Surfing the Bitcoin Waves: Comprehensive Trend Forecasting with Various Trader Types

[1,] Can Ali Ateş, canaliatess@gmail.com
[1,] Emre Çoban, emrecobann02@gmail.com
[1, 2,] Tugba Gurgen Erdogan, tugba@cs.hacettepe.edu.tr

[1] Hacettepe University, Computer Engineering Department, Artificial Engineering, Ankara, Turkey
[2] Hacettepe University, Computer Engineering Department, Software Engineering Research Group, Ankara, Turkey

Abstract

Cryptocurrency trading is becoming increasingly popular worldwide, with many individuals seeking to maximize their profits. One approach they're exploring is following the actions of successful investors, automated bots, or whale traders. The insights that can help traders make better decisions may be uncovered by analyzing their behavior and impact. This study examines the effectiveness of this strategy and seeks to understand how various types of traders impact the Bitcoin market, including fundamental aspects like price fluctuations over time. Additionally, we aim to identify patterns that regular traders can follow to enhance their chances of success in cryptocurrency trading. We employed a time-series forecasting method which involves analyzing past price movements and other critical factors to predict future trends. To ensure the robustness and reliability of the findings, we utilized various advanced techniques, such as machine learning, deep learning, and traditional time-series forecasting models. These powerful tools help us make more accurate predictions and provide strong evidence for our research conclusions. The study demonstrates that some models, such as Linear Regression and Random Forest Regression, didn't work well with features related to "whales", "bots", and "top traders". However, models like XGBoost Regression and Transformer showed positive effects. This suggests that, for now, traders should focus more on basic features like "open", "high", and "low" prices rather than these other factors. As advanced models like XGBoost and Transformer continue to develop, these features may become more important. While it is important to consider different features, relying on traditional indicators currently seems prudent.

1. INTRODUCTION

Cryptocurrencies like Bitcoin and Ethereum have shown remarkable price appreciation over the years, making early investors considerable profits. Additionally, the decentralized nature of cryptocurrencies enables individuals across the globe to participate in the financial markets without the need for traditional banking infrastructure. This democratization of finance allows for greater financial inclusion, particularly in regions where access to conventional banking services is limited or nonexistent. The 24/7 availability of cryptocurrency markets also provides traders with the flexibility to engage in trading activities at any time, accommodating different time zones and personal schedules.

Conversely, articles frequently point out volatility and regulatory uncertainties as major negative effects of cryptocurrency trading. The highly volatile nature of cryptocurrencies can lead to significant financial losses, sometimes wiping out substantial investments within a short period. This volatility is often driven by speculative trading, market manipulation, and news events, making it a high-risk endeavor for traders. Furthermore, the regulatory landscape for cryptocurrencies is still evolving, with many governments and financial institutions struggling to develop clear and consistent regulations. This regulatory uncertainty can create an unpredictable environment for traders, where sudden legal changes or crackdowns on exchanges can disrupt markets and negatively impact investments.

Additionally, the lack of robust consumer protections in many cryptocurrency markets exposes traders to risks of fraud, hacking, and theft.

Despite these negative effects, the opportunities for high returns continue to drive the growing popularity of cryptocurrency trading worldwide This popularity brings new questions to the minds of those hoping to become wealthy through cryptocurrency trading. One key motivation arising from these questions is the idea that "following whales (entities holding significant volumes of cryptocurrency) [1], bots (automated algorithms programmed to execute trades) [2], and top traders (highly influential participants) [3] can provide an effective trading to make more money".

In this study, we focus on the Bitcoin [4] market to investigate this popular idea by examining the classical factors such as "open", "high", and "low" prices, as well as specific factors like "stock market characteristics of whales, bot traders, and top traders". The research was conducted using time-series forecasting techniques, incorporating machine learning, deep learning, and traditional time-series forecasting models. This approach aimed to create a robust experimental environment to obtain trustworthy analyses and results about the effect of whales, bot traders, and top traders on the Bitcoin market. Additionally, this technique improved the interpretability of results and provided insights from the field of artificial intelligence into cryptocurrency trading.

2. RELATED WORK

2.1 Security of Bitcoin Market

Akba et al. [5] employed machine learning and statistical forecasting methods to detect manipulations in Bitcoin prices. The manipulation in the cryptocurrency market involves deceptive strategies aimed at artificially influencing the price of cryptocurrencies, such as sudden price movements unrelated to news, high trade volumes in short periods, pump-and-dump schemes, whale manipulation, spoofing, and insider trading. [6] Key methods included SVM, ARIMA, SARIMAX, and LSTM for price prediction, and sentiment analysis of social media posts to gauge their impact on prices. Anomalies indicating potential manipulations were identified using SVM epsilon SVR. The results showed that combining forecasting methods with sentiment analysis effectively detected manipulation periods, with SVM showing the highest effectiveness when integrated with sentiment analysis.

