Bitcoin Price Prediction using Multivariate LSTM Model with Tweet Sentiments

Yasemin Çevik
Ozyegin University
Istanbul, Turkey
yasemin.cevik@ozu.edu.tr

Umut Oskay
Ozyegin University
Istanbul, Turkey
umut.oskay@ozu.edu.tr

***Abstract*—This paper offers a thorough use of neural network techniques for price prediction of bitcoin. The volatile behavior of cryptocurrency markets has prompted a thorough investigation into the use of predictive models to estimate asset values. In order to predict the price of Bitcoin, this research uses a Multivariate Long Short-Term Memory (LSTM) model, with the sentiment included in tweets serving as an input. This project intends to combine historical price data with textual information from Twitter to estimate future price movements of Bitcoin by utilizing the capabilities of Long Short-Term Memory (LSTM), a type of recurrent neural network noted for its ability to record sequential dependencies. Using a multivariate technique that includes several features extracted from Twitter sentiment analysis and historical performance of Bitcoin, this research aims to improve prediction accuracy. Promising insights into the potential value of social media sentiment in enhancing predictive models for bitcoin price forecasting are provided by the outcomes and implications of this research.**

***Index Terms*—Bitcoin Price Prediction, Multivariate LSTM Model, Sentiment Analysis, Time Series Analysis, Social Media Data, Cryptocurrency Forecasting, Historical Price Data**

I. Introduction

Within the ever-changing cryptocurrency landscape, Bitcoin is a notable bellwether that frequently predicts the sentiment and direction of the market. It is an interesting but difficult asset to estimate the price of because to its volatility character, which is influenced by a wide range of factors. Since they ignore the distinctions between real currencies and cryptocurrencies, certain strategies put out to forecast currency price fluctuations are ineffective [1]. This research proposes a effective approach to Bitcoin price prediction by utilizing the power of a Multivariate Long Short-Term Memory (LSTM) Forecast Model supplemented by a unique dataset that combines the sentiments extracted from tweets linked to Bitcoin.

The unique aspect of our methodology is how we used sentiment from social media as a major element in our forecasting model. We acknowledge the considerable influence of public opinion on social media sites such as Twitter on market fluctuations. Since Bitcoin is still relatively new as a money, its volatility is significantly higher than that of conventional currencies, making it a unique potential for price prediction [2]. Because Bitcoin data is temporally structured, recurrent neural networks (RNNs)[16] and long short-term memory (LSTM) are preferred over conventional multilayer perceptrons (MLPs) [3]. The Multivariate LSTM model[14], which is well-known for its adeptness in managing time series data and its capacity to retain long-term dependencies, is what we use to try to capture the complex dynamics that exist between market data and user sentiment. With the help of this advanced model, we can analyze not just conventional financial indicators but also the nuanced yet potent clues gleaned from the collective views expressed on social media, providing a more complete picture of possible market movements.

Readers will be guided through the complex process of integrating tweet sentiment analysis into the LSTM model for Bitcoin price prediction in this article. For individuals interested in the convergence of financial forecasting, machine learning, and social media analytics, it provides insightful information on how these fields work together to produce more accurate and thorough market forecasts.

II. Dataset Description and Preparation

A. Data Exploration

The datasets used in this project is tweets 'tweets_data' dataset in Kaggle [11] and 'BTC-USD' dataset from YahooFinance [9]. The 'tweets_data' dataset has 76797 entries and 8 attributes and these attributes consist of token, date, reply_count, like_count, retweet_count, quote_count, sentiment_label, sentiment_score. These tweets were collected between 2022-01-01 and 2023-06-22. The 'sentiment_label' attributes has "Positive", "Neutral" and "Negative" features.

The 'BTC-USD' data has 538 entries and 7 attributes and these attributes consist of Date, Open, High, Low, Close, Adj Close, Volume information of Bitcoin between 2022-01-01 and 2023-06-22.

