Bitcoin Signal Prediction via Machine Learning: Integrating Multi-Crypto and Technical Indicators

Zhengri Cui*, Tianrui Zhou*, Pengyang Yu*, Haoquan Wang*
*School of Computing, Dublin City University

***Abstract*—This study investigates Bitcoin signal prediction using Ethereum (ETH) and Ripple (XRP) OHLCV data, integrated with Bitcoin-related technical indicators, on an hourly time interval. In the experimental process, a baseline was established using an LSTM univariate model based solely on Bitcoin signal values. The results demonstrated that integrating Ethereum (ETH), Ripple (XRP), and technical indicators allowed both Logistic Regression and SVC(Support Vector Classifier) to achieve results comparable to the baseline model. Furthermore, XGBoost outperformed the LSTM in both prediction accuracy and F1 score. These findings suggest that technical indicators and other cryptocurrencies have potential roles in predicting Bitcoin's price movement signals.**

I. Introduction

Bitcoin has demonstrated unprecedented growth. Our exploratory analysis reveals that a \$100 investment in Bitcoin in November 2014 would yield over \$16,700 by 2024, outperforming traditional major stocks with a substantial return of 1,182%—the second-best performance, as shown in Fig1. However, its extreme volatility complicates price prediction, necessitating robust analytical frameworks. The interplay between major altcoins (e.g., Ethereum, Ripple) and technical indicators, as used by real-world traders to generate signals and make decisions, has not been fully explored—an aspect this study aims to address.

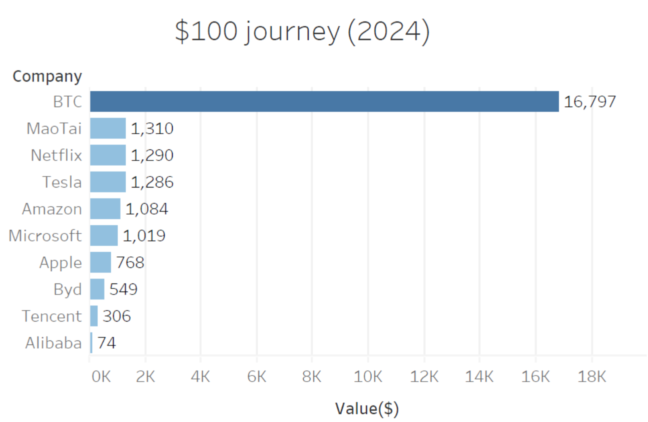

Fig. 1. Data sourced from Yahoo Finance for the period between November 26, 2014, to November 26, 2024. All values are standardized in U.S. dollars and calculated based on the annual average.

II. Related Work

After selecting this topic, we began by exploring how other researchers approached the problem by searching relevant projects on Kaggle. In many cases, we found that models were trained to predict the next closing price of Bitcoin using its historical closing prices. The predictions often aligned almost perfectly with the actual values in visualizations, which initially gave us the illusion that such models could be directly applied to real-world trading. However, as we delved deeper into the subject, we realized that predicting price using price alone is more about tracking the trend rather than truly understanding or forecasting the market dynamics in financial industry.

A GitHub user, upathare1, applied an LSTM model to predict the closing prices of the S&P 500 ETF (SPY) in order to illustrate that predicting prices based solely on historical price data may not be effective [1]. The predicted values closely overlapped with the actual prices in visualizations. However, when evaluated based on returns, the model achieved an accuracy of only 53.5%. In *Deep Learning for Finance*, the authors consistently highlight the critical importance of stationarity in financial data analysis [2].