Liao et al. [7] investigate the security of cryptocurrencies, highlighting their deviation from traditional secure distributed systems frameworks. Instead of relying on established theoretical models, cryptocurrency security goes off the rails on the assumption of majority honest behavior among miners. The study introduces the concept of a "whale attack," where a minority attacker incentivizes rational miners to mine on a fraudulent fork, demonstrating the feasibility of such attacks and leaving open avenues for further exploration into accurately modeling attack costs and strategy spaces.

2.2 Whales & Trading Behaviours of Traders

Azamjon et al. [8] introduce a methodology for forecasting Bitcoin price movements by integrating on-chain data with real-time tweets from the @whale_alert Twitter account. The reinforcement learning and Q-learning techniques are leveraged to make predictions as increase, stable, or decrease. The methodology reached %90 accuracy in predicting the trend of Bitcoin.

Feng et al. [9] is a research study that looks at how people trade Bitcoin when they have important information before others do. We use data about individual transactions to see if there are any trends before big events in the cryptocurrency world. The study found that some people seem to trade Bitcoin with insider knowledge before big events, both good and bad, in the cryptocurrency world. This shows that top traders are making a lot of money by knowing things before others, which can be unfair to regular traders.

2.3 Bots in Cryptocurrency Trading

H. Jazayeriy and M. Daryani's study [10] produces a Smart Price-Action (SPA) algorithm for automated trading in cryptocurrencies, employing dynamic boundaries and a genetic algorithm (GA) for parameter optimization. Fibonacci numbers are used to streamline parameter optimization and suggest further research into alternative technical indicators for order confirmation, as well as exploring effective risk and liquidity management strategies tailored to the SPA algorithm. Experiments show an 81% win rate with less than 0.1 daily trades in the SPA algorithm for daily time frames, while 1-minute intervals show a 50% win rate with over 100 potential trade situations.

Yavuz et al. [11] aim to explore the impact of Bitcoin bots and how to regulate them, considering contrasting views on Bitcoin's disruptive potential and susceptibility to fraud. The study suggests that recent unusual trading activity, likely influenced by significant events, underscores the need for regulation. Such findings align with comprehensive checks conducted during market analysis, highlighting the importance of understanding and managing bot activity for maintaining stability in Bitcoin-related assets like futures and options.

The related works demonstrate that there are a minute amount of experimental studies done to observe the effect of whales, bots, and top traders in the field of cryptocurrency trading using the power of artificial intelligence. The contributions of the study that was carried out are the provide a point of view from the field of AI to investigate the effects of the various trader types over the Bitcoin stock market and investigate the effect of whales, bots, and top traders for cryptocurrency trading operations. In this work, eight different models are conducted for this purpose. The credentials of these models are as follows: Linear Regression, a fundamental machine learning model, examines relationships between variables to make predictions. Random Forests combine multiple decision trees for improved accuracy, while XGBoost refines predictions by focusing on areas of uncertainty. Time series forecasting models like SARIMAX and Prophet specialize in predicting future values based on historical patterns. Deep learning models such as LSTM-FCN and FCN leverage neural networks to capture intricate temporal dependencies, while Transformers excel in processing sequential data. These models offer diverse capabilities for analytical tasks and datasets.

3. METHODOLOGY

In this study, the methodology adopts a simplified version of the CRISP-DM model.