B. Data Preprocessing

Strong data preprocessing was essential in our study in order to extract valuable insights from the combination of sentiment in social media and cryptocurrency market patterns. To begin with, we carefully combined two main datasets: one with tweet data and the other with Bitcoin (BTC) price history. Extensive data cleansing required transforming date columns into a standard datetime format, which is necessary for accurate temporal alignment. We were then able to create correlations between sentiment labels on tweets and the closing values of BTC by combining these datasets. In this project, we only focus on the sentiment scores instead of getting meanings from each word in a tweet. We calculated daily detailed average sentiment scores per sentiment label using group-by operations, which helped to clarify subtle sentiment trends. After that we wanted to calculate the average sentiment, minimum sentiment and maximum sentiment score to correlate them with tweet count. By multiplying the average, minimum and maximum sentiment score for each sentiment label and date by the total number of tweets, the code determines the weighted sentiment. This is an important stage since it prioritizes the feelings with more tweets.

TABLE I
EXAMPLE DATA WITH WEIGHTED AVERAGE SENTIMENT SCORE

date	average_Negative	average_Neutral	average_Positive	Close
2022-01-01	10.88	34.11	48.16	47686.81
2022-01-02	4.45	29.26	19.19	47345.22
2022-01-03	7.38	24.23	26.02	46458.12
2022-01-04	6.52	18.63	19.85	45897.57
2022-01-05	8.48	28.22	18.80	43569.00

TABLE II
EXAMPLE DATA WITH WEIGHTED MINIMUM SENTIMENT SCORE

date	min_Negative	min_Neutral	min_Positive	Close
2022-01-01	7.32	22.11	29.29	47686.81
2022-01-02	3.08	19.70	12.18	47345.22
2022-01-03	5.09	18.11	19.18	46458.12
2022-01-04	4.74	13.54	12.97	45897.57
2022-01-05	5.72	19.70	13.10	43569.00

TABLE III
EXAMPLE DATA WITH WEIGHTED MAXIMUM SENTIMENT SCORE

date	max_Negative	max_Neutral	max_Positive	Close
2022-01-01	13.92	40.83	53.40	47686.81
2022-01-02	5.43	37.13	24.64	47345.22
2022-01-03	10.13	33.13	28.70	46458.12
2022-01-04	8.31	24.72	24.67	45897.57
2022-01-05	11.54	36.98	25.26	43569.00

After the data preprocessing, we wanted to show the patterns or trends in the various variables (columns) in a time-series dataset. Plotting the designated columns in distinct subplots makes it simple to compare the trends among various variables. (Figure 1, 2 and 3)

