Cryptocurrency prices forecasting using multiple Statistics, Machine Learning, Prophet algorithms.

1st Quan Huynh Quang Duong
IS403.N21.HTCL
University of Information Technology
20521224@gm.uit.edu.vn

2nd Nguyen Tu Quyen
IS403.N21.HTCL
University of Information Technology
20521823@gm.uit.edu.vn

3rd Quach Minh Dang
IS403.N21.HTCL
University of Information Technology
16520172@gm.uit.edu.vn

***Abstract*—This report aims to provide comprehensive review of cryptocurrency forecasting methods. Having Binance Coin, Bitcoin and Ethereum data, we implemented simpler to more advanced models, naming Linear Regression, ARIMA, LSTM, Bi-LSTM, GRU, and extra Exponential Smoothing, K-nearest Neighbors, Gradient Boosted Trees, Time Series Anomaly Detection (with Prophet), XGBoost models.**

Keywords—Tine series analysis, forecasting, cryptocurrency, Linear Regression, ARIMA, LSTM, Bi-LSTM, GRU, Exponential Smoothing, K-nearest Neighbors, Gradient Boosted Trees, Time Series Anomaly Detection, Prophet, XGBoost.

I. INTRODUCTION

Cryptocurrency, a form of digital currency relies on blockchain security, is the new rising star of the stock market. In the COVID-19 quarantine period, we saw an enormous growth of the former. Cryptocurrency has been gaining attention since then, although in these times, the digital coins' values have gone down a slope.

Despite the innovation, cryptocurrency is holding extreme potential and at the same time, grave risks. Fortunately, we can apply algorithms to understand the rise and fall of this cryptocurrency revolution. Our forecasting models can offer great help in the field, providing numerous methods to understand the underlying trends of the "coins".

In this report, we are going to see the effectiveness of each 9-10 different models, naming: Linear Regression, ARIMA, LSTM and Bi-LSTM, GRU, Exponential Smoothing, K-nearest Neighbors, Gradient Boosted Trees, Time Series Anomaly Detection with Prophet, XGBoost; on the Binance Coin, Bitcoin and Ethereum dataset for observation.

In the evaluating process, we are relying on Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Explained Variance Score (EVS) to make sure of the forecasting models' efficiency.

II. RELATED RESEARCHES

Our research will compare the prediction of cryptocurrency by using these prediction models: Linear Regression, LSTM, Bi-LSTM, , ARIMA, GRU, ETS, KNN, GBT, Prophet, and XGBoost. In terms of work, Mahir Iqbal and team [1] used ARIMA, Prophet and XGBoosting for time series analysis as a machine learning techniques; and result is ARIMA considered as the best model for forecasting Bitcoin price in the crypto-market with RMSE score of 322.4 and MAE score of 227.3. Another case in point is Ruchi Mittal Shefali Arora and M.P.S Bhatia [2], who predict Automated Cryptocurrency prices prediction using Linear Regression and the result with the highest accuracy 99.383%. Aside from that, Patrick Jaquart, David Dann, Christof Weinhardt [3] used GRU and LSTM to make Short-term bitcoin market prediction, who predict the binary market movement with accuracies ranging from 50.9% to 56.0%. Furthermore, Febri Liantoni [4] used Double Exponential Smoothing to forecast Bitcoin, the result of tests is the lowest mean absolute percentage error (MAPE) value is 2.89%, with the best alpha at 0.9. Others, Tianyu Ray Li and team [5] predict alternative cryptocurrency price using Gradient Boosting Tree Model. In addition, Kate Murray and team [6] forecasting cryptocurrency prices with RNN and other algorithms, and KNN provides an excellent trade-off between the accuracy of the prediction and the computational effort required, with an error 4.8%.

III. METHODOLOGY

A. Materials

The datasets were collected on yahoofinance.com website, including:

- Binance Coin – BNB (11/9/2017 – 6/14/2023),
- Bitcoin – BTC (9/17/2014 – 6/14/2023).
- Ethereum – ETH (11/9/2017 – 6/14/2023).

The records are official and trustworthy, ready for analyzing.

Below table shows the complete description of features of dataset:

TABLE I. FEATURES DESCRIPTION

Feature	**Description**	**Unit**

Date	The time at which each instance or entry has been collected from cryptocurrency stock market.	mm/dd/yyyy format
Open	Open price for each day according to each timestamp.	USD
High	Highest price on that day in which data has been collected.	USD
Low	Lowest price on that day in which data has been collected.	USD
Close	Final price on that day in which data has been collected.	USD
Adj Close	Adjusted closing price.	USD
Volume	Total number of shares or contracts traded.	Numeric

We will be working with the "Close" feature.

TABLE II. STATISTICAL DESCRIPTION

	BNB	**BTC**	**ETH**
Count	2044	3193	2044
Mean	159.678	13496.129	1177.159
STD	180.725	16013.144	1151.280
Min	1.510	178.103	84.308
25%	14.891	745.691	220.875
50%	28.934	7556.238	644.950
75%	307.337	19796.809	1812.780
Max	675.684	67566.828	4812.087