In most studies, LSTM models have demonstrated superior performance. LSTM has emerged has a popular architecture when it comes to forecasting times series data, especially in financial stock markets and cryptocurrency markets. It has been proven to outperform various models such as Auto-regressive integrated moving average (ARIMA) and Transformer Models [3], [4]. They demonstrated this by using financial time series from January 1985 to August 2018 on a monthly basis into both an ARIMA model and LSTM model. Root-Mean-Square Error(RMSE) was then used to evaluate both the models and assess the accuracy of both predictions. The paper concluded that the LSTM-based algorithm improved predictions by 85% on average compared to ARIMA. In addition to this, [3] compared LSTM with a standard Transformer model by using Hewlett Packward Stock data (1962-2022) on both models under identical conditions. The study was evaluated using mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE)– in which LSTM achieved MAE of 0.072, MSE of 0.011, and RMSE of 0.104, outperforming the transformer, which had bigger error rates with MAE of 0.138, MSE of 0.054, and RMSE of 0.0232. These papers suggest that LSTM is more effective with capturing long term dependencies in time series data. An ensemble-based LSTM model was proposed in 2021, for forecasting Bitcoin price, integrating data from multiple time scales [5]. This model used data points from the minute, hourly and daily trading intervals of Bitcoin. The study made separate models for each of these times intervals and trained the LSTM models independently on its respective time scale. Once the model was trained, the predictions outcomes from these models are combined using techniques like weighted averaging and stacking. This enhances the models ability to predict across various market conditions. While this method improved prediction accuracy by leveraging multiple time scales, it would lead to an increase in computational cost due to the training required for each LSTM model and then combining these predictions together using weighted averaging and stacking techniques. [6] explores the use of LSTM in classification of cryptocurrency price by incorporating technical, trading and social media indicators. The study showed by leveraging these features the model could outperform models solely relying on technical indications alone. This combination of sentiment derived from social media platforms and historical price movements increased the accuracy of the models by 12% [6]. This study highlighted the potentially critical role that multi-dimensional data integration can play when enhancing the predictive power of LSTM models. Advanced architectures such as Attention-based LSTMs and Hybrid Models have been explored in recent years and have shown potential [7]. The paper demonstrated how combining both technical indicators and sentiment analysis lead to improved prediction accuracy, it also highlighted the difficulty of effectively pre-processing and weighting all the different types of data.

And more recently, sentiment-based prediction methods using large language models (LLMs) have become increasingly popular. While such approaches have been mainly applied to stock market forecasting [8][9], they are also considered to have strong potential for predicting cryptocurrency markets, including Bitcoin. In particular, Lopez-Lira and Tang (2023) utilized ChatGPT-4 to analyze news headlines and predict stock price movements without any specialized training. Their method resulted in a cumulative return of over 650% between October 2021 and December 2023[10].

Traditional machine learning approaches have primarily used RandomForest algorithms. Chen's study utilized variables including Bitcoin data, technical features of Bitcoin, other cryptocurrencies, commodities, market indices, foreign exchange, public attention, and dummy variables for the week. By comparing RandomForest and LSTM, the results showed that RandomForest outperformed LSTM in terms of performance. Additionally, Chen emphasized that using too many independent variables in the LSTM model can actually degrade its predictive performance[11]. In the work in [12], an impressive result was achieved, with a buy/sell signal accuracy exceeding 92% by utilizing technical indicators. They performed feature selection using the Chi-square distribution after engineering the technical indicators. However, most technical indicators related to Bitcoin are numerical data, and significant multicollinearity exists among them. The Chi-square distribution is generally suitable for categorical feature selection, and in the presence of multicollinearity, feature importance can become distorted. Therefore, we raise concerns about the methodology presented in this paper. Nonetheless, we believe that predicting direction using technical indicators is still a valuable approach to explore.

III. Methodology

All experiments were conducted using Google Colaboratory's high-RAM runtime mode to handle the computational demands of the LSTM training process and to perform grid search for optimal hyperparameter tuning of the other models.

For the implementation of the model and the dataset used in this study are available on GitHub. The repository can be accessed at: GitHub repository.

A. Data Collection

This study uses the Bitstamp API to capture hourly OHLCV (opening price, high price, low price, closing price, trading volume) data of Bitcoin (BTC), Ethereum (ETH) and Ripple (XRP) against the US dollar, covering a three-year market cycle from February 2022 to February 2025. This period covers several key market nodes, including the downward cycle of the crypto market and the latest round of rebound, providing a good foundation for studying price dynamics and asset linkage at different stages. In terms of data size, the dataset of each cryptocurrency contains 26,281 hourly records, which is equivalent to the continuous accumulation of 24 data points per day in the past three years, providing stable and detailed time series information support for subsequent analysis and modeling.