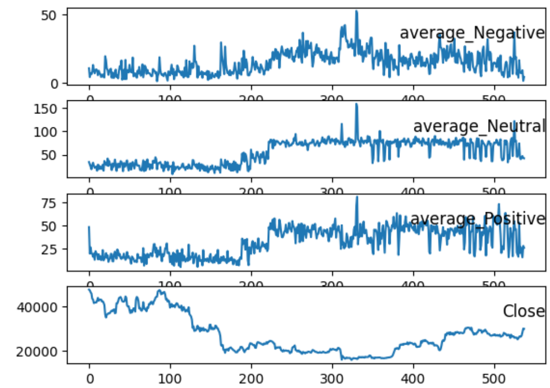

Fig. 1. Feature Comparison Weighted Average

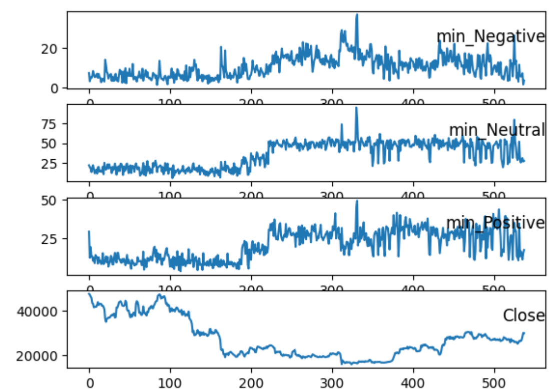

Fig. 2. Feature Comparison Weighted Minimum

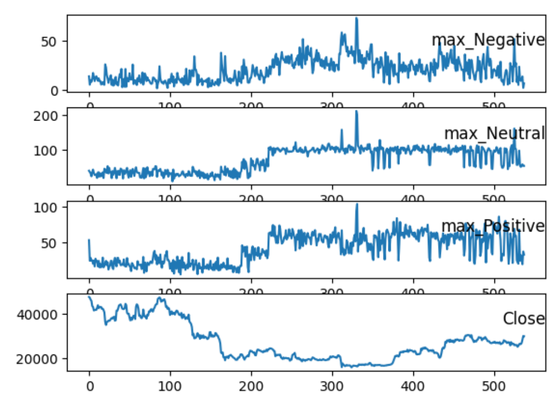

Fig. 3. Feature Comparison Weighted Maximum

III. METHODS

A. Software

This study depends on several significant Python libraries, whose dependencies contribute uniquely in the analysis of

Bitcoin price prediction. Pandas [6] and NumPy [7] are fundamental for facilitating data manipulation and numerical operations to enable one to ease in handling time series data efficiently. The matplotlib [5] is of fundamental importance in the visualization of data trends as well as results that are necessary in interpreting the performance of a model. Scikit-learn [4] aids in different types of machine learning tasks that include data preprocessing as well as evaluation of a model. At the heart of this predictive modeling in the research is TensorFlow's [8] Keras [10] API specifically its Sequential model and its Long Short-Term Memory (LSTM) layers that are instrumental in creating and developing deep learning models for time series prediction. Besides, the Keras libraries Dense and Dropout layers help with model architecture while the preprocessing, model selection, and metrics modules in sklearn aid data formatting and measurement. The libraries help one another, and together, they come out with one toolkit for elaborate machine learning analysis.

B. Multivariate LSTM Model

A Multivariate Long Short-Term Memory (LSTM) network is a Recurrent Neural Network (RNN) designed specially to handle and predict based on more than one input variable. These multivariate LSTMs can be seen as an expansion of the ordinary LSTMs. While the latter is primarily used for sequence prediction problems based on simple data with each time step having a single feature at that time (such as stock prices for each trading day going back a few years or more), the former is capable of handling complex situations where each time step may in fact accommodate multiple dimensions or features such as not only stock price but also trading volume, interest rates, and other economic factors. Multivariate LSTMs come in very handy where it's necessary to keep in view the relation of different input features with each other and also the sequence nature of the data. For instance, such would be forecasting of financial time series, where a large variety of economic indicators take part and are taken into consideration simultaneously.

C. Metrics

In order to assess the efficacy of the Multivariate LSTM (Long Short-Term Memory) model used to predict Bitcoin prices, we used several metrics.

1) Mean Absolute Error (MAE): It's simply the average of the absolute errors between predicted and actual values. This straightforward metric provides a sense of how far the predictions deviate from the true values on average.

2) Mean Absolute Percentage Error (MAPE): MAPE is a measure that describes the error in percentages with respect to the actual values. It serves as an indicator to understand the extent of errors in comparison to the scale of the data.

3) Root Mean Squared Error (RMSE): This is the square root of the MSE. Like MSE, it penalizes larger errors more heavily yet is more interpretable if the original data's units are used.

4) R-Squared (R^2 Score): This metric shows the proportion of variance in the dependent variable that can be explained through the independent variables. It indicates how closely the data points align with the fitted regression line.

IV. Evaluation

The data is scaled using the MinMaxScaler from scikit-learn, which makes sure that each feature contributes proportionately to the final forecast. To convert the data into a format appropriate for time series forecasting, the function create_dataset is developed; it generates data point sequences according to a given time step. After this implementation, the dataset is split into training and test sets and 45% of the dataset is set aside for testing.