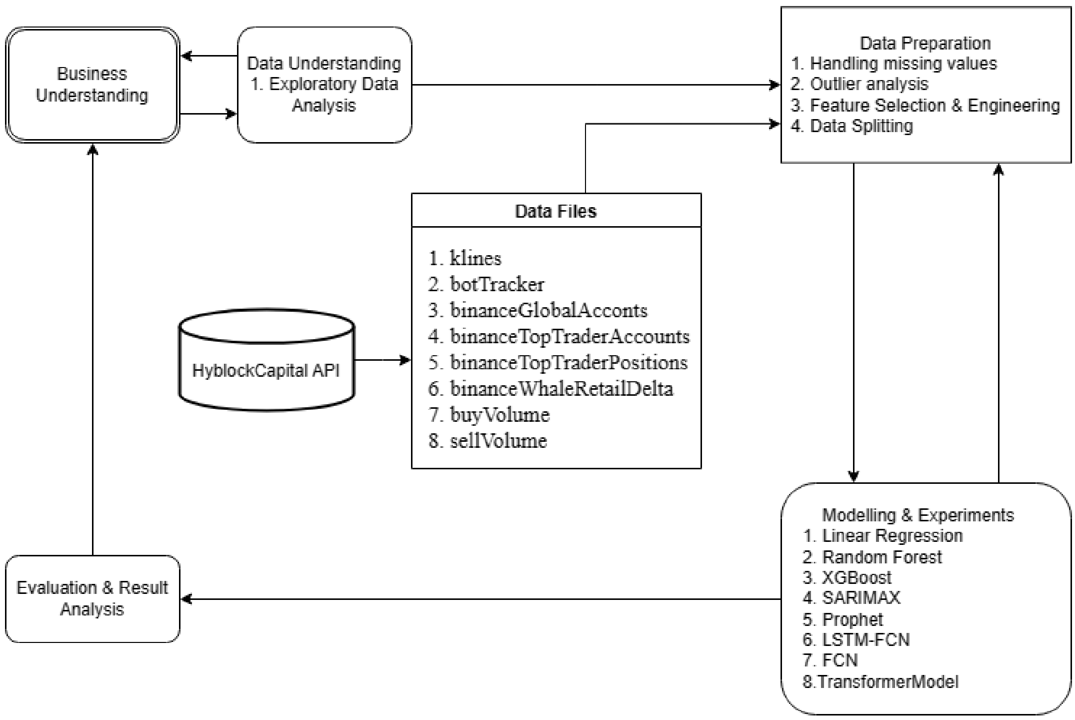


Figure 1: Adopted CRISP-DM Methodology

1) Business Understanding:

The goal of this project is to thoroughly analyze trends in Bitcoin by studying the behaviors of three key groups which are large holders of cryptocurrency (referred to as "whales"), influential participants in the stock market (widely recognized as “top traders”), and automated trading algorithms (commonly known as "bots").

The scope of the project contains several steps:

1. Collect the distributed data files from HyperCapital API endpoints.
2. Combine these data files into one data frame for ease of use.
3. Exploratory data analysis to gain insight into features regarding statistical summarization, instance count, and data types.
4. Apply preprocessing of the data with visualizations to handle missing values and outliers. Feature selection and engineering to reduce the feature count and increase the data quality.
5. Splitting the data into train, validation, and test subsets. Also, splits targets and features for all subsets.
6. Developing both machine learning and deep learning-based models for trend forecasting analysis of Bitcoin to get an idea about various traders' effects on the stock market.
7. Test and evaluate the models to get robust results about the research.

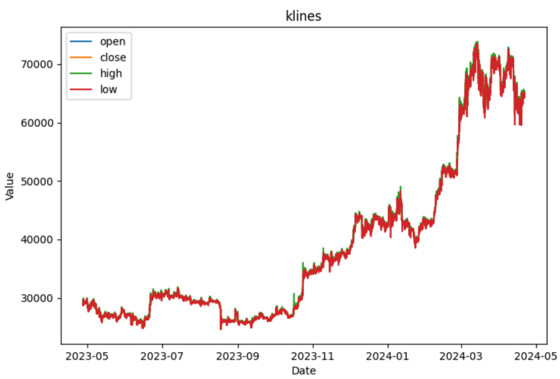
The main research questions that prompted us to conduct this study were:

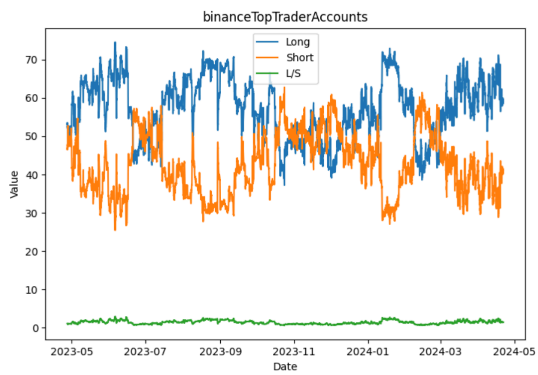
1. What key market indicators and trader behaviors influence Bitcoin price movements?
2. How can machine learning and deep learning models effectively forecast trends in the Bitcoin market?
3. How can the system accommodate different trader types and their risk preferences?

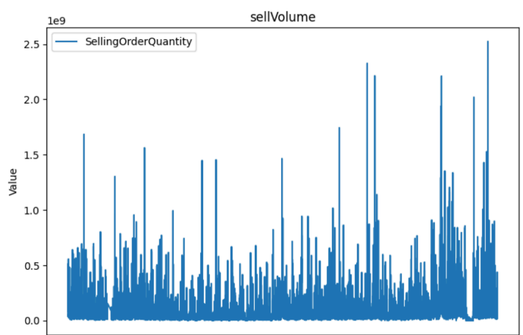
2) Data Understanding and Exploration:

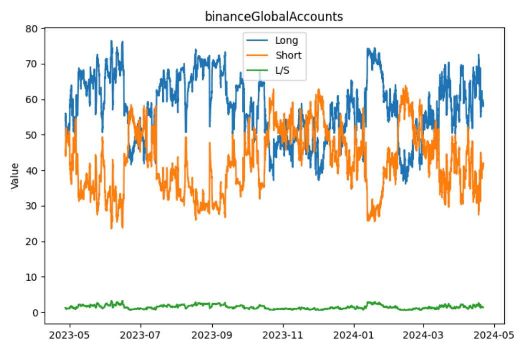
The study was conducted over 8 different data files that were time-stamped into 15 minutes. These data files were gathered from the HyBlock Capital API Endpoint, and all of these data files are combined into one data frame for ease of use and storage. Except for the date feature, all features must be in numeric format to be convenient for machine learning and deep learning models. The dataset can only be accessible from the API key. The data files that are used for this study are gathered from this API key endpoint. These data files are:

1. **klines:** The dataset has “Open”, “Close”, “High” and “Low” price features for the Bitcoin stock market.
2. **botTracker:** The dataset has an “estimatedBotCount” feature which indicates the prediction of bot activity based on the searching for frequently occurring unique order sizes.
3. **binanceGlobalAccounts:** The dataset has “Long”, “Short”, and “Ratio of Short and Long” features that represent the total number of accounts that are long or short strategy on Binance.
4. **binanceTopTraderAccounts:** The dataset has “Long”, “Short”, and “Ratio of Short and Long” features that represent the total number of top trader accounts (top %20) that are long or short strategies on Binance.
5. **binanceTopTraderPositions:** The dataset has “Long”, “Short”, and “Ratio of Short and Long” features that represent the total number of top traders positions (top %20) that are long or short strategy on Binance.
6. **binanceWhaleRetailDelta:** The dataset has a “WhaleRetailPositionDelta” feature that indicates the difference between the percentage of long positions held by top trader accounts ("whales") and the percentage of long positions held by global accounts ("retail").
7. **buyVolume:** The dataset has a “BuyingOrderQuantity” feature that indicates the quantity of buying orders executed during a specified period.
8. **sellVolume:** The dataset has a “SellingOrderQuantity” feature that indicates the quantity of selling orders executed during a specified period.