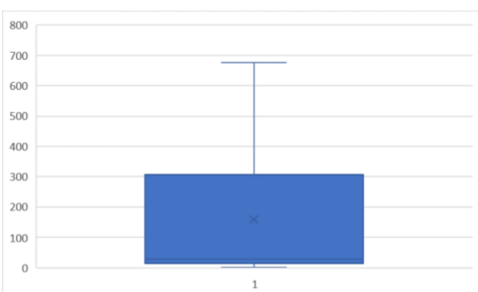

Fig. 1. BNB's Box & Whisker chart

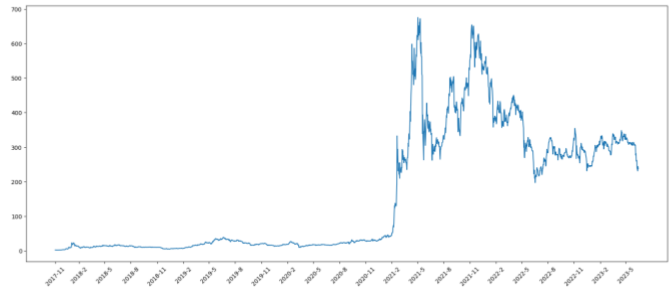

Fig. 2. BNB's Line chart

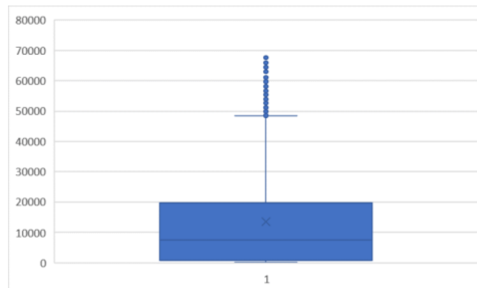

Fig. 3. BTC's Box & Whisker chart

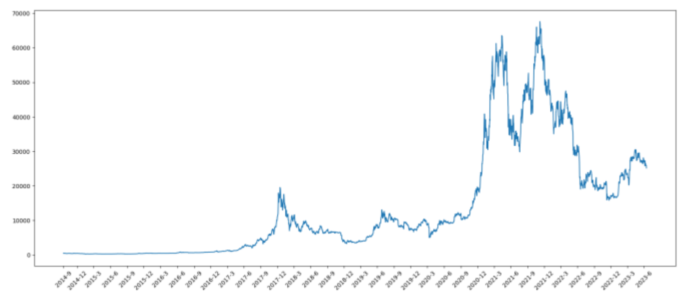

Fig. 4. BTC's Line chart

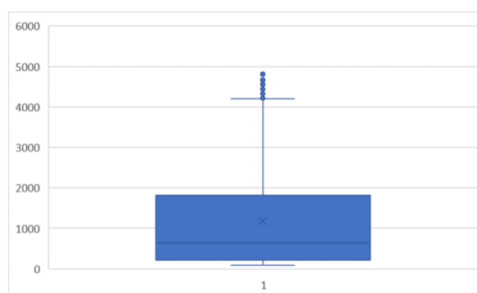

Fig. 5. ETH's Box & Whisker chart

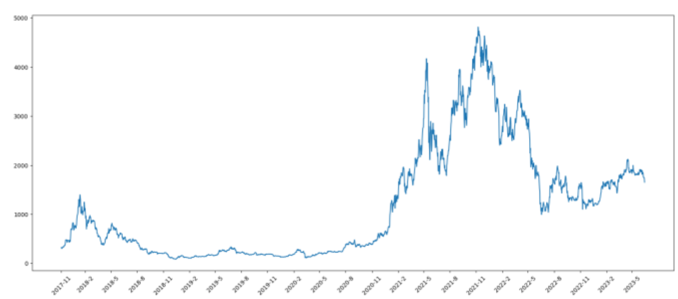

Fig. 6. ETH's Line chart

B. Modelling

1) Linear Regression

Linear Regression is one of the most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression algorithm shows a linear relationship between a dependent variable and one or more independent variables. The linear regression model provides a sloped straight line representing the relationship between the variables. [7]

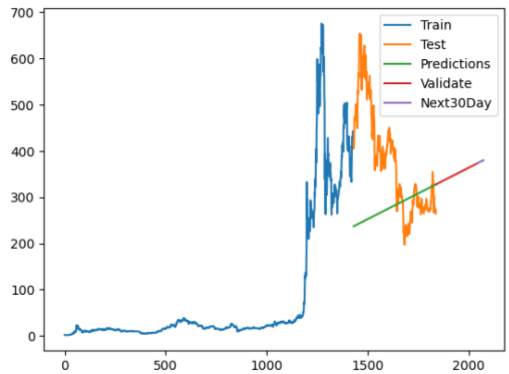

Fig. 1. Linear Regression model's predictions for BNB data

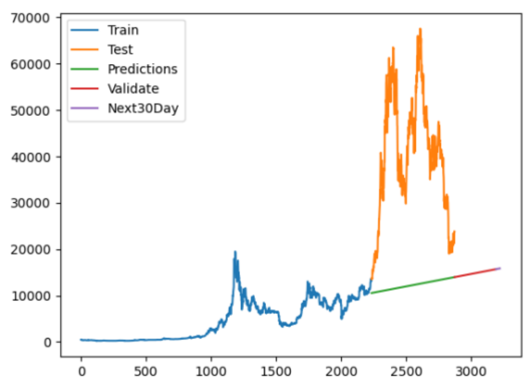

Fig. 2. Linear Regression model's predictions for BTC data

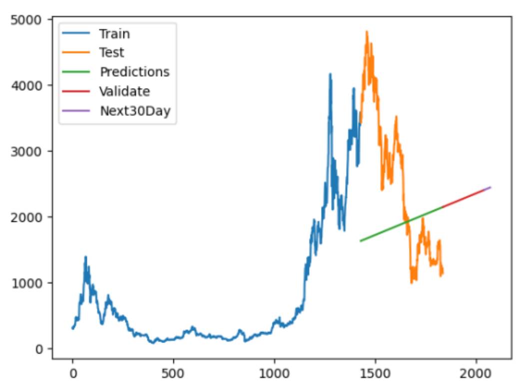

Fig. 3. Linear Regression model's predictions for ETH data

2) Autoregressive Integrated Moving Average (ARIMA)

An ARIMA model is a class of statistical models for analyzing and forecasting time series data. It is really simplified in terms of using it, Yet this model is really powerful. It combines autoregressive (AR) and moving average (MA) components, along with differencing operations, to capture the underlying patterns and predict future values based on the historical behavior of the data.

The parameters of the ARIMA (p,d,q) model are defined as follows:

- p: The number of lag observations included in the model, also called the lag order.
- d: The number of times that the raw observations are differenced, also called the degree of difference.
- q: The size of the moving average window, also called the order of moving average. [8]

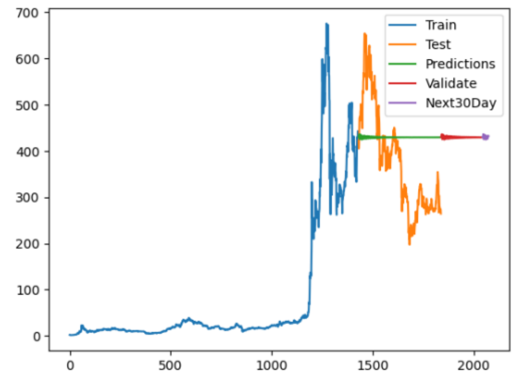

Fig. 4. ARIMA model's predictions for BNB data

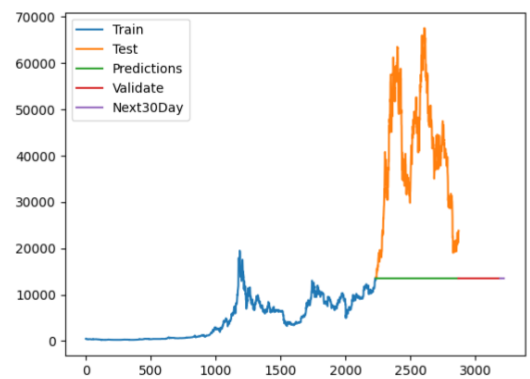

Fig. 5. ARIMA model's predictions for BTC data

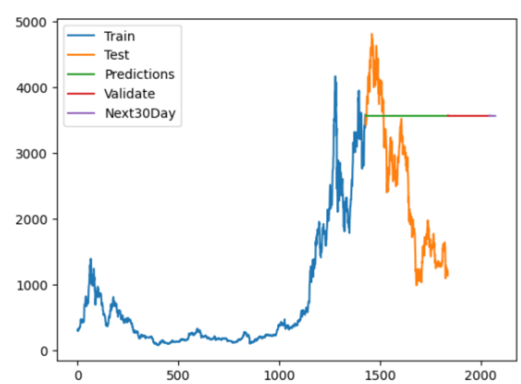

Fig. 6. ARIMA model's predictions for ETH data

3) Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory

Long Short-Term Memory networks – usually just called "LSTM" which were introduced by Hochreiter & Schmidhuber (1997).