Subsequently, we aligned and merged the data of the three cryptocurrencies based on the precise timestamp of each record. The main purpose of this merging operation is to construct the possible price linkage or trend transmission relationship between BTC, ETH and XRP at the same time point. Finally, we store the organized unified data set as a single time series file so that it can be efficiently read and uniformly processed in the subsequent feature engineering and model training stages. This not only helps to fully mine cross-currency information, but also ensures the continuity and integrity of the data in time, laying a solid foundation for building a more expressive prediction model.

B. Exploratory Data Analysis

We first conducted exploratory data analysis on the merged Bitcoin data to verify data quality, observe basic distribution characteristics, and confirm the degree of correlation between variables, providing clear guidance for subsequent feature engineering and model building.

1) Data Overview and Distribution: First, we visually compared the hourly data captured by the Bitstamp API with the daily closing price data provided by Yahoo Finance. It can be clearly seen from the line chart that the trends of the two are almost exactly the same, and the closing prices at most time points are highly overlapped.

Next, we conducted a correlation heat map analysis on the opening price, highest price, lowest price, closing price, trading volume of BTC and the volume-weighted average price of ETH and XRP. The results show that the four basic

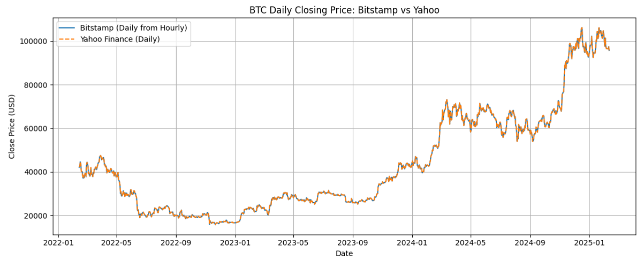

Fig. 2. Comparison of BTC Daily Closing Prices from Bitstamp (Aggregated Hourly) and Yahoo Finance

price fields of BTC (opening, closing, highest, and lowest) are highly positively correlated, and the correlation coefficient is close to 1.0, indicating that the information of these fields is highly redundant, so only one or two representative indicators can be retained when launching subsequent modeling.

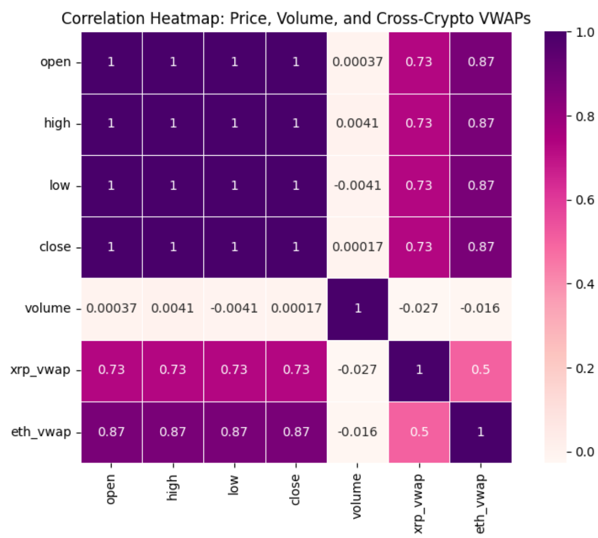

Fig. 3. Correlation heatmap between BTC close price, XRP_VWAP, and ETH_VWAP.

In addition, the hourly trading volume of BTC has a very low correlation with the price field, which means that short-term trading volume fluctuations are not very helpful in predicting price trends and need to be further converted or used with caution. The VWAP features of ETH and XRP are significantly positively correlated with BTC prices, with the correlation between ETH and BTC prices reaching 0.87 and XRP also reaching 0.73. This shows that cross-currency data has a high reference value for predicting Bitcoin price trends, and should be given priority in incorporating model inputs in the feature engineering stage.

2) Target variable analysis: This study sets the price increase or decrease direction of the next hour as the target variable, where "1" means that the closing price of Bitcoin in the next hour is higher than the current hour, and "0" means that the closing price is lower than the current hour. By drawing the distribution graph of the target variable, we observed that the proportion of the two categories of "up" and "down" is roughly the same, indicating that the data does not show obvious imbalance at the classification level, which allows the subsequent model to be directly evaluated using common indicators such as accuracy and F1 value without additional oversampling or undersampling operations.