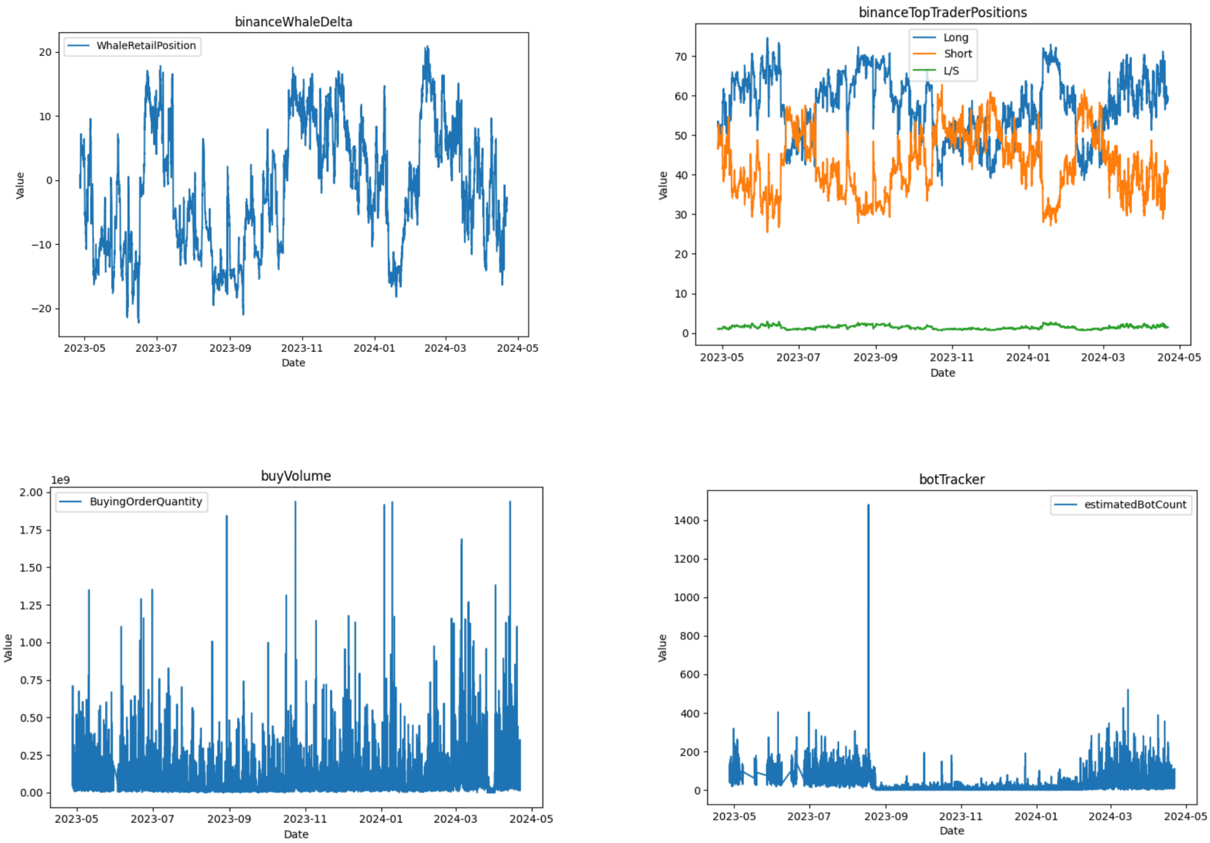


Figure 2: Visualization of data files

The data used in the study contains the classical and problem-specific features, so it can capture the problems that were investigated. Therefore, we selected this dataset to use in this study. The "close" feature is the most important because we are observing the effects of various trader types on this feature. On the other hand, the features gathered from binanceGlobalAccounts, binanceTopTraderAccounts, binanceTopTraderPositions, binanceWhaleRetailDelta data files are critical because we are investigating the effect of these features. The study was conducted for time series forecasting, this type of problem is a supervised problem and the used API contains the data that is labeled therefore we did not need to label any data.

These data files are combined into one data frame. The data frame is indexed with the Date feature and it has 34645 rows with 17 columns. Information on the data frame is demonstrated in Figure 3, and statistical summarization demonstrated in Figure 4.

#	Column	Non-Null Count	Dtype
0	open	34556 non-null	float64
1	close	34556 non-null	float64
2	high	34556 non-null	float64
3	low	34556 non-null	float64
4	estimatedBotCount	31376 non-null	float64
5	globalAccounts_Long	34535 non-null	float64
6	globalAccounts_Short	34535 non-null	float64
7	globalAccounts_LS	34535 non-null	float64
8	topTraderAccounts_Long	34532 non-null	float64
9	topTraderAccounts_Short	34532 non-null	float64
10	topTraderAccounts_LS	34532 non-null	float64
11	topTraderPositions_Long	34532 non-null	float64
12	topTraderPositions_Short	34532 non-null	float64
13	topTraderPositions_LS	34532 non-null	float64
14	WhaleRetailPosition	34516 non-null	float64
15	BuyingOrderQuantity	34185 non-null	float64
16	SellingOrderQuantity	34185 non-null	float64

Figure 3. Data Frame Information with column names, instance counts, and data types

	count	mean	std	min	25%	50%	75%	max
open	34556.0	3.870601e+04	1.389279e+04	24803.0000	2.747807e+04	3.384460e+04	4.366677e+04	7.369010e+04
close	34556.0	3.870723e+04	1.389374e+04	24804.1000	2.747805e+04	3.384505e+04	4.366707e+04	7.369010e+04
high	34556.0	3.877141e+04	1.393646e+04	24917.9000	2.752205e+04	3.391520e+04	4.372985e+04	7.388140e+04
low	34556.0	3.863796e+04	1.384659e+04	24581.0000	2.743295e+04	3.377965e+04	4.359468e+04	7.348000e+04
estimatedBotCount	31376.0	3.178120e+01	3.906006e+01	0.0000	4.000000e+00	1.500000e+01	5.025000e+01	1.481000e+03
globalAccounts_Long	34535.0	5.763517e+01	8.930339e+00	36.1500	5.071000e+01	5.773000e+01	6.512500e+01	7.649000e+01
globalAccounts_Short	34535.0	4.236483e+01	8.930339e+00	23.5100	3.487500e+01	4.227000e+01	4.929000e+01	6.385000e+01
globalAccounts_LS	34535.0	1.472252e+00	5.436524e-01	0.5662	1.028800e+00	1.365700e+00	1.867400e+00	3.253500e+00
topTraderAccounts_Long	34532.0	5.730141e+01	7.819026e+00	37.2300	5.098000e+01	5.768000e+01	6.354000e+01	7.456000e+01
topTraderAccounts_Short	34532.0	4.269859e+01	7.819026e+00	25.4400	3.646000e+01	4.232000e+01	4.902000e+01	6.277000e+01
topTraderAccounts_LS	34532.0	1.423772e+00	4.548787e-01	0.5931	1.040000e+00	1.362900e+00	1.742700e+00	2.930800e+00
topTraderPositions_Long	34532.0	5.730141e+01	7.819026e+00	37.2300	5.098000e+01	5.768000e+01	6.354000e+01	7.456000e+01
topTraderPositions_Short	34532.0	4.269859e+01	7.819026e+00	25.4400	3.646000e+01	4.232000e+01	4.902000e+01	6.277000e+01
topTraderPositions_LS	34532.0	1.423772e+00	4.548787e-01	0.5931	1.040000e+00	1.362900e+00	1.742700e+00	2.930800e+00
WhaleRetailPosition	34516.0	-1.342348e+00	9.697971e+00	-22.2700	-9.820000e+00	-1.800000e+00	6.532500e+00	2.091000e+01
BuyingOrderQuantity	34185.0	6.587411e+07	9.789401e+07	155491.9912	1.905564e+07	3.581914e+07	7.232488e+07	1.937422e+09
SellingOrderQuantity	34185.0	6.658235e+07	1.005621e+08	124912.7976	1.939795e+07	3.634217e+07	7.297891e+07	2.524511e+09