LSTM is an artificial neural network used in the fields of artificial intelligence and deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. Such a recurrent neural network (RNN) can process not only single data points (such as images), but also entire sequences of data (such as speech or video). LSTM is a special kind of RNN, which shows outstanding performance on a large variety of problems.

The LSTM neural network structure introduces three logical structures, input gate, output gate, and forgetting gate, based on classical cyclic neural network.

In the LSTM network structure, the input gate , output gate, and forgetting gate use i, o, and f respectively, and the memory unit is C, the input data is X, and the implicit state is H. [9]

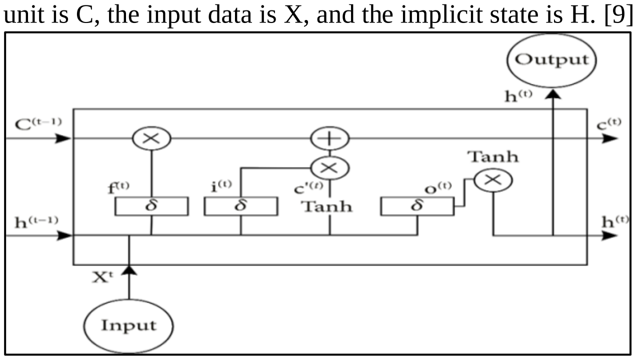

Fig. 7. Schematic diagram of the LSTM neural network structure

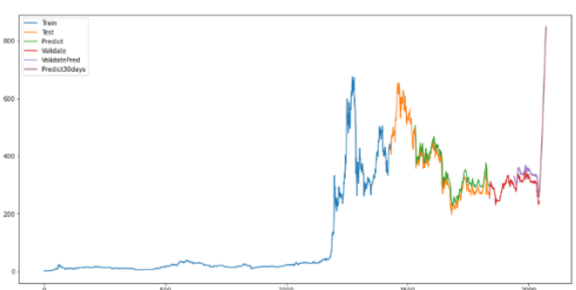

Fig. 8. LSTM model's predictions for BNB data

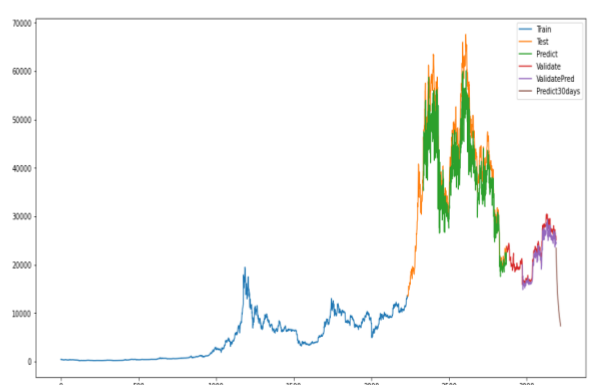

Fig. 9. LSTM model's predictions for BTC data

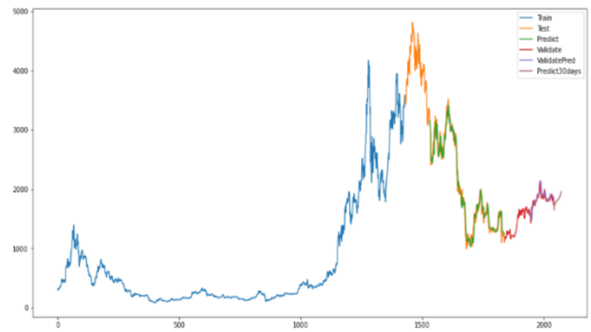

Fig. 10. LSTM model's predictions for ETH data

Bi-LSTM: A bidirectional LSTM consists of two LSTM layers: a forward LSTM and a backward LSTM. The forward LSTM processes the input sequence from the beginning to the end, while the backward LSTM processes the input sequence from the end to the beginning. The outputs from the two LSTMs are then concatenated to obtain the final output.

By processing the input sequence in both directions, a bidirectional LSTM can capture information from both past and future contexts. This can be useful in tasks where the context is important for understanding the input sequence, such as natural language processing. For example, in machine translation, bidirectional LSTMs can be used to encode the source sentence in both directions, which can help the decoder to generate more accurate translations.

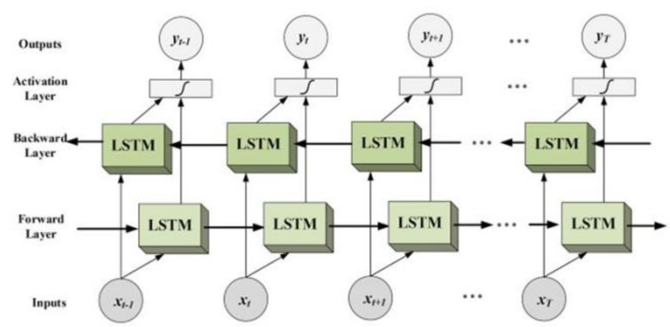

Fig. 11. Bi-LSTM visualization

4) Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is a type of Recurrent Neural Network (RNN). GRU neural network solves the gradient explosion problem of the recurrent neural network (RNN) model. The core of this algorithm is a combination of a sigmoid function and some relational operations, which together constitute the GRU control and information selection mechanism. [10]

GRU uses an update gate and reset gate to determine what past information can be kept or forgotten. While GRU is similar to LSTM, it combines LSTM's forget and input gates into a single update gateway. [11]

The update gate is represented by z_t, the input is represented by x_t,the reset gate is represented by r_t, and the parameters representing output are y_t.

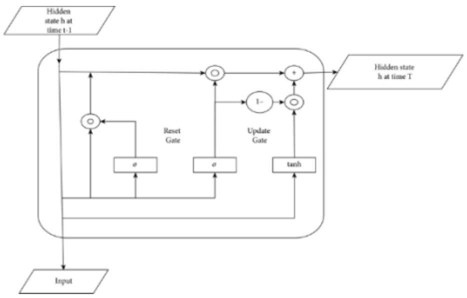

Fig. 12. GRU internal control gate structure.

