency in USD.

Including attributes

- Date: Represents the date of the trading day.
- Open: Refers to the opening price of Bitcoin/Ethereum/Dogecoin on that particular day.
- High: Indicates the highest price reached by Bitcoin/Ethereum/Dogecoin during that day.
- Low: Represents the lowest price reached by Bitcoin/Ethereum/Dogecoin during that day.
- Close: Refers to the closing price of Bitcoin/ETH/Dogecoin on that day.
- Adj Close: Represents the adjusted closing price, which accounts for factors like dividends and stock splits.
- Volume: Refers to the trading volume of Bitcoin on that day, i.e., the total number of Bitcoin/Ethereum/Dogecoin units traded.

As the goal is to forecast the price, only data relating to column “Close" (USD) will be analyzed

B. DESCRIPTIVE STATISTICS

	BTC	ETH	DOGE
Count	2285.000	2285.000	2285.000
Mean	24483.298	1383.339	0.071
Median	18251.604	1197.595	0.091
Mode	3236.762	84.308	0.002
Min	8601.796	233.028	0.003
25%	20041.738	1274.619	0.059
50%	37849.664	2088.574	0.089
75%	73083.500	4812.087	0.685
Max	20041.738	1274.619	0.059
Std	6741.750	3156.510	0.003
Variance	333121045.601	1434232.661	0.008
Kurtosis	69846.738	4727.779	0.683
Skewness	0.762	0.703	2.005
Range	-0.509	-0.528	5.603

TABLE 1. Descriptive Statistics of Cryptocurrencies

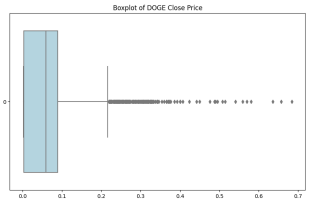

FIGURE 1. DOGE's Box Plot

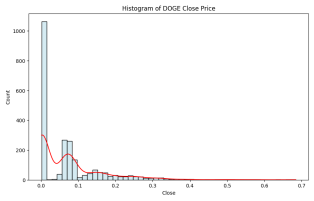

FIGURE 2. DOGE's Histogram

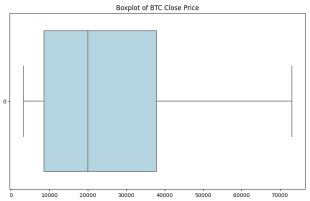

FIGURE 3. BTC's Box Plot

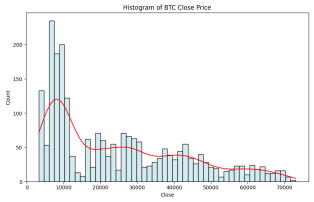

FIGURE 4. BTC's Histogram

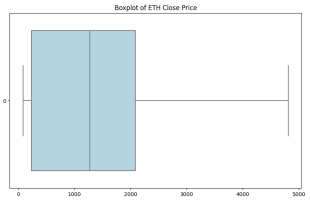

FIGURE 5. ETH's Box Plot

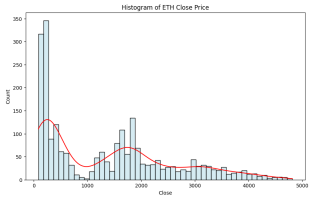

FIGURE 6. ETH's Histogram

- Across all three cryptocurrencies, there are remarkable differences in mean values, indicating diverse price levels. Moreover, the considerable range between minimum and maximum values highlights the wide

fluctuations in prices, portraying substantial volatility within the market.
- The high standard deviation, positive kurtosis, and skewness values suggest non-normal distributions with fat tails and right skewness. This indicates frequent occurrence of outliers and a tendency for prices to be skewed towards higher values, which means the occurrence of high prices isn't significant.
- These measures contributes to a summary that it is potential for high profits but also heightened risks when investing in these three cryptocurrencies.

IV. METHODOLOGY

A. LINEAR REGRESSION

In statistics, Linear Regression is a supervised learning algorithm that simulates a mathematical relationship between a dependent variable and independent variables, enabling predictions for continuous or numeric variables. 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 (Response variable).
- $X_1, X_2, \ldots, X_k$ 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. ARIMAX

ARIMAX is a time series forecasting model that combines Autoregressive Integrated Moving Average (ARIMA) model with exogenous variables. It extends ARIMA by including external predictors (denoted as X) to improve forecasting accuracy. This model involves specifying AR, I, and MA components, along with the exogenous variables, estimating model parameters, and making forecasts [3].

An ARIMAX model depicted by the following equation:

$$Y_t = \alpha + \underbrace{\beta_1 X_{1,t} + \ldots + \beta_r X_{r,t}}_{\text{exogenous variables}} + \underbrace{\phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p}}_{\text{AR term}} + \underbrace{\varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}}_{\text{MA term}}$$

C. VECTOR AUTOREGRESSION

Vector Autoregression (VAR) is a multivariate forecasting algorithm that is used when two or more time series influence each other.