Figure 4. Statistical Summary of the Data Frame

3) Data Preparation:

Missing value analysis was conducted over each data file, at first there seemed to be no null values but after the merge operation of all data files into one combined data frame we observed there were some missing values for each feature.

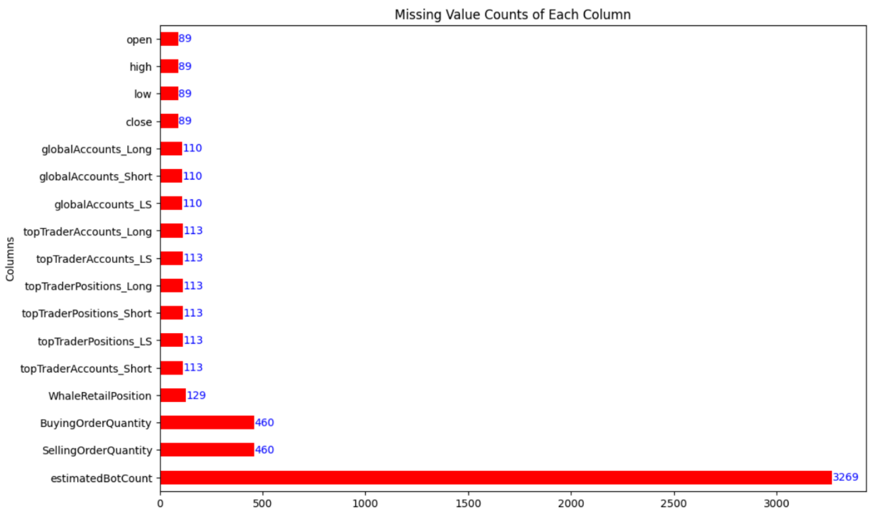


Figure 5. Count of missing values

Most of the missing values were from the feature “estimatedBotCount” around 3269 NaN values in 34645 data, as shown in Figure 5. Since The API returns NaN when the system does not detect any bot, therefore, we filled the NaN values with 0 in the estimatedBotCount.

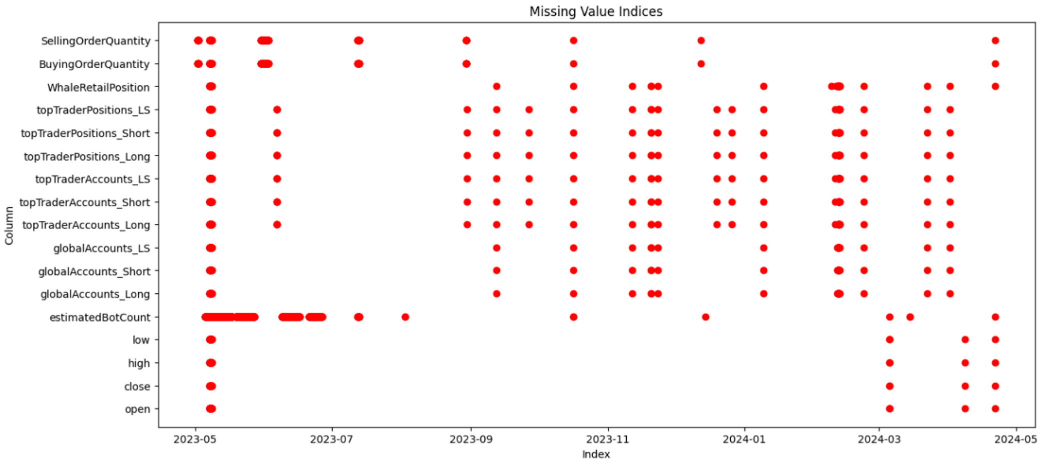


Figure 6. NaN Values Indices

Afterward, we visualized the missing indices of each feature to see if they were missed in common indices and we observed that at the end of the file, as demonstrated in Figure 6, there was a gap where almost all of them were missing we dropped this part which is around 89 NaN values coming each after each thus we decided to drop these NaN values. After this operation, there were only some minor missing parts from some of the features. We decided to use linear interpolation for these data points since we know there are not too many missing points that could lead to bad results afterward.