GRU algorithm is proceeded by the following steps:

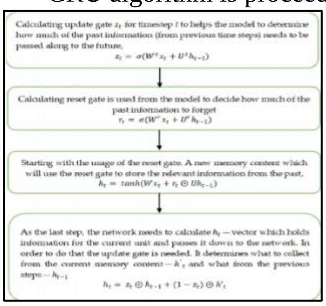

Fig. 13. GRU algorithm steps

After determining h_t at the current moment, the output of this neural unit can be completed, and finally, y_t data can be obtained after calculation. The output process is as follows:

$$y_t = \sigma(W_o h_t)$$

Because of this special result, each unit has new input information and genetic information at the previous moment, which makes the network have the characteristics of dealing with nonlinear feature sequences. [12]

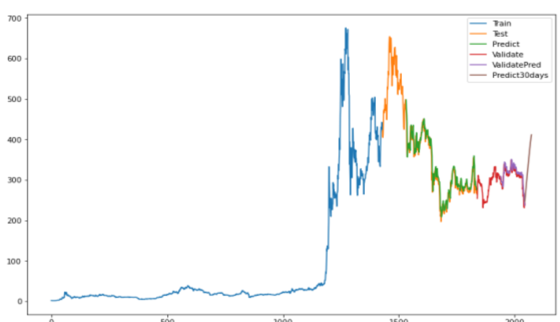

Fig. 14. GRU model's predictions for BNB data

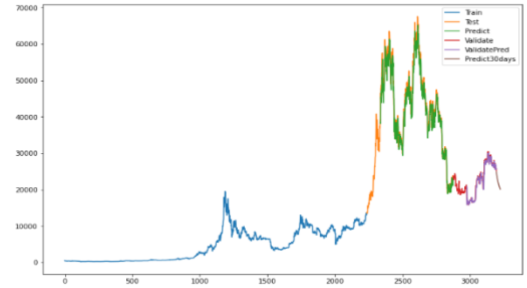

Fig. 15. GRU model's predictions for BTC data

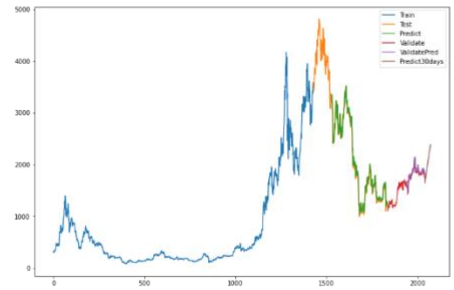

Fig. 16. GRU model's predictions for ETH data

5) Exponential Smoothing (ETS)

The ETS models are a family of time series models with an underlying state space model consisting of a level component, a trend component (T), a seasonal component (S), and an error term (E). [13]

There are various types of exponential smoothing forecasting methods. But the one that is best for time series data is Triple Exponential Smoothing. It is an extension of Exponential Smoothing that explicitly adds support for seasonality to the univariate time series. This method is also called Holt-Winters Exponential Smoothing, named for two contributors to the method: Charles Holt and Peter Winters. [14]

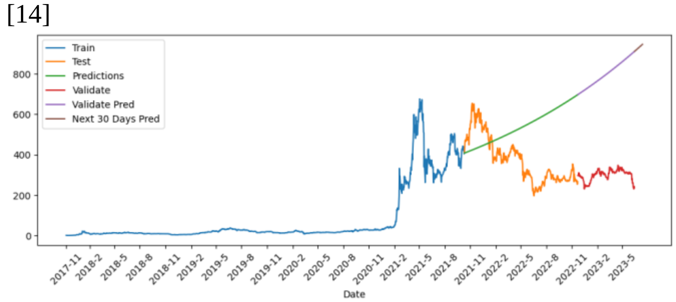

Fig. 17. ETS model's predictions for BNB data

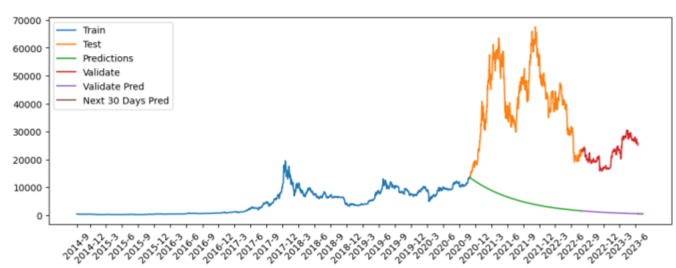

Fig. 18. ETS model's predictions for BTC data

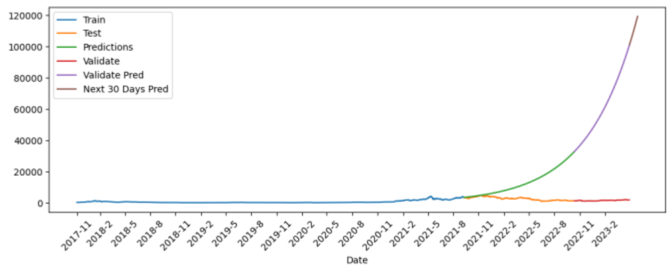

Fig. 19. ETS model's predictions for ETH data

Conclusion: ETS is just a statistics model, we can not get any good recommendation from it. However, it provided us more insight into the data by showing the exponential change in trends.

6) K-nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) is a non-parametric classification algorithm, also known is a supervised machine learning algorithm that is predominantly used for classification purposes.

We use KNN regression with the continuous or numerical datasets. Because it is suitable for datasets where the target variable is a continuous value that you want to predict.

KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same *neighbourhood*. KNN regression also uses the same distance functions as KNN classification.

$$\text{Euclidean: } d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$\text{Manhattan: } d = \sqrt{\sum_{i=1}^{n}|x_i - y_i|}$$

KNN regression is an instance based lazy learning algorithm. It learns complex target function quickly without losing information. For a given input x of training data, K observations with xi in the proximity are considered and the average of the response of those K independent variables gives y:

$$\hat{y}\,(x) = \frac{1}{k}\sum_{x_i \in N_{k(x)}} y_i$$

Where $N_{k(x)}$ depicts K closest points in the neighborhood of x. various distance measures quantify closeness between points but Euclidean distance is commonly practiced. [15]

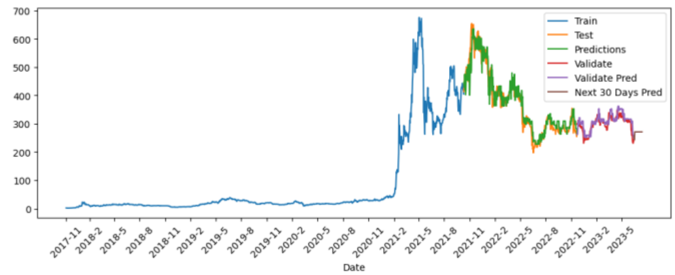

Fig. 20. KNN model's predictions for BNB data

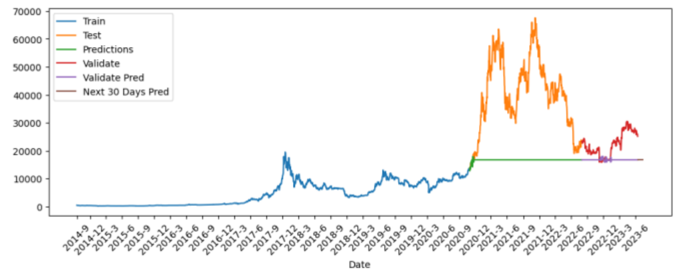

Fig. 21. KNN model's predictions for BTC data

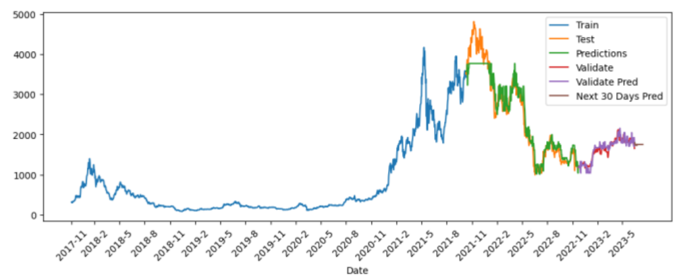

Fig. 22. KNN model's predictions for ETH data

Conclusion: KNN model did a decent job predicting the underlying trends of BNB and ETH data, but quite bad on BTC data. The reason behind it is that Bitcoin had a unforseen growth that blew up, making it hard for a simpler regression model.

