USING STATISTICAL MODEL AND MACHINE LEARNING FOR CRYPTOCURRENCY PRICE PREDICTION

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, 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-Mar-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

	DOGE	BTC	ETH
Count	2193	2193	2193
Mean	0.067	22727.117	1298.755
Median	0.053952	19191.63086	1213.599976
Mode	0.002653	6741.75	none
Min	0.002	3236.762	84.308
25%	0.003	8368.83	228.73
50%	0.054	19191.631	1213.6
75%	0.084	35510.289	1924.566
Max	0.685	67566.828	4812.087
Std	0.09	16431.978	1145.552
Variance	0.008166821	270009888.9	1312289.819
Kurtosis	6.38367187	-0.652902729	-0.181911084
Skewness	2.186196643	0.702529234	0.81204465
Range	0.683	64330.066	4727.779

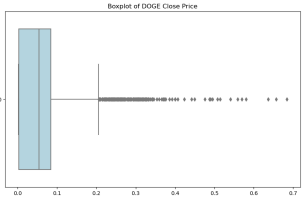

FIGURE 1. DOGE's Box Plot

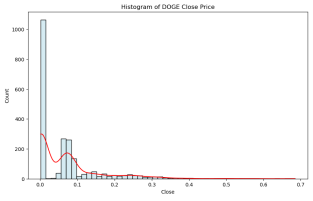

FIGURE 2. DOGE's Histogram

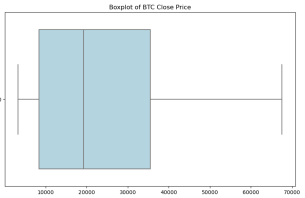

FIGURE 3. BTC's Box Plot

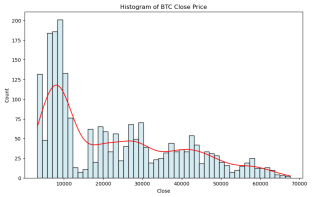

FIGURE 4. BTC's Histogram

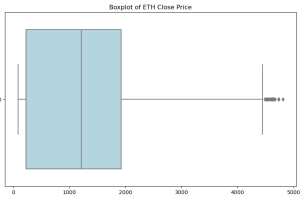

FIGURE 5. ETH's Box Plot

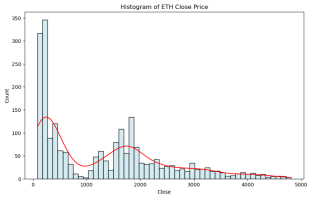

FIGURE 6. ETH's Histogram

- 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:

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

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. [8] 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:

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

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. Unlike traditional feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs, making them ideal for tasks such as time series prediction, natural language processing, and speech recognition. 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 more complex cell structure with three gates: Forget Gate: Determines what information to discard 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.

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

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

Output Gate: Regulates the amount of information passed to the output.

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

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. They combine the forget and input gates of the LSTM into a single "update gate". This update gate controls the amount of information from the previous hidden state to let through, and the amount of new information to add. The GRU's operations can be summarized as:

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

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

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

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

Where:

- z_t is the update gate.
- r_t is the reset gate.
- $\tilde{h}_t$ is the candidate hidden state.

GRUs achieve comparable performance to LSTMs in many tasks but with fewer parameters, making them computationally more efficient.

G. XGBOOST

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm known for its efficiency and performance. It's a specific implementation of gradient boosting, which is an ensemble method that combines multiple weak learners (typically decision trees) to create a strong predictive model. Here's a breakdown of XGBoost:

1) Core Concepts

Boosting: XGBoost sequentially builds decision trees, with each tree attempting to correct the errors made by previous trees. **Gradient Descent:** It uses gradient descent to minimize a loss function, guiding the model towards better predictions. **Regularization:** XGBoost incorporates regularization techniques (L1 and L2) to prevent overfitting, ensuring the model generalizes well to unseen data.

2) Mathematical Formulation

The objective function that XGBoost seeks to minimize is:

$$Obj(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}i) + \sum k = 1^K \Omega(f_k)$$

Where:

- (n) is the number of training examples.
- $\hat{y}_i$ is the loss function measuring the difference between the true label (y_i) and the predicted label ($\hat{y}_i$).
- ($\Omega(f_k)$) is the regularization term that penalizes the complexity of the individual trees (f_k), with (K) being the number of trees.