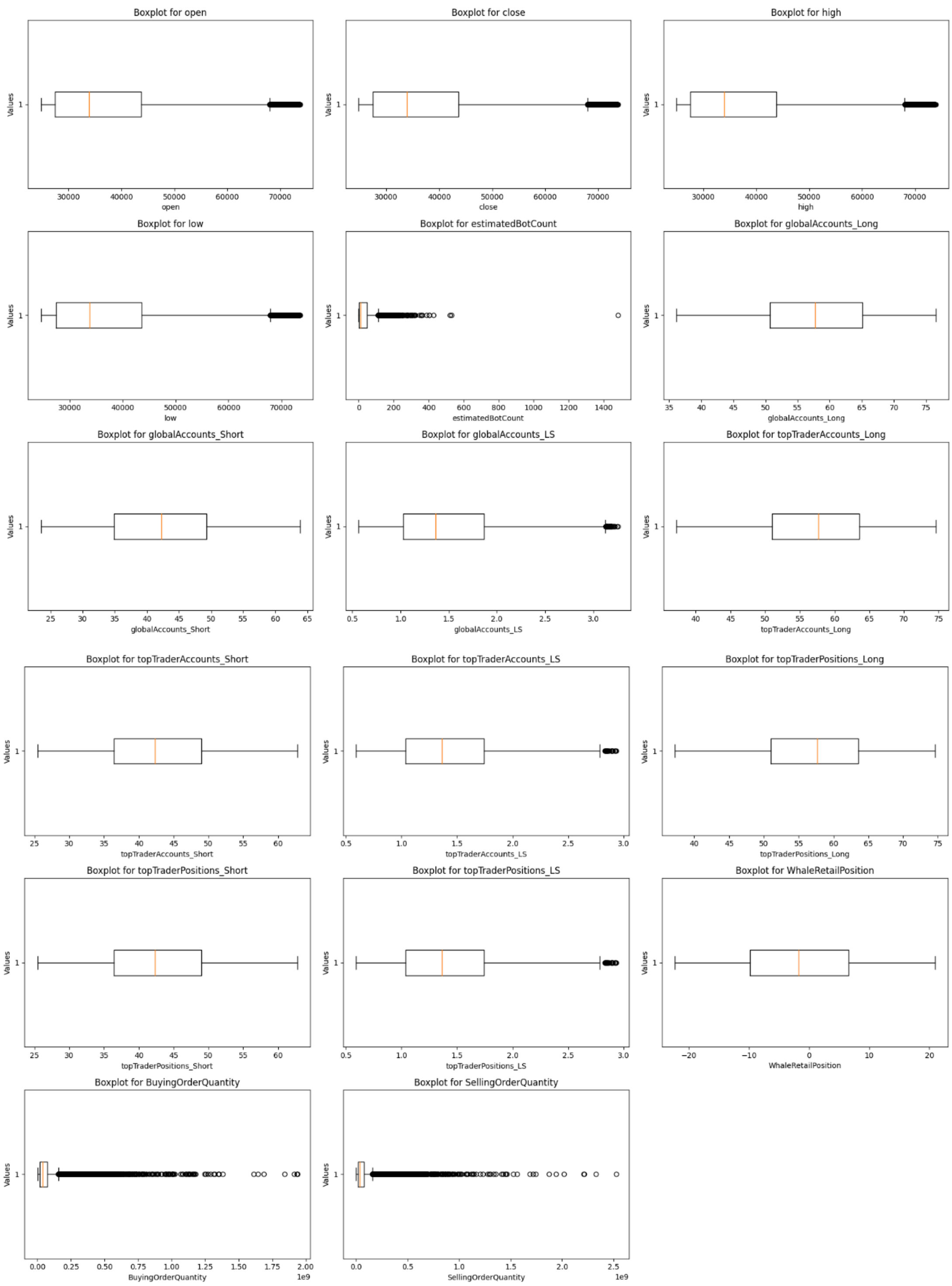


Figure 7. Boxplot Analysis for Outlier Detection

Outlier analysis was conducted using the boxplot figures, as we observed from the boxplots some of the features have outliers as shown in Figure 7, but the data used for the project is time-series so, these points can be extreme. However, these outliers provide valuable insights into extreme market conditions and trading behaviors, while respecting the inherent volatility of cryptocurrency markets. Therefore, the outliers will be protected instead of dropping or smoothing them.

After these preprocessing steps, we cleaned our data from missing values observed outliers, and decided to keep them for the sake of this project since we are dealing with extreme conditions in the real market. At the end in our hands, we have our data cleaned and ready to model.

We decided to drop features that are not necessary to our work. For example, there are "LongTopTraderPositions", "ShortTopTradePositions" and "TopTraderPositions_L/S" which is a long ratio over short. Since "TopTraderPositions_L/S" includes both information encoded, we decided to drop "Long" and "Short" features.