7) Gradient Boosted Trees (GBT)

Gradient Boosted Trees is an ensemble learning method that combines multiple weak learners (i.e., decision trees) to create a strong model. It has been shown to be effective in a wide range of applications, especially there is finance.

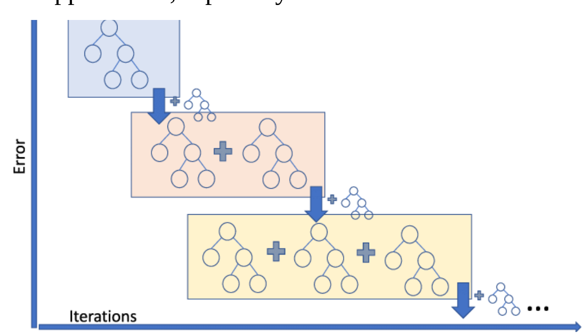

Fig. 23. GBT model visualizing

Firstly, initialize the model with a constant value, such as the mean of the target variable. For each iteration (or tree), fit a new decision tree to the negative gradient of the loss function (i.e., squared error) with respect to the current model predictions.

$$SE = \frac{\Sigma(y\_pred - y\_true)^2}{n}$$

where Σ represents the sum from 1 to n, y_pred is the predicted value, y_true is the actual value, and n is the number of samples. After that, add the new tree to the current model by multiplying its predictions by a learning rate (a hyperparameter that controls the contribution of each tree) and adding it to the previous predictions. [16]

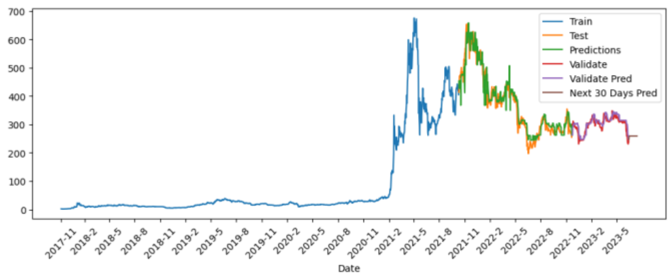

Fig. 24. GBT model's predictions for BNB data

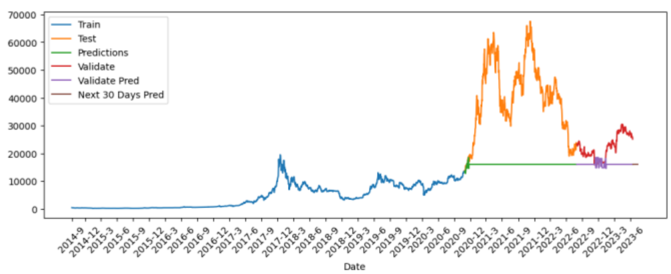

Fig. 25. 1GBT model's predictions for BTC data

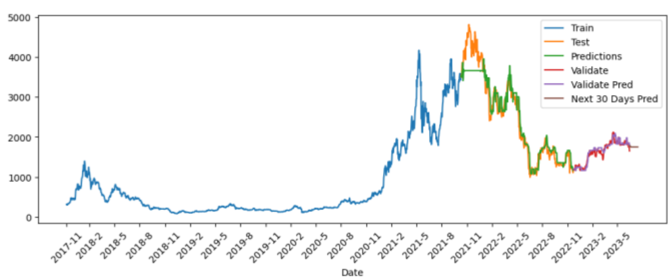

Fig. 26. GBT model's predictions for ETH data

Conclusion: Just like KNN model, GBT did a decent job predicting the underlying trends of BNB and ETH data, but quite bad on BTC data. The reason behind it is that Bitcoin had a unforseen growth that blew up, making it hard for a simpler regression model.

8) Time Series Anomaly Detection with Prophet

Time series data is a collection of observations obtained through repeated measurements over time. Plot the points on a graph, and one of your axes would always be time. [17]

When building a time series model, the dataset may have anomalies or outliers. Anomalies are observations or data points that deviate from normal behavior. When anomalies are left undetected in the dataset, they harm the model's performance [18]. So we will use Prophet to build an anomaly detection model.

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. [19]

Time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation: [20]

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

Where:

- **g(t)**: piecewise linear or logistic growth curve for modelling non-periodic changes in time series (trend).
- **s(t)**: periodic changes (e.g. weekly/yearly seasonality).
- **h(t)**: effects of holidays (user provided) with irregular schedules.
- **ε_t**: error term accounts for any unusual changes not accommodated by the model.

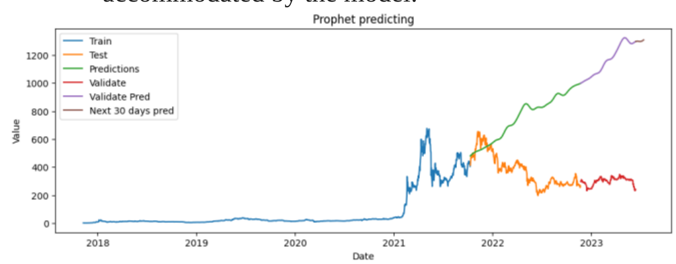

Fig. 27. 2Prophet model's predictions for BNB data

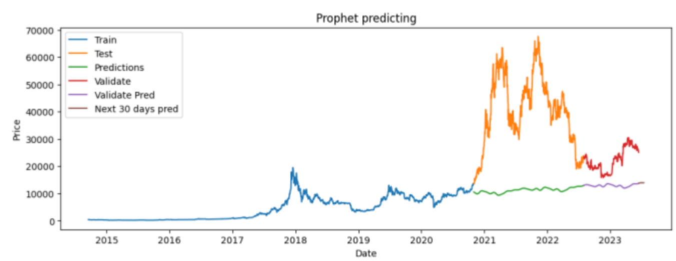

Fig. 28. 3Prophet model's predictions for BTC data

Prophet model's predictions for BNB data

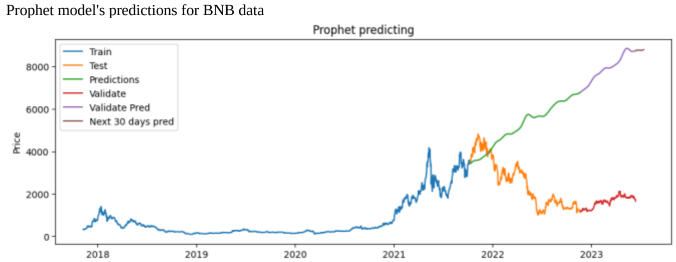

Fig. 29. 4Prophet model's predictions for ETH data

Next, we are going to handle anomaly by plotting them. Anomaly will be detected if the absolute of residuals between the predicted values and actual values are greater than 3 times of its standard deviation.