3) Key Features

Tree Pruning: XGBoost uses a pre-pruning strategy to control the complexity of individual trees, preventing them from growing too deep. **Sparsity Awareness:** It effectively handles sparse data (datasets with many missing values) by learning the optimal default direction for missing values. **Parallel Processing:** XGBoost leverages parallel processing capabilities for faster training, making it suitable for large datasets. **Built-in Cross-Validation:** It has built-in cross-validation functionality, simplifying model evaluation and parameter tuning.

4) Advantages of XGBoost

High predictive accuracy. Robustness to outliers. Handles missing data well. Feature importance analysis. Fast training and prediction speed.

5) Applications

XGBoost is widely used in various domains, including: Classification (e.g., fraud detection, spam filtering). Regression (e.g., predicting house prices, stock market forecasting). Ranking (e.g., search engine result ranking, recommendation systems).

H. LIGHTGBM

LightGBM (Light Gradient Boosting Machine) is another powerful gradient boosting framework, similar in concept to XGBoost but known for its speed and efficiency, particularly with large datasets. It achieves this through several algorithmic innovations:

1) Mathemactical Formulation

$$Obj(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \lambda\Omega(T)$$

Where:

- (n) is the number of data instances.
- (l(y_i, $\hat{y}_i$)) represents the loss function, measuring the difference between the true label ((y_i)) and the predicted label (($\hat{y}_i$)). Common loss functions include

squared error for regression and logarithmic loss for classification.
- (T) represents the set of decision trees in the ensemble.
- $(\Omega(T))$ is a regularization term to penalize model complexity and prevent overfitting.
- (λ) is a regularization parameter that controls the strength of the penalty.

2) Key Features

Gradient-based One-Side Sampling (GOSS): GOSS focuses on instances with larger gradients while downsampling those with smaller gradients. This speeds up training without significantly impacting accuracy. **Exclusive Feature Bundling (EFB):** EFB bundles mutually exclusive features (features that rarely take non-zero values simultaneously) to reduce the number of features, resulting in faster training and reduced memory usage. **Histogram-based Decision Tree Algorithm:** LightGBM uses a histogram-based algorithm for splitting features during tree construction, which is more efficient than pre-sorted algorithms, especially for large datasets. **Leaf-wise Tree Growth:** Instead of level-wise growth (growing all leaves at a level simultaneously), LightGBM grows trees leaf-wise, adding leaves one at a time to the leaf with the highest delta loss. This can lead to more accurate models but requires careful monitoring for overfitting.

3) Advantages of LightGBM

Faster training speed: Due to GOSS, EFB, and histogram-based algorithms. Lower memory usage: EFB reduces the number of features and the histogram-based algorithm uses less memory compared to pre-sorting. Higher accuracy: Leaf-wise growth strategy can lead to more complex trees that potentially capture data patterns better. Handles large datasets effectively: Optimized for speed and memory efficiency, making it well-suited for massive datasets.

4) Applications

LightGBM is used across various domains for: Classification: Object detection, image classification. Regression: Price prediction, demand forecasting. Ranking: Recommendation systems, search result ranking.

5) Comparison with XGBoost

While both LightGBM and XGBoost are gradient boosting algorithms, LightGBM typically outperforms XGBoost in terms of training speed and memory consumption, particularly for large datasets. However, XGBoost might offer better accuracy for smaller datasets. The choice between the two depends on the specific application and dataset characteristics.

V. RESULT

A. EVALUATION METHODS

VI. CONCLUSION

A. SUMMARY

B. FUTURE CONSIDERATIONS

ACKNOWLEDGMENT

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] Metsileng, L. D., Moroke, N. D., & Tsoku, J. T. (2018). Modelling the BRICS exchange rates using the Vector Autoregressive (VAR) model. *Journal of Economics and Behavioral Studies*, 10(5(J)), 220–229. https://doi.org/10.22610/jebs.v10i5(j).2511