An LSTM model was defined following the splitting of the training and test sets. A 50 unit LSTM layer, a dropout layer for regularization, and a dense layer for output included in the structure. The Adam optimizer and mean squared error loss function are used to construct the model. With a batch size of 1, the LSTM model is trained for 15 epochs on the training set.

For predictions, the test set used and the MinMaxScaler's inverse tranformation is used to return the predictions to their initial scale. After that we used matplotlib to visualize the actual and predicted values. In our project, because of the complication of the plotting results, it's decided to apply smoothing techniques so that the complication is eliminated. The forecasts are smoothed out by applying a simple moving average to reduce short-term volatility. The smoothing window size is set to 11.

For this project we have implemented 3 multivariate LSTM models which are Weighted Average Sentiment Score Multivariate LSTM Model, Weighted Minimum Sentiment Score Multivariate LSTM Model and Weighted Maximum Sentiment Score Multivariate LSTM Model as it is explained in 4.A, 4.B and 4.C sections.

TABLE IV
Model Comparison

	MAE	MAPE	RMSE	R^2 Score
Weighted Average LSTM	4269.84	19.27	5905.88	-0.69
Weighted Average Smoothed LSTM	3019.81	13.88	3855.71	0.28
Weighted Min LSTM	4398.88	19.88	5942.72	-0.71
Weighted Min Smoothed LSTM	3101.96	14.25	3947.91	0.25
Weighted Max LSTM	4329.05	19.50	5891.03	-0.68
Weighted Max Smoothed LSTM	3098.63	14.18	3932.36	0.25

A. Implementation of Weighted Average Sentiment Score Multivariate LSTM Model

The result of MAE (4569.84) indicates that the model's predictions differ from the real value by 4329.05 units. The smoothed MAE (3019.81), we can see that it is decreased after smoothing, suggesting that the smoothed forecasts are closer to the actual values.

MAPE (19.27%) result indicates that there is 19.27% relative difference between the model's predictions and the actual value. On the other hand, smoothed version of MAPE (13.88%) improved in terms of percentage error.

The result of RMSE (5905.88) shows that errors are spread more widely and the smoothed RMSE (3855.71) indicates that the model's predictions closer to the actual values and significant errors are less common.

R^2 Score (-0.69) result is negative and a negative R^2 Score indicates that the model is not able to adequately represent the data's trend. When we examine the smoothed R^2 Score (0.28), we can see it increased and and it shows that more variance of the original data has been accounted for by smooth's predictions.

We can see that this multivariate LSTM model with using weighted average sentiment score, demonstrates the best results rather than minimum and maximum sentiment scores