Here are the anomalies in our data.

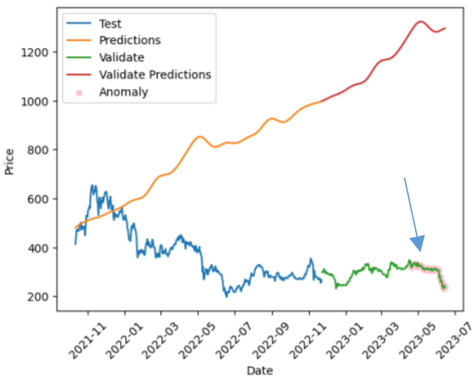

Fig. 30. 5Anomalies of BNB data

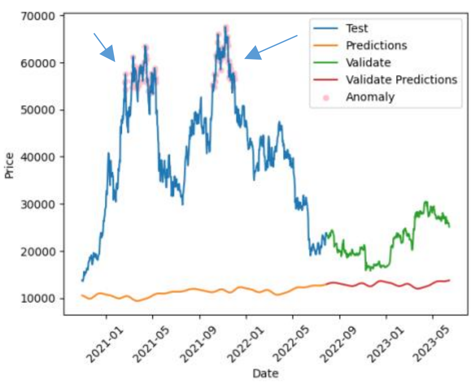

Fig. 31. 6Anomalies of BTC data

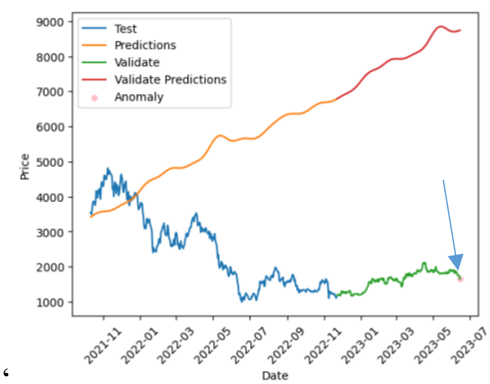

Fig. 32. 7Anomalies of ETH data

9) *Extreme Gradient Boosting (XGBoost)*

XGBoost is an implementation of Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.[21]

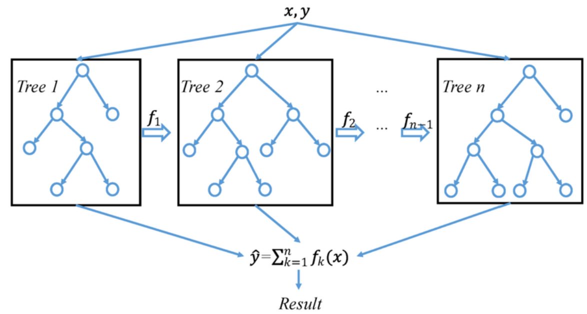

Fig. 33. 8XGBoost model visualising

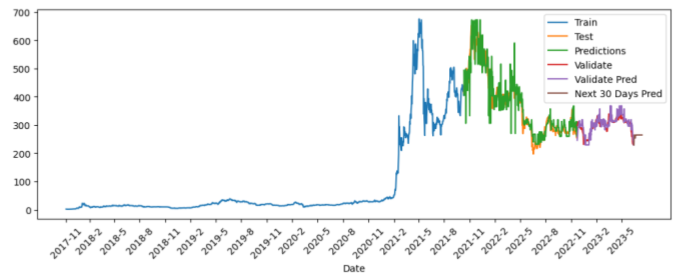

Fig. 34. 9XGBoost model's predictions for BNB data

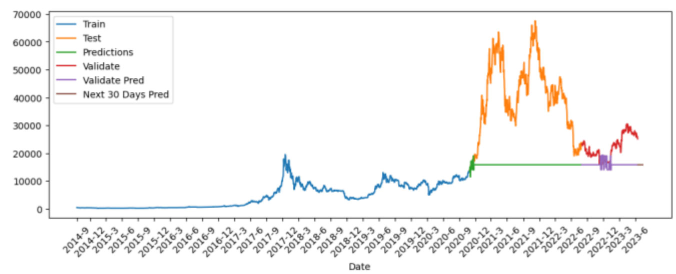

Fig. 35. 10XGBoost model's predictions for BTC data

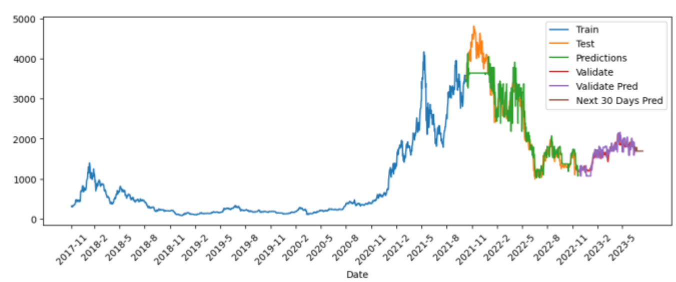

Fig. 36. 11XGBoost model's predictions for ETH data

Conclusion: Comparing to GBT, XGBoost did not make any differences that is worth noting. We can even see the inefficiency of the model in the evaluating error and score metrics.

C. *Evaluating*

The metrics we are using are Root Mean Squared Error, Mean Absolute Percentage Error and Explained Variance Score.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{1,i} - x_{2,i}\right)^2}$$

Fig. 37. 12RMSE formula

$$MAPE = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{x_{2,i} - x_{1,i}}{x_{1,i}} \right|$$

Fig. 38. 13MAPE formula

$$EV = 1 - \frac{Var[Y - \tilde{Y}]}{Var[Y]}$$

Fig. 39. 14EV formula

Here are a big table containing the scores and errors of all previous models:

Model	Train - Test - Val	RMSE			MAPE			EV		
		BNB	BTC	ETH	BNB	BTC	ETH	BNB	BTC	ETH
Linear	6:2:2	384.508	21744.928	2681.006	0.848	0.402	0.827	0.011	0.093	0.014
	7:2:1	164.599	31021.516	1376.006	0.283	0.656	0.434	-0.447	-0.009	0.434
	8:1:1	130.258	19447.194	1361.74	0.463	0.368	0.931	-0.463	-0.175	-0.405
	AVG	**226.455**	**24071.213**	**1361.735**	**0.531**	**0.475**	**0.731**	**-0.300**	**-0.030**	**0.014**
ARIMA	6:2:2	335.937	24827.845	3983.879	0.671	0.512	1.118	-4.845	0	-4.44
	7:2:1	124.531	29915.812	1520.038	0.339	0.621	0.793	0	0	0
	8:1:1	109.579	12838.527	1321.289	0.389	0.323	0.908	-0.002	0	0
	AVG	**190.016**	**22527.395**	**2275.069**	**0.466**	**0.485**	**0.940**	**-1.616**	**0.000**	**-1.480**
LTSM	6:2:2	183.787	3121.797	321.894	0.338	0.050	0.069	-0.050	0.976	0.891
	7:2:1	14.162	4598.482	120.222	0.034	0.074	0.047	0.96	0.9	0.974
	8:1:1	10.017	2014.385	83.570	0.023	0.049	0.039	0.769	0.973	0.861
	AVG	**69.322**	**3244.888**	**175.229**	**0.132**	**0.058**	**0.043**	**0.865**	**0.950**	**0.909**
Bi-LSTM	6:2:2	67.697	2836.230	211.599	0.115	0.047	0.052	0.828	0.982	0.954
	7:2:1	20.841	3772.482	110.550	0.039	0.064	0.034	0.978	0.96	0.99
	8:1:1	16.066	1632.480	82.326	0.045	0.031	0.038	0.885	0.986	0.946
	AVG	**34.868**	**2747.064**	**134.825**	**0.066**	**0.047**	**0.041**	**0.897**	**0.976**	**0.963**
GRU	6:2:2	29.001	1615.601	134.935	0.050	0.030	0.032	0.94	0.993	0.963
	7:2:1	14.751	2495.037	97.686	0.036	0.043	0.037	0.964	0.973	0.982
	8:1:1	9.831	1293.870	86.634	0.021	0.028	0.045	0.773	0.98	0.879
	AVG	**17.861**	**1,801.503**	**106.418**	**0.036**	**0.034**	**0.038**	**0.892**	**0.982**	**0.941**
ETS	6:2:2	767,133.133	29,061.864	10,206.232	858.649	0.741	2.293	-47042015.32	-0.158	-100.001
	7:2:1	249.295	37,570.469	7,742.734	0.708	0.832	3.913	-1.764	-0.156	-18.575
	8:1:1	96.170	41,574.735	1,126.347	0.337	1.125	0.775	-0.095	-4.161	0.224
	AVG	**255,826.199**	**36,069.023**	**6,358.438**	**286.565**	**0.899**	**2.327**	**-15,680,672.393**	**-1.492**	**-39.451**
KNN	6:2:2	200.623	18,720.236	1,515.814	0.370	0.287	0.394	0.007	0.321	0.001
	7:2:1	33.461	26,974.463	312.602	0.064	0.533	0.091	0.914	0.017	0.918
	8:1:1	24.206	3,605.806	186.411	0.066	0.080	0.093	0.627	0.921	0.781
	AVG	**86.097**	**16,433.502**	**671.609**	**0.167**	**0.300**	**0.193**	**0.516**	**0.420**	**0.567**
GBT	6:2:2	203.422	19,145.031	1,669.215	0.377	0.291	0.451	-0.001	0.295	-0.032
	7:2:1	30.630	27,582.398	331.654	0.062	0.551	0.089	0.929	0.012	0.909
	8:1:1	24.600	3,554.286	176.202	0.070	0.077	0.086	0.612	0.921	0.805
	AVG	**86.217**	**16,760.572**	**725.690**	**0.170**	**0.306**	**0.209**	**0.513**	**0.409**	**0.561**
TSAD with Prophet	6:2:2	189.351	18,261.278	953.480	0.317	0.439	0.245	-0.619	0.232	-0.344
	7:2:1	461.980	32,013.567	3,426.798	1.322	0.683	1.791	-4.309	-0.033	-2.543
	8:1:1	190.275	33,817.925	2,229.814	0.686	0.931	1.530	0.2	-1.741	-0.505
	AVG	**280.535**	**28,030.923**	**2,203.364**	**0.775**	**0.684**	**1.189**	**-1.576**	**-0.514**	**-1.131**
XGBoost	6:2:2	203.138	19,258.079	1,683.644	0.377	0.304	0.457	-0.004	0.288	-0.036
	7:2:1	43.836	27,778.458	375.735	0.080	0.558	0.106	0.851	0.008	0.883
	8:1:1	25.956	4,334.414	187.111	0.072	0.092	0.094	0.521	0.882	0.778
	AVG	**90.977**	**17,123.650**	**748.830**	**0.176**	**0.318**	**0.219**	**0.456**	**0.393**	**0.542**

Fig. 40. Models metric table

LSTM, Bi-LSTM and GRU are the best models based on 3 different metrics. They are all Machine Learning/Deep Learning models that fit best to Time Series data. We can rely on these 3 models to recommend or give advice in the stock market.

We are going to compare the unsupervised predictions for the next 30 days, with one machine learning model and one regression model to compare the difference between two types of algorithms.

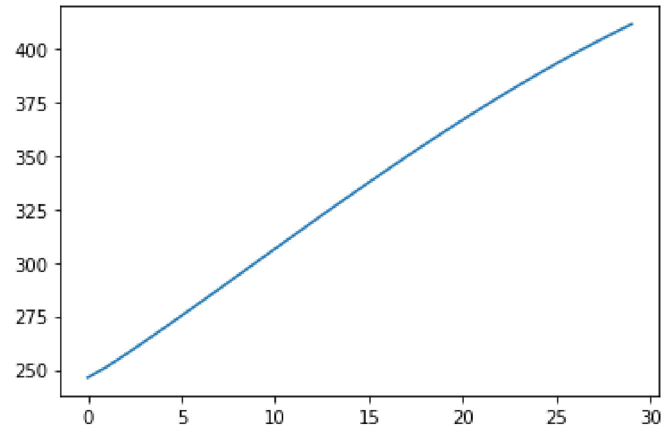

Fig. 41. GRU model's predictions for the next 30 days of BNB data

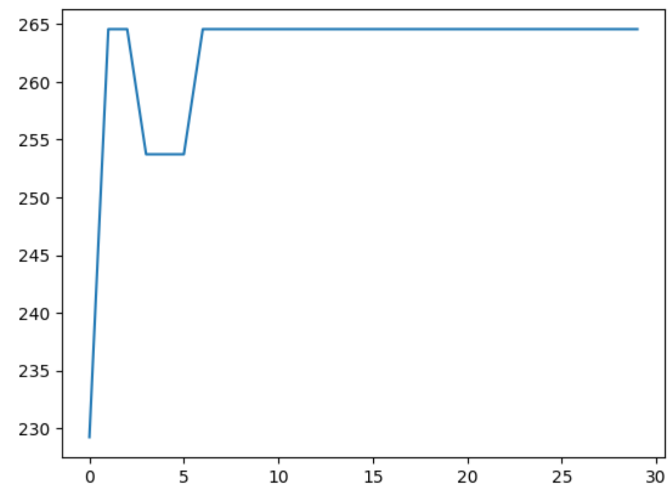

Fig. 42. XGBoost model's predictions for the next 30 days of BNB data

While both models predicted there will be a rise in value of BNB, XGBoost predict that there will also be a slight decrease. Note that, the predictions start from the day June 15th.