It is considered as a generalization of univariate AR models or a combination between the two or more models and the univariate time series models. Each variable in a VAR is explained by its own lagged values and the lagged values of all the other variables in the equation.The basic VAR (p) model is given by:

$$Y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + ... + A_p y_{t-p} + \varepsilon_t$$

Where:
- $y_t - i$: the "ith lag" of y_t
- c is a k-vector of constants
- A_i: time-invariant (k × k)-matrix
- ε_t: a vector of error terms with k element

In matrix form:

$$Y_t = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_k \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ . \\ . \\ c_k \end{bmatrix} + \begin{bmatrix} \phi_{11} & . & . \\ \phi_{21} & . & . \\ . & & \\ . & & \\ \phi_{k1} & . & . \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ . \\ . \\ y_{k,t-1} \end{bmatrix} + ... + \begin{bmatrix} \phi_{11} & . & . \\ \phi_{21} & . & . \\ . & & \\ . & & \\ \phi_{k1} & . & . \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ . \\ . \\ y_{k,t-p} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ \varepsilon_k \end{bmatrix}$$

D. RECURRENT NEURAL NETWORKS (RNN)

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed for sequential data, where the connections between nodes form a directed graph along a temporal sequence. This allows RNNs to exhibit temporal dynamic behavior for a time sequence.

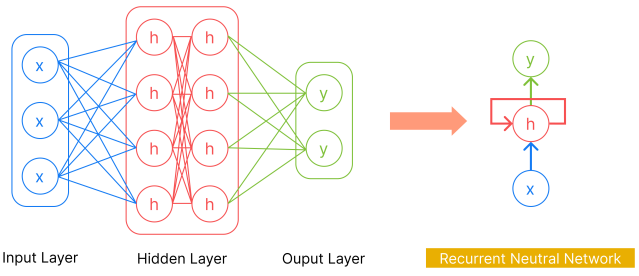

FIGURE 7. Model of RNN architectural

- **Input Layer**: Networks have only one input layer.
- **Hidden Layer**: Networks have multiple hidden layers.
- **Output Layer**: Networks have only one output layer.

A basic RNN can be represented as follows:

$$h_t = f(W_{xh} x_t + W_{hh} h_{t-1} + b_h)$$

$$y_t = g(W_{hy} h_t + b_y)$$

Where:

- x_t is the input at time step t.
- h_t is the hidden state at time step t.
- y_t is the output at time step t.
- W_{xh}, W_{hh}, and W_{hy} are weight matrices.
- b_h and b_y are bias vectors.
- f and g are activation functions.

E. LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) networks are a type of RNN designed to address the vanishing gradient problem that can occur with traditional RNNs when processing long sequences. LSTMs achieve this through a memory cell which is controlled by three gates:

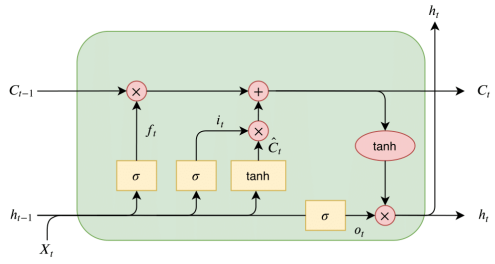

FIGURE 8. Model of LSTM architectural

- **Forget Gate**: Controls what information is discarded from the cell state.

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

- **Input Gate**: Controls what new information is added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

- **Output Gate**: Controls what information is output form the cell state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Where:

- W_f, W_i, W_C, W_o are the weight matrices of **f**orget gate, **i**nput gate, **C**ell state, and **o**utput gate respectively.
- b_f, b_i, b_C, b_o are the bias vectors of **f**orget gate, **i**nput gate, **C**ell state, and **o**utput gate respectively.
- σ is the sigmoid activation function.
- $\tanh$ is tanh activation function that gives an output of vector from -1 to +1.

Then the cell state C_t and hidden state h_t are then updated as follows:

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

$$h_t = o_t * \tanh(C_t)$$

F. GATED RECURRENT UNIT (GRU)

Gated Recurrent Units (GRUs) are another type of RNN with a simplified gating mechanism compared to LSTMs, which selectively update the hidden state at each time step. The GRU has only two gating mechanisms:

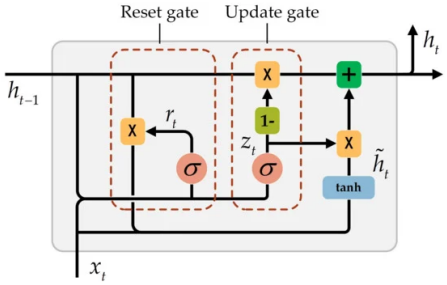

FIGURE 9. Model of GRU architectural

- **Update Gate**: Controls the amount of information from the previous hidden state to let through, and the amount of new information to add.

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

- **Reset Gate**: Controls the amount of information to forget from the previous hidden state.

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

The candidate hidden state is calculated based on the reset gate and the previous hidden state.

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b)$$

Then the hidden state is updated based on the update gate and the candidate hidden state.