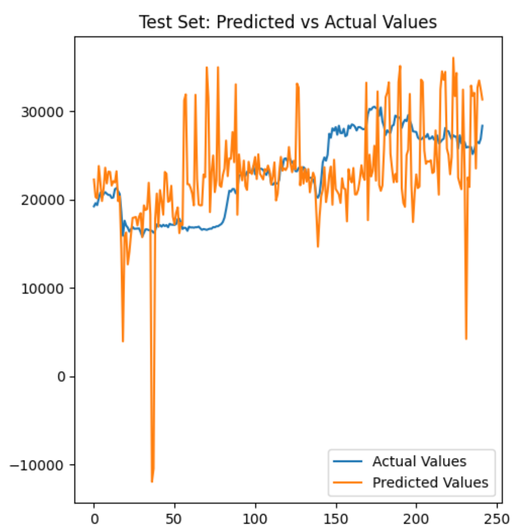

Fig. 4. Test Set: Predicted vs Actual Values Weighted Average Data

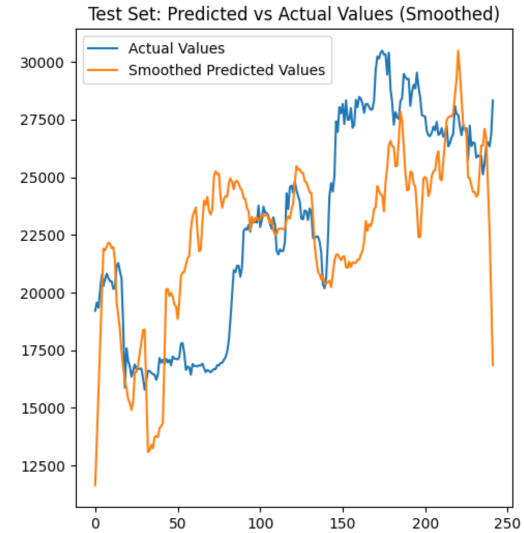

Fig. 5. Test Set: Predicted vs Actual Values Weighted Average Data (Smoothed)

B. Implementation of Weighted Minimum Sentiment Score Multivariate LSTM Model

The result of MAE (4398.88) indicates that the model's predictions differ from the real value by 4398.88 units. The smoothed MAE (3101.96), we can see that it is decreased after smoothing, suggesting that the smoothed forecasts are closer to the actual values.

MAPE (19.88%) result indicates that there is 19.88% relative difference between the model's predictions and the actual value. On the other hand, smoothed version of MAPE (14.25%) improved in terms of percentage error.

The result of RMSE (5942.72) shows that errors are spread more widely and the smoothed RMSE (3947.91) indicates that the model's predictions closer to the actual values and significant errors are less common.

R^2 Score (-0.71) result is negative and a negative R^2 Score indicates that the model is not able to adequately represent the data's trend. When we examine the smoothed R^2 Score (0.25), we can see it increased and and it shows that more variance of the original data has been accounted for by smooth's predictions.

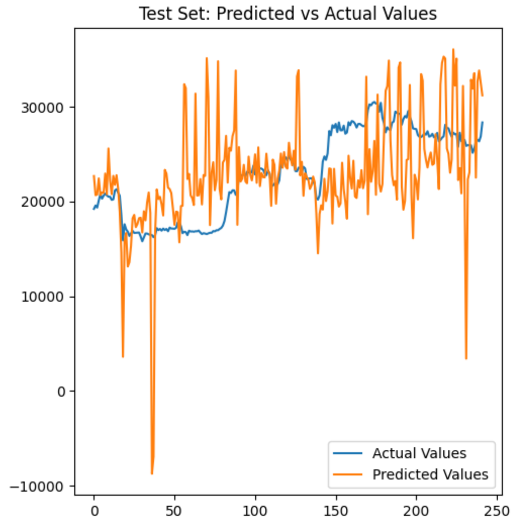

Fig. 6. Test Set: Predicted vs Actual Values Weighted Minimum Data

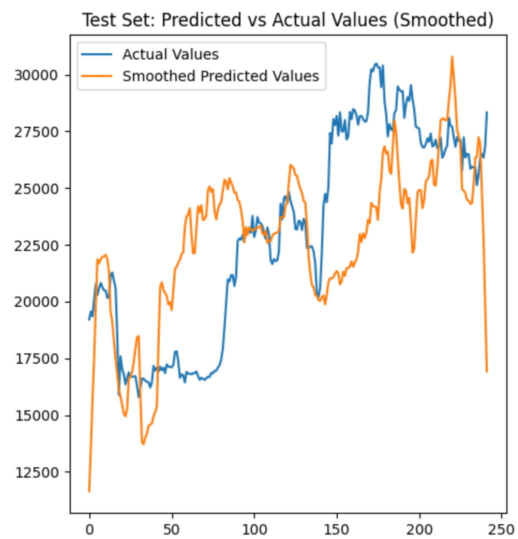

Fig. 7. Test Set: Predicted vs Actual Values Weighted Minimum Data (Smoothed)

C. Implementation of Weighted Maximum Sentiment Score Multivariate LSTM Model

In this model, we calculated the maximum sentiment score and multiply them with the tweet count according to their date and sentiment label, and used the data for the LSTM model that we created. We showed the regular MAE, MAPE, RMSE and R^2 Score results and then we demonstrated the smoothed versions.