Date	Open	High	Low	Close*	Adj Close**	Volume
Jun 17, 2023	238.82	241.27	237.50	241.27	241.27	485,867,392
Jun 16, 2023	236.27	242.80	232.83	239.11	239.11	488,582,672
Jun 15, 2023	237.61	240.16	231.07	236.28	236.28	502,420,772
Jun 14, 2023	243.89	251.63	234.04	237.57	237.57	738,634,212
Jun 13, 2023	231.08	245.64	229.31	243.89	243.89	864,407,068
Jun 12, 2023	235.40	238.47	222.07	231.05	231.05	1,014,433,334
Jun 11, 2023	239.09	239.35	234.43	235.44	235.44	376,213,808
Jun 10, 2023	260.70	260.72	233.17	239.09	239.09	1,068,424,942

Fig. 43. Official BNB data price

Looking at the real data, we should not have any conclusions yet. However, both models are going on a right direction, while XGBoost could oversee a tiny slump of the series.

ACKNOWLEDGMENT

First and foremost, we would like to thank Assoc Prof. Dr. Nguyen Dinh Thuan and Mr. Nguyen Minh Nhut for all the knowledge they enlightened us.

This report could not have made it without the help of Assoc Prof. Dr. Nguyen Dinh Thuan and Mr. Nguyen Minh Nhut.

REFERENCES

[1] M. Iqbal, M. Iqbal, F. Jaskani, K. Iqbal, and A. Hassan, "Time-Series Prediction of Cryptocurrency Market using Machine Learning Techniques," *EAI Endorsed Trans. Creat. Technol.*, vol. 8, no. 28, p. 170286, Aug. 2021, doi: 10.4108/eai.7-7-2021.170286.

[2] R. Mittal, S. Arora, and M. P. S. Bhatia, "AUTOMATED CRYPTOCURRENCIES PRICES PREDICTION USING MACHINE LEARNING," *ICTACT J. SOFT Comput.*, vol. 08, no. 04, 2018.

[3] P. Jaquart, D. Dann, and C. Weinhardt, "Short-term bitcoin market prediction via machine learning," *J. Finance Data Sci.*, vol. 7, pp. 45–66, Nov. 2021, doi: 10.1016/j.jfds.2021.03.001.

[4] F. Liantoni and A. Agusti, "Forecasting Bitcoin using Double Exponential Smoothing Method Based on Mean Absolute Percentage Error," *JOIV Int. J. Inform. Vis.*, vol. 4, Apr. 2020, doi: 10.30630/joiv.4.2.335.

[5] T. R. Li, A. S. Chamrajnagar, X. R. Fong, N. R. Rizik, and F. Fu, "Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model," *Front. Phys.*, vol. 7, 2019, Accessed: Jun. 18, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fphy.2019.00098

[6] K. Murray, A. Rossi, D. Carraro, and A. Visentin, "On Forecasting Cryptocurrency Prices: A Comparison of Machine Learning, Deep Learning, and Ensembles," *Forecasting*, vol. 5, no. 1, Art. no. 1, Mar. 2023, doi: 10.3390/forecast5010010.

[7] "Linear Regression in Machine learning - Javatpoint," *www.javatpoint.com*. https://www.javatpoint.com/linear-regression-in-machine-learning (accessed Jun. 17, 2023).

[8] P. Pathak, "How to Create an ARIMA Model for Time Series Forecasting in Python," *Analytics Vidhya*, Oct. 29, 2020. https://www.analyticsvidhya.com/blog/2020/10/how-to-create-an-arima-model-for-time-series-forecasting-in-python/ (accessed Jun. 17, 2023).

[9] Y. Yang, "Application of LSTM Neural Network Technology Embedded in English Intelligent Translation," *Comput. Intell. Neurosci.*, vol. 2022, p. e1085577, Sep. 2022, doi: 10.1155/2022/1085577.

[10] J. Rong, D. Wang, B. Zhang, and Y. Wang, "Research on the Forecast of the Combined GRU Model Based on the Optimized Bat Algorithm in the Bank Transaction Volume," *Mob. Inf. Syst.*, vol. 2022, p. e6837395, Jun. 2022, doi: 10.1155/2022/6837395.

[11] M. Yurtsever, "Gold Price Forecasting Using LSTM, Bi-LSTM and GRU," *Eur. J. Sci. Technol.*, Dec. 2021, doi: 10.31590/ejosat.959405.

[12] A. Darmawahyuni, S. Nurmaini, S. Sukemi, W. Caesarendra, V. Bhayyu, and M. Naufal Rachmatullah, "Deep Learning with a Recurrent Network Structure in the Sequence Modeling of Imbalanced Data for ECG-Rhythm Classifier," *Algorithms*, vol. 12, pp. 1–12, Jun. 2019, doi: 10.3390/a12060118.

[13] J. Perktold *et al.*, "statsmodels/statsmodels: Release 0.14.0." Zenodo, May 05, 2023. doi: 10.5281/ZENODO.593847.

[14] J. Brownlee, "A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python," *MachineLearningMastery.com*, Aug. 19, 2018. https://machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/ (accessed Jun. 17, 2023).

[15] R. Goyal, P. Chandra, and Y. Singh, "Suitability of KNN Regression in the Development of Interaction based Software Fault Prediction Models," *IERI Procedia*, vol. 6, pp. 15–21, Jan. 2014, doi: 10.1016/j.ieri.2014.03.004.

[16] "Gradient Boosting model." Accessed: Jun. 17, 2023. [Online]. Available: https://towardsdatascience.com/machine-learning-part-18-boosting-algorithms-gradient-boosting-in-python-ef5ae6965be4

[17] "What is Time Series Data? | Definition, Examples, Types & Uses," *InfluxData*. https://www.influxdata.com/what-is-time-series-data/ (accessed Jun. 17, 2023).

[18] "Anomaly Detection Model on Time Series data," *Engineering Education (EngEd) Program | Section*. https://www.section.io/engineering-education/anomaly-detection-model-on-time-series-data/ (accessed Jun. 17, 2023).

[19] "Prophet," *Prophet*. http://facebook.github.io/prophet/ (accessed Jun. 17, 2023).

[20] A. Choudhary, "Generate Quick and Accurate Time Series Forecasts using Facebook's Prophet (with Python & R codes)," *Analytics Vidhya*, May 10, 2018. https://www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/ (accessed Jun. 17, 2023).

[21] "XGBoost," *GeeksforGeeks*, Sep. 18, 2021. https://www.geeksforgeeks.org/xgboost/ (accessed Jun. 17, 2023).