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

Where:

- z_t and r_t are the update gate and reset gate.
- W_z, W_r, W, b_z, b_r, b are the weight matrices and bias vectors of the update gate, reset gate, and candidate hidden state respectively.
- σ is the sigmoid activation function.
- $\tanh$ is the hyperbolic tangent activation function.

G. XGBOOST

XGBoost (eXtreme Gradient Boosting) is a powerful machine learning algorithm that uses an ensemble of decision trees and gradient boosting, which combines multiple weak learners to create a strong predictive model, to minimize prediction errors.

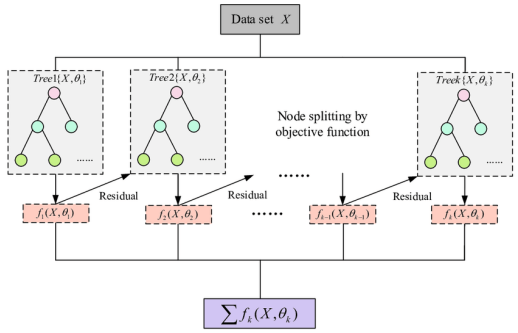

FIGURE 10. Model of XGBoost architectural

- **Loss Function**: Measures the difference between the true label y_i and the predicted label $\hat{y}_i$,

$$l(y_i, \hat{y}_i)$$

- **Regularization Term**: Penalizes model complexity to prevent overfitting in each the f tree.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

Where:

- T is the number of leaves in the f tree.
- w is the leaf weight of the f tree.
- λ, γ are the hyperparameters.

- **Objective Function** = Loss Function + Regularization Term

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \Omega(f_k)$$

Decision trees are built sequentially to minimize the objective function. At each step, a new tree f_t with input features x_i is added to correct the errors of the current model.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

- **Second-order Taylor expansion** is used to optimized the objective function based on gradient g and hessian h of the loss function.

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

Where:

- n is the number of data instances.
- $g_i = \frac{\partial l(y_i, \hat{y}_i)}{\partial \hat{y}_i}$ and $h_i = \frac{\partial^2 l(y_i, \hat{y}_i)}{\partial \hat{y}_i^2}$

And to find the optimal leaf weight w_j for each leaf node j, the objective function is minimized. Then we have the equation:

$$w_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

H. LIGHTGBM

LightGBM (Light Gradient Boosting Machine) is an advanced machine learning algorithm that builds upon the Gradient Boosting Decision Tree (GBDT) framework. It uses a histogram-based algorithm to speed up training and reduce memory usage.

LightGBM grows trees based on leaf-wise, while most other boosting tools (including XGBoost) are based on level-wise. Leaf-wise selects nodes to grow the tree based on optimizing the entire tree, while level-wise optimizes on the branch under consideration. This allows LightGBM to achieve lower loss than other boosting algorithms.

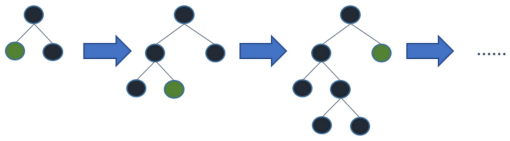

FIGURE 11. Leaf-wise tree expansion in LightGBM

Additionally, LightGBM enhances the efficiency and scalability of GBDT through the integration of two novel techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) [9].

- **Gradient-based One-Side Sampling (GOSS)**
  GOSS keeps all the instances with large gradients and performs random sampling on the instances with small gradients. In order to compensate for the influence on the data distribution, when computing the information gain, GOSS introduces a constant multiplier for the data instances with small gradients. Specifically, GOSS first sorts the data instances according to the absolute value of their gradients and selects the top $a \times 100\%$ instances. Then it randomly samples $b \times 100\%$ instances from the rest of the data. After that, GOSS amplifies the sampled data with small gradients by a constant $\frac{1-a}{b}$ when calculating the information gain.

- **Exclusive Feature Bundling (EFB)**
  High-dimensional data often have many features, but most of them are sparse, meaning they frequently take on zero values. Because many features are mutually exclusive (never nonzero at the same time), we can combine these features into Exclusive Feature Bundles. By a carefully designed feature scanning algorithm, we can build thesame feature histograms from the feature bundles as those from individual features. In this way, thecomplexity of histogram building changes from $O(\#data \times \#feature)$ to $O(\#data \times \#bundle)$, while #bundle « #feature.

However, LightGBM regression has disadvantages in generating the overfitting model as well as being sensitive the noise.

V. EXPERIMENT

A. FEATURE ENGINEERING

In this section, we focus on constructing diverse input features derived from technical indicators related to the Close price only. These features aim to enhance the predictive capabilities of our models and support continuous forecasting for out-of-sample (OOS) data. The selected indicators include the Moving Average Convergence/Divergence (MACD), Bollinger Bands (Upper and Lower), 26-period Moving Average (MA_{26}), and 20-period Exponential Moving Average (EMA_{20}).