3) Key conclusions: Combining the previous analysis, the following three conclusions can be drawn: the Bitstamp API merged data we retrieved is highly consistent with the public and reliable data source Yahoo Finance, the data quality is good and it is suitable for subsequent analysis; secondly, the basic price feature has high redundancy, and it may be necessary to streamline the input variables by constructing difference or amplitude indicators in the feature engineering stage; finally, the VWAP data of Ethereum and Ripple have obvious predictive potential, and the cross-currency linkage information can be introduced in the subsequent modeling to improve the prediction performance.

C. Data Preprocessing

Upon examining the data collected via the Bitstamp API, we identified duplicated entries for XRP, ETH, and BTC at the timestamp 2024-12-07 03:00:00, which matched the values from exactly one hour earlier. Since the same duplication occurred across all datasets at the identical timestamp, we infer that this was likely due to MCAR (Missing Completely At Random) missingness, for which the server automatically imputed the previous hour's data. To address this, we replaced the duplicated values at that timestamp with the average values of the respective day.

D. Feature Engineering

a.Bollinger Bands Related Indicators:

- **BB_ma (Middle Band)**: Simple moving average (SMA) of the closing price over a window of n periods. Used as the centerline for Bollinger Bands.

$$BB\_ma = SMA_n(Close)$$

- **BB_upper (Upper Band)**: Indicates the upper boundary of price movement.

$$BB\_upper = BB\_ma + k \cdot \sigma$$

  where k is a constant as a Multiplier for the standard deviation (typically $k = 2$), and σ is the standard deviation of the closing prices over the past n periods.

- **BB_lower (Lower Band)**: Indicates the lower boundary of price movement.

$$BB\_lower = BB\_ma - k \cdot \sigma$$

- **BB_bandwidth**: Measures the width of the Bollinger Bands, representing market volatility.

$$BB\_bandwidth = \frac{BB\_upper - BB\_lower}{BB\_ma}$$

- **BB_%b**: Indicates the position of the current closing price within the Bollinger Bands.

$$\%b = \frac{Close - BB\_lower}{BB\_upper - BB\_lower}$$

A value near 1 suggests the price is close to the upper band; near 0 suggests closeness to the lower band.It can Judge whether the price is at a relatively high/low level.

b. Volatility and Momentum Indicators:

- **ATR (Average True Range)**: Measures market volatility based on price ranges.

$$\text{TR}_t = \max\big(High_t - Low_t, |High_t - Close_{t-1}|, |Low_t - Close_{t-1}|\big)$$

$$ATR_t = EMA_n(TR)$$

where TR is the True Range, and EMA_n is the exponential moving average over n periods.

- **CCI (Commodity Channel Index)**: Identifies whether the asset is overbought or oversold.

$$CCI = \frac{TP - SMA_n(TP)}{0.015 \cdot MeanDeviation}$$

where TP (Typical Price) = $\frac{High+Low+Close}{3}$.

- **RSI14 / RSI30 (Relative Strength Index)**: Measures momentum by comparing average gains and losses.

$$RSI = 100 - \frac{100}{1+RS}, \quad RS = \frac{AverageGain}{AverageLoss}$$

RSI is typically calculated over 14 or 30 periods. RSI ¿ 70 indicates overbought, ¡ 30 oversold.

- **low-high**: Measures daily price range, often used as a volatility proxy.

$$low\_high = High - Low$$

- **open-close**: Measures net price change during the day.

$$open\_close = Close - Open$$

c. Trend Indicators:

- **MACD (Moving Average Convergence Divergence)**: Difference between short-term and long-term exponential moving averages of the closing price.

$$MACD = EMA_{12}(Close) - EMA_{26}(Close)$$

It measures the deviation between short-term trend and long-term trend and can be regarded as a "speedometer" for the direction and intensity of the trend.

- **Signal**: 9-period exponential moving average of the MACD line, used to generate trading signals.

$$Signal = EMA_9(MACD)$$

It can smooth the signal line of MACD and capture the buying and selling points. When MACD crosses Signal $\rightarrow$ it may be a buy signal. When MACD goes below Signal $\rightarrow$ it may be a sell signal.