The result of MAE (4329.05) indicates that the model's predictions differ from the real value by 4329.05 units. The smoothed MAE (3098.63), we can see that it is decreased after smoothing, suggesting that the smoothed forecasts are closer to the actual values.

MAPE (19.50%) result indicates that there is 19.50% relative difference between the model's predictions and the actual value. On the other hand, smoothed version of MAPE (14.18%) improved in terms of percentage error.

The result of RMSE (5891.03) shows that errors are spread more widely and the smoothed RMSE (3932.36) indicates that the model's predictions closer to the actual values and significant errors are less common.

R^2 Score (-0.68) result is negative and a negative R^2 Score indicates that the model is not able to adequately represent the data's trend. When we examine the smoothed R^2 Score (0.25), we can see it increased and and it shows that more variance of the original data has been accounted for by smooth's predictions.

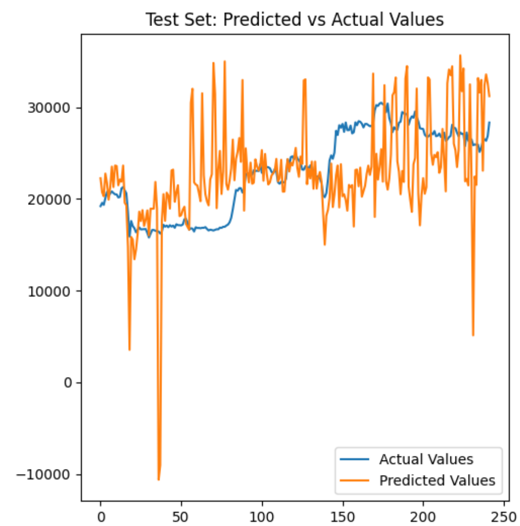

Fig. 8. Test Set: Predicted vs Actual Values Weighted Maximum Data

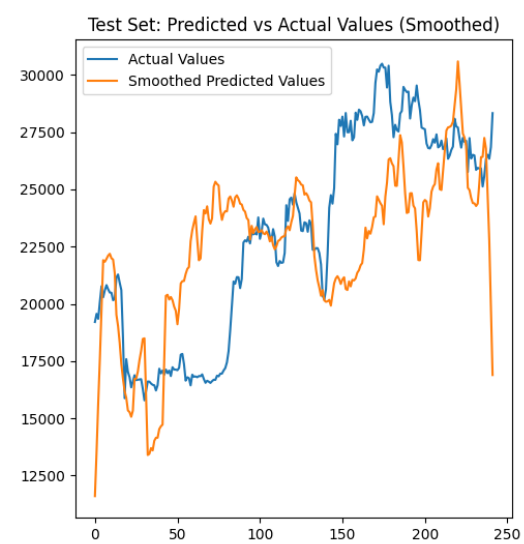

Fig. 9. Test Set: Predicted vs Actual Values for Weighted Maximum Data (Smoothed)

D. Conclusion

In this project titled "Crypto-currencies Price Prediction Using News and Social Networks Data," a Multivariate LSTM was used to predict the Bitcoin price[12], combining various sources of data. The bitcoin price[13] time series sourced from YahooFinance was combined with the sentiment analysis of social media data, obtained on Kaggle that emphasized public opinion influence over financial markets. This technique involved examining daily sentiments labels such as negative, neutral and positive in their frequencies to a certain level of understanding cryptocurrency pricing through various mechanisms.

The model performance was assessed using MAE, MAPE, RMSE and R2 score as the multiple metrics. The subtle predominance by the Weighted Average LSTM model in comparison to the weighted minimum and maximum models as shown graphically, accentuates its efficiency and depict that there is a complex pattern of relationship between different variables.