MACD: The MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. It is calculated by subtracting the 26-period EMA from the 12-period EMA. The MACD helps identify changes in the strength, direction, momentum, and duration of a trend in a security's price.

$$MACD = EMA_{12} - EMA_{26}$$

Bollinger Band (BBand): BBands consist of a middle band (20-period moving average) and two outer bands (standard deviations away from the middle band). The upper and lower bands provide a relative definition of high and low prices. The upper BBand is calculated by adding two standard deviations to the 20-period moving average, and the lower BBand is calculated by subtracting two standard deviations from the 20-period moving average

$$UpperBand = MA_{20} + (2 \times \text{Standard Deviation})$$

$$LowerBand = MA_{20} - (2 \times \text{Standard Deviation})$$

MA_{26}: The MA_{26} is a simple moving average calculated by averaging the closing prices over the last 26 periods. It helps smooth out price data to identify the direction of the trend over a medium-term period.

$$MA_{26} = \frac{1}{26}\sum_{i=1}^{26} \text{Close}_i$$

EMA_{20}: The EMA_{20} is a type of moving average that places a greater weight and significance on the most recent data points. The EMA20 responds more quickly to recent price changes compared to the simple moving average. It is used to identify short-term trends.

$$EMA_{20} = (Close_t \times \frac{2}{1+20}) + (EMA_{20,t-1} \times (1 - \frac{2}{20+1}))$$

B. DATASET SPLIT RATIO

The dataset is divided into training and testing sets with three ratios of 70:30, 80:20, and 90:10. The training set is used to train the model, while the testing set is used to evaluate the model's performance.

To improve the dataset, we will use the following preprocessing techniques: data cleansing, feature selection.

C. EVALUATION METHODS

This paper uses three metrics to evaluate the performance of the models: Root Mean Squared Error (RMSE) Mean Absolute Percentage Error (MAPE) and Symmetric Mean Absolute Percentage Error (SMAPE)

With:

- y_i: Actual value
- $\hat{y}_i$: Predicted value
- n: Number of data points

- **Root Mean Squared Error (RMSE)**: Measures the average of the squared differences between the predicted and actual values.

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

(Best value = 0, Worst value = $+\infty$)

- **Mean Absolute Percentage Error (MAPE)**: Measures the average of the absolute percentage differences between the predicted and actual values.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

(Best value = 0, Worst value = $+\infty\%$)

- **Symmetric Mean Absolute Percentage Error (SMAPE)**: Measures the symmetric average of the absolute percentage differences between the predicted and actual values.

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \times 100$$

(Best value = 0, Worst value = 200%)

D. RESULTS

Model	Ratio	RMSE	MAPE	SMAPE
Linear Regression	7:3	1.486,62	2,88	2,91
	8:2	**1.684,85**	**2,66**	**2,69**
	9:1	2.171,14	2,89	2,91
ARIMAX	7:3	16.983,58	29,76	37,00
	8:2	**11.725,95**	**16,94**	**18,74**
	9:1	15.950,28	21,44	25,09
VAR	7:3	20.608,81	34,19	44,80
	8:2	**1438,54**	**22,80**	**24,58**
	9:1	18.135,05	23,62	28,40
RNN	7:3	1.640,65	3,27	3,33
	8:2	**1.514,22**	**2,17**	**2,18**
	9:1	2.009,36	2,62	2,66
LSTM	7:3	1.629,24	2,93	2,97
	8:2	1.584,69	3,27	3,21
	9:1	**1.782,90**	**2,46**	**2,45**
GRU	**7:3**	**1.142,67**	**1,96**	**1,95**
	8:2	1.363,21	2,53	2,50
	9:1	1.659,08	2,12	2,12
XGBoost	7:3	3.553,40	8,20	8,46
	8:2	3.630,37	6,43	6,70
	9:1	**4.502,40**	**6,30**	**6,30**
LightGBM	**7:3**	**3.172,67**	**5,88**	**5,92**
	8:2	4.251,13	8,97	9,52
	9:1	4.757,29	6,35	6,57

TABLE 2. BTC Dataset

Based on the result table above, we can see that the GRU model is the best-performing model among the 8 implemented models in predicting BTC stock prices with a 7:3 ratio, as it has the lowest RMSE, MAPE, and SMAPE values.

Model	Ratio	RMSE	MAPE	SMAPE
Linear Regression	7:3	101,12	3,28	3,31
	8:2	**111,42**	**3,01**	**3,03**
	9:1	147,02	3,66	3,68
ARIMAX	7:3	680,62	21,78	25,41
	8:2	**475,30**	**14,22**	**14,69**
	9:1	745,42	20,44	23,37
VAR	7:3	729,78	19,41	22,73
	8:2	**575,95**	**19,87**	**19,47**
	9:1	1.047,97	28,03	34,15
RNN	7:3	127,70	5,81	6,01
	8:2	170,59	7,43	7,13
	9:1	**151,67**	**4,62**	**4,77**
LSTM	7:3	79,93	2,56	2,52
	8:2	**81,12**	**2,13**	**2,16**
	9:1	108,02	2,71	2,71
GRU	7:3	102,98	4,27	4,13
	8:2	**92,33**	**3,02**	**2,97**
	9:1	109,55	2,70	2,74
XGBoost	7:3	143,88	5,54	5,35
	8:2	**144,28**	**3,98**	**3,95**
	9:1	175,47	4,58	4,61
LightGBM	7:3	130,99	5,13	4,97
	8:2	**137,65**	**4,23**	**4,21**
	9:1	183,24	5,10	5,12

TABLE 3. ETH Dataset

Based on the result table above, we can see that the LSTM model is the best-performing model among the 8 implemented models in predicting ETH stock prices with a 8:2 ratio, as it has the lowest RMSE, MAPE, and SMAPE values.