- **EMA10 / EMA30**: Exponential Moving Average of closing prices over 10 or 30 periods.

$$EMA_t = \alpha \cdot Close_t + (1-\alpha)\cdot EMA_{t-1}, \quad \alpha = \frac{2}{n+1}$$

They reflect the recent and medium-term price trends respectively. EMA10 is more sensitive and suitable for capturing short-term changes. EMA30 is smoother and suitable for identifying medium-term trends.

- **VWAP (Volume Weighted Average Price)**: VWAP reflects the average price of an asset, weighted by volume, over a specific period. It gives a more accurate representation of the average transaction price compared to the simple average.

$$VWAP = \frac{\sum_{i=1}^{n}(P_i \cdot V_i)}{\sum_{i=1}^{n} V_i}$$

where P_i is the price at time i (e.g., close price), and V_i is the trading volume at that time.
VWAP is often used as a benchmark to determine whether the current price is high or low. If the price is above the VWAP, it indicates a upward trend. If the price is below the VWAP, it indicates a downward trend.

d: Dimensionality Reduction with PCA

In our experiment, we used both Logistic Regression and SVM models, which can be affected by multicollinearity. To address this, we performed correlation analysis, and the results are shown in Fig 4. The analysis revealed that there were very high multicollinearity issues between certain variables. This could have a detrimental impact on the prediction performance of the Logistic Regression and SVM models. To address this, we performed correlation analysis, and the results are shown below. The analysis revealed that there were very high multicollinearity issues between certain variables. This could have a detrimental impact on the prediction performance of the Logistic Regression and SVM models. Therefore, we extracted variables with high correlations and applied Principal Component Analysis for dimensionality reduction. The highly correlated variables were transformed into PCA1 and PCA2, and the visualization of the results is shown in Fig 5. Additionally, PCA1 covered 90.5% of the information from the original features with multicollinearity, and PCA2 covered 7.4% of the information. We performed dimensionality reduction using Principal Component Analysis (PCA) on the following features: `xrp_vwap`, `eth_vwap`, `bb_ma`, `bb_upper`, `bb_lower`, `ema_10`, and `ema_30`. The first two principal components, denoted as `PCA1` and `PCA2`, were retained for use in the model.

The actual features used in each model are as in table I:

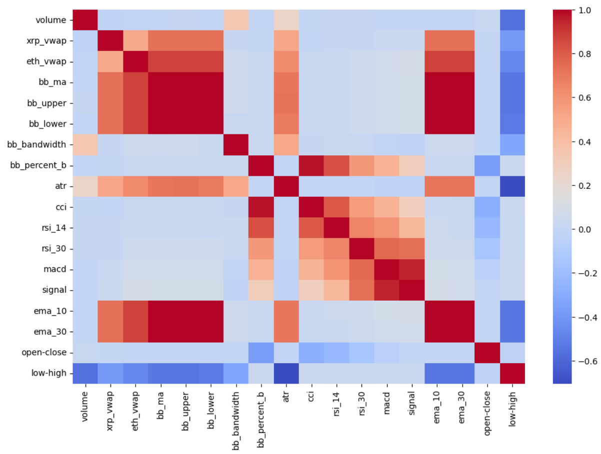

Fig. 4. Correlation heatmap among all variables

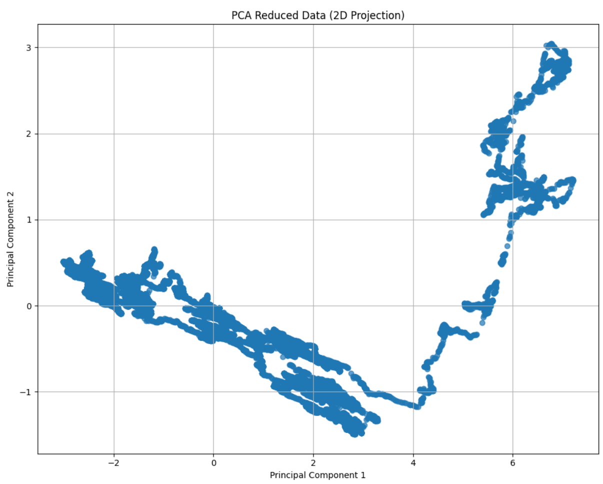

Fig. 5. Visualization of PCA1 with PCA2

E. Modeling

The entire dataset was split in accordance with the characteristics of time series data, in order to facilitate comparison. 80% of the data was allocated to the training set, and 20% to the test set, with 20% of the training set being further used as a validation set for each model. The class distribution in both the training and test sets was found to be balanced. This information is shown in the Fig 6.