Future improvements might include larger set of data or consideration both more complicated and simple models such as transformers architecture or GRU in order to make higher accuracy, flexibility. It also implied a willingness to innovate since the model could integrate various aspects in new products.

By and large, the project effectively combines finance, computer science, and psychology revealing interesting information about crypto currency markets dynamics as well as recognition of machine learning in financial forecasting.

E. Future Work

We obtained that the average, minimum and maximum sentiment scores correlated to tweet counts give the result of a more optimal prediction rather than main sentiment scores. For more optimal results, the other features for instance tweet likes counts, re-tweet counts, comment counts, the account reliability, volume and some other features could be considered to be used. The model parameteres can be changed so that model complexity can increase or decrease. Thus, better results can be achieved. In addition, Transformer models or GRU might be used since Transformer models are effective for modeling complex time series data relationships and GRU's have time efficient structures. As an other future work, we are planning to extend the size of the dataset so model can learn better. Additionally, backtesting operations can be made since it helps us to understand the effectiveness of a model on historical data. These methods are planned to be used in the future work.

REFERENCES

[1] Kim YB, Kim JG, Kim W, Im JH, Kim TH, Kang SJ, Kim CH. Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies. PLoS One. 2016 Aug 17;11(8):e0161197. doi: 10.1371/journal.pone.0161197. PMID: 27533113; PMCID: PMC4988639.

[2] M. Briere, K. Oosterlinck, and A. Szafarz, "Virtual currency, tangible ' return: Portfolio diversification with bitcoins," Tangible Return: Portfolio Diversification with Bitcoins (September 12, 2013), 2013.

[3] McNally, S., Roche, J., & Caton, S. (2023). Predicting the Price of Bitcoin Using Machine Learning. In Proceedings of the 26th Euromicro International Conference on Parallel Distributed and Network-Based Processing.

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[5] Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95.

[6] McKinney, W. (2011). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 51–56.

[7] Oliphant, T. E. (2006). Guide to NumPy: Python for scientific computing. Trelgol Publishing.

[8] Abadi, M., Agarwal, A., Barham, P., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. arXiv preprint arXiv:1603.04467. or TensorFlow. (n.d.). TensorFlow: An open-source machine learning framework for everyone. Retrieved from https://www.tensorflow.org/

[9] Yahoo Finance. (2024, January 16). Apple Inc. (AAPL) Historical Data. https://finance.yahoo.com/quote/AAPL/history/

[10] Chollet, F. et al. (2015). Keras: The Python Deep Learning library. Retrieved from https://keras.io/

[11] Kapadnis, S. (2023). Bitcoin Tweets. Kaggle. https://www.kaggle.com/datasets/sujaykapadnis/bitcoin-tweets

[12] Ji S, Kim J, Im H. A Comparative Study of Bitcoin Price Prediction Using Deep Learning. Mathematics. 2019; 7(10):898. https://doi.org/10.3390/math7100898

[13] A. M. Balfagih and V. Keselj, "Evaluating Sentiment C1assifiers for Bitcoin Tweets in Price Prediction Task," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 5499-5506, doi: 10.1109/BigData47090.2019.9006140.

[14] N. Aslam, F. Rustam, E. Lee, P. B. Washington and I. Ashraf, "Sentiment Analysis and Emotion Detection on Cryptocurrency Related Tweets Using Ensemble LSTM-GRU Model," in IEEE Access, vol. 10, pp. 39313-39324, 2022, doi: 10.1109/ACCESS.2022.3165621.

[15] R. Albariqi and E. Winarko, "Prediction of Bitcoin Price Change using Neural Networks," 2020 International Conference on Smart Technology and Applications (ICoSTA), Surabaya, Indonesia, 2020, pp. 1-4, doi: 10.1109/ICoSTA48221.2020.1570610936.

[16] S. McNally, J. Roche and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning," 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Cambridge, UK, 2018, pp. 339-343, doi: 10.1109/PDP2018.2018.00060.