Model	Ratio	RMSE	MAPE	SMAPE
Linear Regression	7:3	0,01	4,62	4,69
	8:2	**0,01**	**3,99**	**4,04**
	9:1	0,01	5,13	5,23
ARIMAX	7:3	0,0524	31,25	41,65
	8:2	**0,0487**	**32,24**	**34,58**
	9:1	0,0757	39,69	56,11
VAR	**7:3**	**0,03**	**18,42**	**21,21**
	8:2	0,04	32,18	29,94
	9:1	0,06	29,87	38,82
RNN	7:3	0,0081	6,93	6,65
	8:2	0,0097	9,36	9,82
	9:1	**0,0111**	**6,35**	**6,38**
LSTM	7:3	0,01	11,12	11,88
	8:2	0,01	11,03	11,77
	9:1	**0,01**	**4,51**	**4,45**
GRU	**7:3**	**0,0067**	**5,50**	**5,64**
	8:2	0,0096	9,00	8,56
	9:1	0,0081	4,17	4,12
XGBoost	7:3	0,02	10,13	9,85
	8:2	**0,02**	**6,95**	**7,21**
	9:1	0,02	10,68	11,41
LightGBM	7:3	0,0134	9,26	9,01
	8:2	**0,02**	**8,81**	**8,03**
	9:1	0,0300	12,90	11,43

TABLE 4. DOGE Dataset

Based on the result table above, we can see that the GRU model is the best-performing model among the 8 implemented models in predicting DOGE stock prices with a 9:1 ratio, as it has the lowest RMSE, MAPE, and SMAPE values.