1) BaseLine Model - LSTM
As highlighted in numerous previous studies, the Long Short-Term Memory (LSTM) model has consistently demonstrated strong performance in Bitcoin price prediction tasks. In this study, we established a univariate LSTM model as the baseline, focusing solely on hourly upward (1) and downward (0) signals. The input sequence length was varied between 2 and 10 for experimentation. Considering the importance of temporal order in time series data, the dataset was split into training and testing sets with an 80:20 ratio. From the training data, 20% was further set aside as a validation set without shuffling, in order to preserve the sequential nature of the data.
The model architecture is as follows. The first LSTM layer was tested with node sizes of 40, 50, 60, 70, and 80, respectively. To prevent overfitting, a dropout rate of 20% was applied. The second layer was a Dense layer with 60 nodes, using the Rectified Linear Unit (ReLU) as the activation function. The final output layer consisted of a single node with a Sigmoid activation function, appropriate for binary classification. The model was optimized using the Adam optimizer, with learning rates of 0.01, 0.005, 0.002, and 0.001 tested. To ensure sufficient training, the number of epochs was set to 2000. ModelCheckpoint was employed to save the best-performing model parameters, and EarlyStopping was implemented with a patience of 50 to halt training if no improvement in validation loss was observed. The training process is illustrated in Fig7.

2) Logistic Regression Algorithm
We implemented a Logistic Regression model to predict the signal of Bitcoin movements. Since logistic regression is a linear model that is sensitive to multicollinearity, we applied Principal Component Analysis to reduce the dimensionality of features with high collinearity. However, to prevent potential information loss, PCA was applied only to a subset of the features, while the remaining variables were retained in their original form. We then adjusted the overall feature set so that the Variance Inflation Factor (VIF) for all variables remained below 10, thereby enhancing both the stability and interpretability of the model. Prior to model training, all features were standardized using a Standard Scaler

TABLE I
FEATURES USED FOR EACH MODEL

Model	Features Used
LSTM (Baseline Model)	Price signals (increase:1, decrease:0)
Logistic Regression/SVM	ATR, RSI14/RSI30, Low-High, CCI, PCA1, MACD, BB_bandwidth, Open-Close, PCA2
XGBoost	XRP_VWAP, ETH_VWAP, BB_Ma, BB_Upper, BB_Lower, BB_Bandwidth, BB_%b, ATR, CCI, RSI14/RSI30, MACD, Signal, EMA10, Open-Close, Low-High, Price_Change

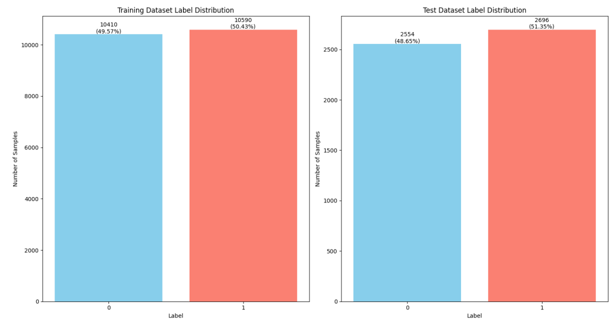

Fig. 6. Distribution of the target class in the training and test datasets

Fig. 7. The visualization of the training and validation loss of the LSTM model.

to ensure they are on the same scale.

We conducted hyperparameter tuning using GridSearchCV, exploring:

- Regularization types: L1 (Lasso), L2 (Ridge), and ElasticNet penalties
- Regularization strength: Inverse parameter C values [0.01, 0.1, 1, 10, 100]
- Optimization algorithms: 'liblinear' and 'saga' solvers
- L1 ratio of 0.5 for ElasticNet regularization

The optimal parameters identified were L1 regularization with C=0.1 using the 'liblinear' solver. Model performance was evaluated using accuracy, precision, and recall on both training and test sets. The logistic regression model achieved comparable performance to our baseline models, with test accuracy of 54.1%, demonstrating that even this simple linear model could capture some predictive patterns in the Bitcoin price movements.