E. VISUALIZE

1) BTC Dataset

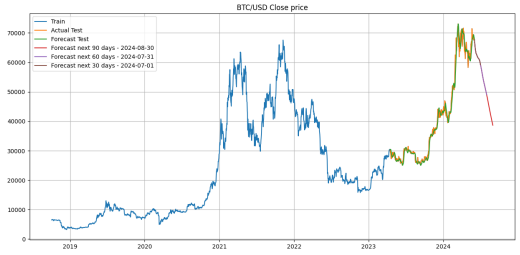

FIGURE 12. Linear Regression model's result with 8:2 ratio

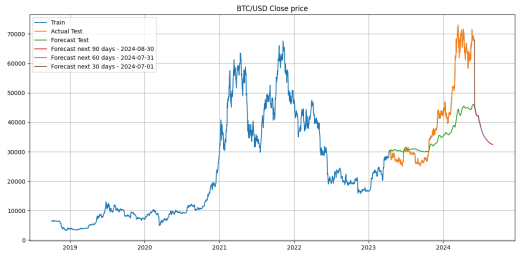

FIGURE 13. ARIMAX model's result with 8:2 ratio

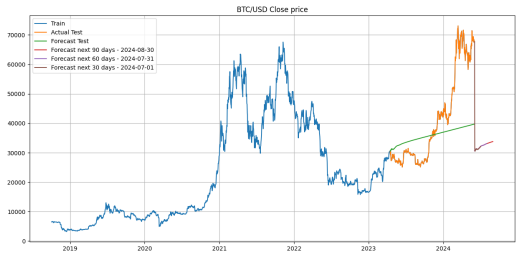

FIGURE 14. VAR model's result with 8:2 ratio

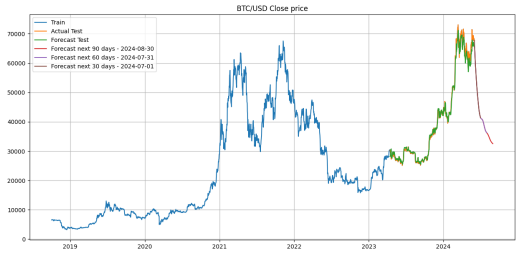

FIGURE 15. RNN model's result with 8:2 ratio

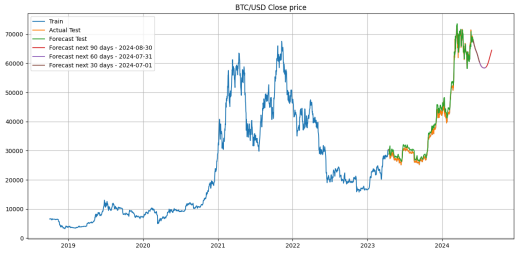

FIGURE 16. LSTM model's result with 8:2 ratio

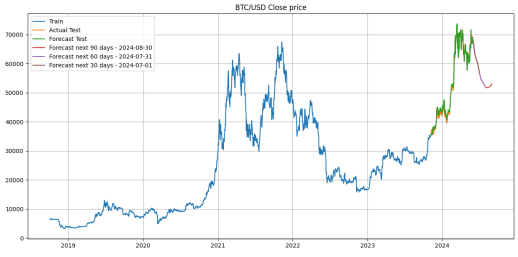

FIGURE 17. LSTM model's result with 9:1 ratio

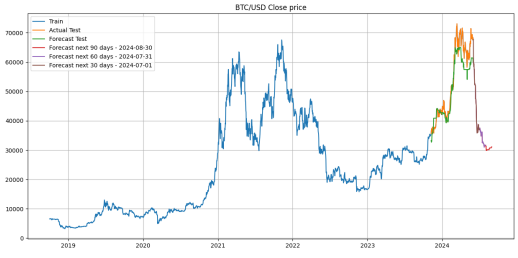

FIGURE 18. XGBoost model's result with 9:1 ratio

FIGURE 19. LightGBM model's result with 7:3 ratio

2) ETH Dataset

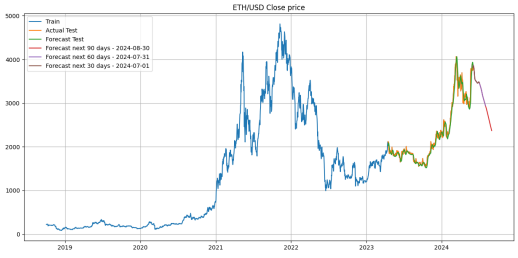

FIGURE 20. Linear Regression model's result with 8:2 ratio

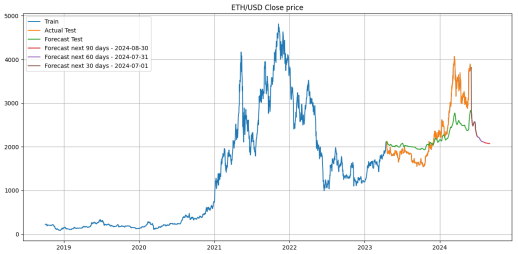

FIGURE 21. ARIMAX model's result with 8:2 ratio

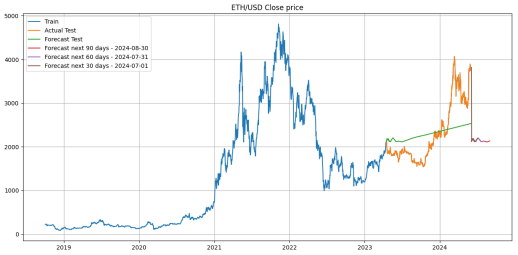

FIGURE 22. VAR model's result with 8:2 ratio

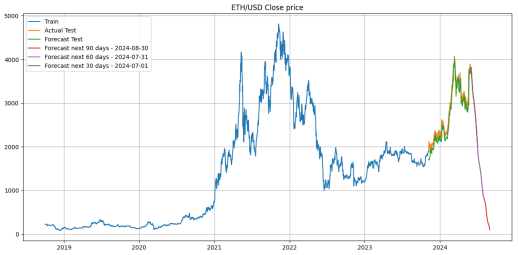

FIGURE 23. RNN model's result with 9:1 ratio

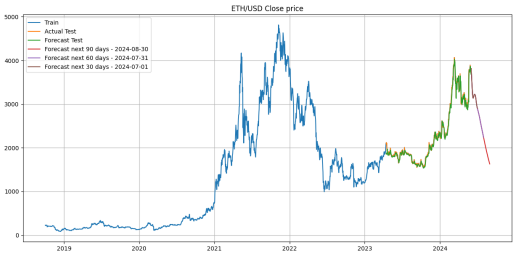

FIGURE 24. LSTM model's result with 8:2 ratio

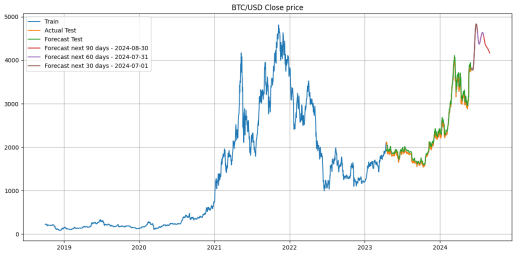

FIGURE 25. GRU model's result with 8:2 ratio

FIGURE 26. XGBoost model's result with 8:2 ratio

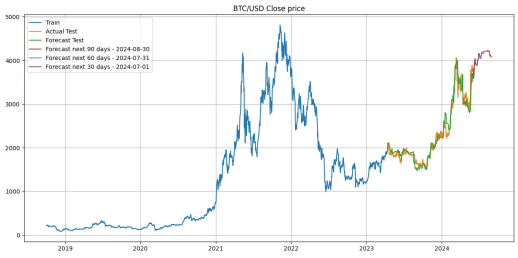

FIGURE 27. LightGBM model's result with 8:2 ratio

3) DOGE Dataset

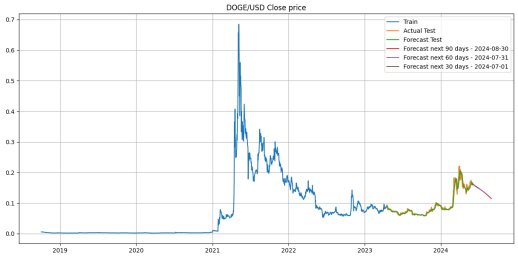

FIGURE 28. Linear Regression model's result with 8:2 ratio

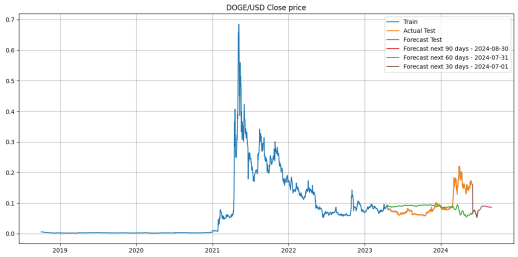

FIGURE 29. ARIMAX model's result with 8:2 ratio

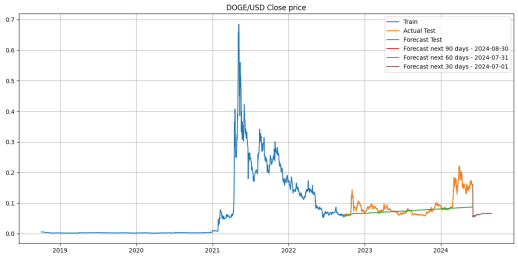

FIGURE 30. VAR model's result with 7:3 ratio

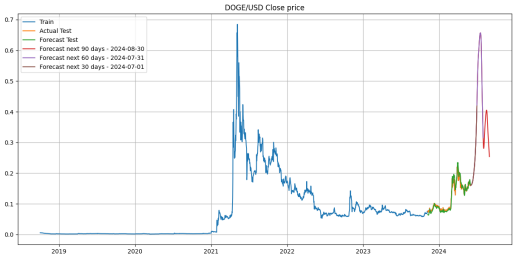

FIGURE 31. RNN model's result with 9:1 ratio

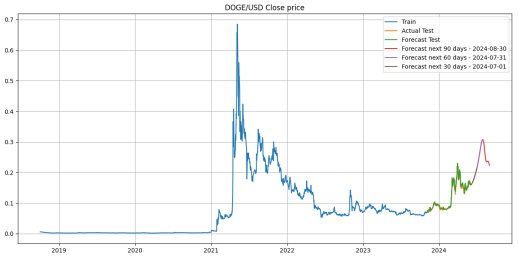

FIGURE 32. LSTM model's result with 9:1 ratio

FIGURE 33. GRU model's result with 7:3 ratio

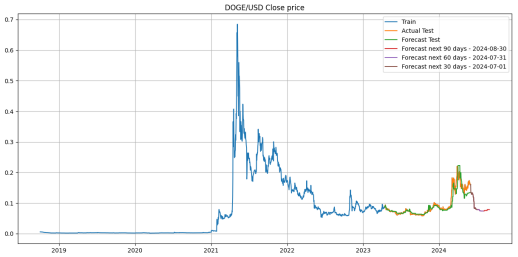

FIGURE 34. XGBoost model's result with 8:2 ratio

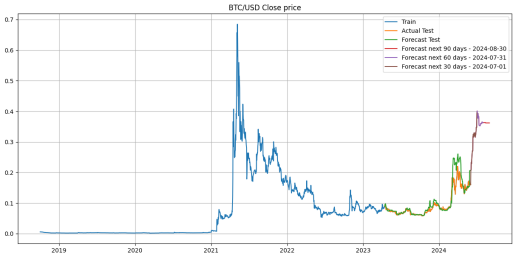

FIGURE 35. LightGBM model's result with 8:2 ratio

VI. CONCLUSION

A. SUMMARY

Machine learning models have emerged as a promising tool for accurately predicting and evaluating the value of cryptocurrencies, akin to their application in stock trading. In this study, we performed a detailed comparative analysis of various statistical and machine learning models to enhance cryptocurrency price prediction by integrating technical indicators. The models examined included Linear Regression, ARIMAX, RNN, GRU, LSTM networks, VAR, XGBoost, and LightGBM, using a dataset comprising Bitcoin (BTC), Ethereum (ETH), and Dogecoin (DOGE). Our evaluation metrics were RMSE, MAPE, and SMAPE, with GRU and LSTM models demonstrating exceptional performance in capturing the intricate patterns of cryptocurrency price movements.

However, the dataset used in this analysis is influenced by numerous factors such as market sentiment, regulatory news, macroeconomic trends, and technological advancements. These external factors can introduce noise and unpredictability, posing significant challenges for accurate price prediction. Consequently, as the best performing models among the eight models executed, GRU and LSTM may still face difficulties in maintaining consistent accuracy over time due to the inherent volatility and multifaceted nature of the cryptocurrency market.