3) Support Vector Machine Algorithm

In order to establish a classical machine learning benchmark for Bitcoin movement predictions, we developed a Support Vector Machine (SVM) model using both linear and non-linear kernels. The SVM in this study was trained to determine the direction of hourly Bitcoin prices, either upwards (1) or downwards (0). We also use features like RSI, BB_bandwidth and others in this model. These features provide a concise view of the short-term behavior of the market by capturing both daily volatility and absolute price levels. Before model training, all inputs were standardised using Standardscaler in order to achieve optimal SVM performance.

Two model variants were calculated: one using a Radial Basis Function (RBF) kernel, with hyperparameters tuned via `RandomizedSearchCV`, and the other using a linear kernel via `LinearSVC`. The parameter space of the RBF kernel included a regularization term C ranging from 0.1 to 2. To maintain computational efficiency, the randomized search was limited to 10 iterations with 3-fold cross-validation. The dataset was split using an 80:20 ratio for training and testing, and 20% of the training data was allocated to validation.

The model performance was assessed using accuracy, precision, and recall on both the training and testing sets. Both SVM variants achieved accuracies in the range of 53% to 56%, with the non-linear SVM model demonstrating a slight advantage in recall. These SVM models serve as a comparison with more complex architectures such as LSTM model and XGBoost model.

4) XGBoost Algorithm

In the XGBoost model, all predefined technical indicators were included through feature engineering, and autoregressive elements were also considered. Specifically, the ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) were analyzed, revealing a significant spike at lag 1. Based on this, the previous price change (`previous_diff`) was added as an independent variable. Although multicollinearity may exist between the technical indicators, this was not considered an issue in this study, as XGBoost was used as a predictive model rather than a feature selection tool. Ultimately, 17 variables were fed into the XGBoost model, as listed in Table I, then Grid Search was conducted using the following parameter ranges:

- `max_depth`: [3, 4, 5]
- `learning_rate`: [0.001, 0.01]
- `n_estimators`: [300, 500, 700]
- `subsample`: [0.6, 0.8, 1.0]
- `colsample_bytree`: [0.6, 0.8]

The model was trained to minimize the *log loss* metric. The best hyperparameter combination identified through grid search was:

- `colsample_bytree`: 0.6
- `learning_rate`: 0.01
- `max_depth`: 3
- `n_estimators`: 300
- `subsample`: 0.8

IV. Evaluation/Results

TABLE II
Model Performance Comparison on Training and Test Sets

	Training Set		Test Set	
	Accuracy	F1-Score	Accuracy	F1-Score
LSTM(baseline)	0.55	0.55	0.54	0.54
Logistic Regression	0.53	0.53	0.54	0.53
SVC	0.56	0.55	0.53	0.50
XGBoost	0.58	0.58	0.56	0.56

In this study, we conducted a comparative analysis of four models—LSTM, logistic regression, SVC, and XGBoost—to predict Bitcoin increase and decrease signals. As shown in Table II, the XGBoost model outperformed the others on both the training set (accuracy: 0.58, F1-score: 0.58) and the test set (accuracy: 0.56, F1-score: 0.56). Notably, XGBoost was trained using all features—including technical indicators, XRP/ETH VWAP, and the previous day's price change—without dimensionality reduction, suggesting that it can effectively leverage information even in high-dimensional feature spaces.

However, due to a high degree of multicollinearity among these features, caution must be taken when interpreting feature importance derived from the model.

Meanwhile, the baseline LSTM model, which utilized only univariate signal data as input, showed relatively consistent performance (accuracy: 0.55 on the training set and 0.54 on the test set), but demonstrated limited predictive power. This highlights the difficulty of capturing complex cryptocurrency market patterns using only a single signal time series.