B. FUTURE CONSIDERATION

To advance our research in cryptocurrency price prediction, we aim to leverage a diverse range of technical indicators such as moving averages, oscillators, and volume-based metrics. By integrating these indicators, we seek to create a comprehensive dataset that captures the multifaceted dynamics of market behavior. This approach will facilitate the development of a sophisticated system capable of generating reliable buy/sell signals and forecasting future price trends, thereby assisting investors in minimizing risks effectively. Our strategy includes exploring hybrid models that combine statistical methods with advanced machine learning algorithms like ARIMAX with GRU or LSTM networks. This integration aims to enhance forecast accuracy and robustness, ensuring our predictive system can adapt to the volatile nature of the cryptocurrency market. Continuous refinement through techniques such as hyperparameter tuning and ensemble learning will further strengthen the predictive performance, providing valuable tools for informed decision-making and risk management in cryptocurrency trading.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to Associate Professor Nguyen Dinh Thuan and T.A. Nguyen Minh Nhut for generously dedicating your time and expertise to guide us through the implementation of our project. Your invaluable suggestions and unwavering support were instrumental in our successful completion of the project. We are deeply thankful and privileged to have had the opportunity to learn under your guidance. Your significant contributions to our learning journey are sincerely appreciated. We look forward to continuing to receive your guidance and encouragement in the future.

REFERENCES

[1] Sun, G. (2024). Cryptocurrency price prediction based on Xgboost, LightGBM and BNN. *Applied and Computational Engineering*, 49(1), 273–279. https://doi.org/10.54254/2755-2721/49/20241414

[2] Yuan, Z. (2023). Gold and Bitcoin Price Prediction based on KNN, XGBoost and LightGBM Model. *Highlights in Science, Engineering and Technology*, 39, 720–725. https://doi.org/10.54097/hset.v39i.6635

[3] Haydier, E., Albarwari, N., & Ali, T. H. (2023, December 1). The Comparison Between VAR and ARIMAX Time Series Models in Forecasting. *Iraqi Journal of Statistical Sciences*. https://doi.org/10.33899/iqjoss.2023.181260

[4] Dutta, A., Kumar, S. S., & Basu, M. (2020). A gated Recurrent unit approach to bitcoin price prediction. *Journal of Risk and Financial Management*, 13(2), 23. https://doi.org/10.3390/jrfm13020023

[5] Gunarto, D. M., Saadah, S., & Utama, D. Q. (2023). Predicting cryptocurrency price using RNN and LSTM method. *Jurnal Sistem Informasi Dan Komputer/Jurnal Sisfokom*, 12(1), 1–8. https://doi.org/10.32736/sisfokom.v12i1.1554

[6] Huang, J.-Z., Huang, W., & Ni, J. (2019). Predicting bitcoin returns using high-dimensional technical indicators. *The Journal of Finance and Data Science*, 5(3), 140–155. https://doi.org/10.1016/j.jfds.2018.10.001

[7] Mudassir, M., Bennbaia, S., Ünal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-05129-6

[8] Liao, S. (2023). Predicting the price of Bitcoin, Dogecoin and Ethereum by machine learning. *BCP Business & Management, 38, 3389–3395*. https://doi.org/10.54691/bcpbm.v38i.4312

[9] Guo, X., Ha, M., Tao, X., Li, S., Li, Y., Zhu, Z., Shen, Z., & Ma, L. (2024). Multi-Task Learning with Sequential Dependence Toward Industrial Applications: A Systematic Formulation. *ACM Transactions on Knowledge Discovery from Data, 18(5), 1-29*. https://doi.org/10.1145/3640468

[10] Demir, S., Mincev, K., Kok, J., & Paterakis, N. G. (2019). Introducing Technical Indicators to Electricity Price Forecasting: A feature engineering study for linear, ensemble, and deep machine learning models. *Applied Sciences, 10(1), 255*. https://doi.org/10.3390/app10010255

[11] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794)*.

[12] Agrawal, M., Shukla, P. K., Nair, R., Nayyar, A., & Masud, M. (2022). Stock prediction based on technical indicators using deep learning model. *Computers, Materials & Continua/Computers, Materials & Continua (Print), 70(1), 287–304*. https://doi.org/10.32604/cmc.2022.014637

[13] Vol. 12 No. 3 (2024) | *International Journal of Intelligent Systems and Applications in Engineering. (2023c, March 6)*. https://ijisae.org/index.php/IJISAE/issue/view/126

[14] Bagade, K., & Bhosale, V. (2022). Artificial Intelligence based Stock Market Prediction Model using Technical Indicators. *International Journal of Innovative Technology and Exploring Engineering, 11(6), 34–39*. https://doi.org/10.35940/ijitee.f9915.0511622

[15] Htun, H. H., Biehl, M., & Petkov, N. (2023). Survey of feature selection and extraction techniques for stock market prediction. *Financial Innovation, 9(1)*. https://doi.org/10.1186/s40854-022-00441-7