The logistic regression model, using PCA-reduced technical indicators and XRP/ETH VWAP features, exhibited stable performance (accuracy between 0.53 and 0.54), though no significant improvement was observed. Nonetheless, the fact that its performance was comparable to that of the LSTM model, which has been well-regarded for its strong performance in time series prediction, implies that the feature engineering process adopted in this study had a positive impact even on the simpler logistic regression model.

On the other hand, the SVC model was tuned to closely match the baseline performance on the training set, but experienced a notable performance drop on the test set with a potential issue of overfitting.

V. Conclusion

Overall, the relatively strong performance of XGBoost suggests that tree-based models can be effective in capturing the complex and nonlinear relationships present in the cryptocurrency market. However, the F1-score of approximately 0.56 on the test set remains insufficient for direct application to trading strategies. This limitation reflects the inherent challenges of forecasting Bitcoin price movements, given their high volatility and susceptibility to various external factors.

VI. Discussion

In this experiment, the LSTM model was trained using univariate time series data based solely on Bitcoin signals. However, it is anticipated that incorporating the technical indicators applied in this study into the model would lead to improved performance. Additionally, considering more comprehensive factors, such as news and sentiment analysis scores, could further enhance the model's predictive performance.

One key insight gained during this project is that, in real markets, the strategic judgment of when to buy and when to sell is more crucial than repeatedly running the model to detect signals. Specifically, while this experiment was conducted using a one-hour interval, in actual financial markets, capturing key turning points in the market might prove to be more effective than continuously running prediction models. Future research should consider integrating these perspectives, aiming to improve the model by detecting turning points in market trends.

References

[1] upathare1, "Bitcoin price prediction using lstm," https://github.com/upathare1/BTC-LSTM, 2020, mIT License.

[2] S. Kaabar, *Deep Learning for Finance*. O'Reilly, 2024.

[3] I. Sonata and Y. Heryadi, "Comparison of lstm and transformer for time series data forecasting," in *2024 7th International Conference on Informatics and Computational Sciences (ICICoS)*, 2024, pp. 491–495, available at: https://ieeexplore.ieee.org/abstract/document/10636892.

[4] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401, available at: https://ieeexplore.ieee.org/abstract/document/8614252.

[5] M. Shin, D. Mohaisen, and J. Kim, "Bitcoin price forecasting via ensemble-based lstm deep learning networks," in *2021 International Conference on Information Networking (ICOIN)*, 2021, pp. 603–608, available at: https://ieeexplore.ieee.org/abstract/document/9333853.

[6] M. Ortu, N. Uras, C. Conversano, S. Bartolucci, and G. Destefanis, "On technical trading and social media indicators for cryptocurrency price classification through deep learning," *Expert Systems with Applications*, vol. 198, p. 116804, 03 2022.

[7] F. Liu, Y. Li, B. Li, J. Li, and H. Xie, "Bitcoin transaction strategy construction based on deep reinforcement learning," *Applied Soft Computing*, vol. 113, p. 107952, 2021, available at: https://www.sciencedirect.com/science/article/pii/S1568494621008747. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621008747

[8] C. Liu, A. Arulappan, R. Naha, A. Mahanti, J. Kamruzzaman, and I.-H. Ra, "Large language models and sentiment analysis in financial markets: A review, datasets, and case study," *IEEE Access*, vol. 12, pp. 134 041–134 061, 2024, available at: https://ieeexplore.ieee.org/abstract/document/10638546.

[9] M. Pelster and J. Val, "Can chatgpt assist in picking stocks?" *Finance Research Letters*, vol. 59, p. 104786, 2024, available at: https://www.sciencedirect.com/science/article/pii/S1544612323011583. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1544612323011583

[10] A. Lopez-Lira and Y. Tang, "Can chatgpt forecast stock price movements? return predictability and large language models," *SSRN*, 2023, available at: http://dx.doi.org/10.2139/ssrn.4412788.

[11] J. Chen, "Analysis of bitcoin price prediction using machine learning," *Risk and Financial Management*, 2023, available at: https://doi.org/10.3390/jrfm16010051.

[12] L. K. M. E. M. A. S. Abdelatif Hafid, Mohamed Rahouti, "Predicting bitcoin market trends with enhanced technical indicator integration and classification models," 2024, available at: https://arxiv.org/abs/2410.06935.