We split the data into train, validation, and test sets. Train data consists of %90 of data, validation %5 of data, and test %5 of data. The main reason behind this splitting strategy is that our data contains nearly 34K records so %5 of the whole can capture enough instances for a test or validation set. Therefore, we can use the majority of our data to train the models.

Since some features such as prices have high numbers compared to other features we scaled data by using the Min-Max scaling technique, so the model wouldn't give importance to one feature rather than all. Also, it made training faster.

4) Modelling and Experiments:

All project features, except the Date feature, are in numeric data type. Among the data files "binanceGlobalAccounts," "binanceTopTraderAccounts," and "binanceTopTraderPositions," the features "Long," "Short," and "Ratio of Short and Long" are present. Since the "Ratio of Short and Long" feature encapsulates the information of the other two, "Long" and "Short" features within these three data files are deemed redundant. The project utilizes the remaining features. The project will rely on various predictive models tailored to address different aspects of the Bitcoin stock market forecasting task. These models are

1. Machine Learning Models
   a. Linear Regression
   b. Random Forests Regression
   c. XGBoost Regression
2. Traditional Time-Series Forecasting Models
   a. SARIMAX Model
   b. Prophet
3. Deep Learning Models
   a. LSTM-FCN
   b. FCN Model
   c. Transformers

Since we are dealing with a forecasting problem we need to pick a metric meaningful for numerical regression such as Mean Squared Error, Mean Absolute Error, or Root Mean Squared Error. We trained our models using mean squared error but in the end, when we compared their performances on the test dataset we also wanted to see MAE, RMSE, and R2 score to evaluate their predictive performance across different error scales and better understand their robustness in real-world scenarios.

Experimental Setup

We have designed a sliding window approach to our dataset, which is generally done in time series forecasting problems. We did this only for Machine Learning and Deep Learning models since SARIMAX and Prophet don't need windowed data. We used window size 24, step size 1, and forecasting horizon 1 for all models.

Machine Learning Models

Sklearn's linear regression model is used at first to see how well a basic model predicts our data. As we can see from the evaluation part the model performed quite well becoming the best model according to RMSE.

Afterward, we used RandomForestRegressor but for this model, we used Optuna, a hyperparameter tuning framework, to get an understanding of optimal parameters and trained the model with those specified parameters (n_estimators=50, max_depth=25, min_samples_split=15, min_samples_leaf=10). This model also performed quite well becoming the 2nd best model according to RMSE.

In the end, we wanted to train with the XGBoost model which is known for beating Deep Learning models at some tasks. In our study, since we did not specifically focus on optimization this model performed well but not well as the other two ML models. Could be handled via optimization but we decided to move on to other models.

Traditional Time-Series Forecasting Models

For the SARIMAX model, we used the AutoArima library to determine the optimal order and seasonal order. We used p=2, d=1, q=4 as order, and for the seasonal order we used 0=P, 0=D, 0=Q, 0=m. The model seems to capture trends but there are offsets at some part resulting in a higher RMSE and making the model not so efficient for our study.

While training the Prophet model we log-transformed data instead of scaling it via StandardScaler since it is a commonly used technique for the Prophet model. The model has the same problem as SARIMAX seems to capture trends but too much offset resulting in a higher error rate.

Deep Learning Models

We trained 3 Deep Learning models for our study hoping to achieve a comparable good performance at forecasting. The first model was LSTM-FCN [12, 13], this model is implemented in most of the modern time series libraries as we observed even though a simple model performs quite well on benchmark datasets. We trained this model with 256 LSTM units and 128 convolution filters with kernel size 7. Trained model for 100 epochs but used early stopping with patience 21, used initial learning rate 1e-3 but again we used learning rate scheduler with patience 5, factor 0.25.

The second model FCN [14] is inspired by the LSTM-FCN model with similar architecture except it has no LSTM part and includes only convolution parts. Except it has 3 convolution blocks each block has 64, 128, and 256 filters respectively and each have 7, 5, and 3 kernel sizes respectively. Trained with the same number of epochs again using early stopping and LR-scheduler.

For the final model we have trained a TransformerModel [15], this model consists of a linear layer followed by a transformer encoder layer with 1 head 128 dimensions of feed-forward network, and 64 dimensions. At the end 0.1 probability dropout applied. This model seems to perform better with larger window sizes this is due to the ability of transformers to capture range dependence much better. However, in our study, we observed that the best-performing DL model is LSTM-FCN.

Effect Of The Extra Features

After these experiments, we decided to dive into the investigation of the effects of the features that we provided such as TopTraderPositionsL/S, whaleRetailDelta, etc. For this purpose, we trained the models without extra features that represent the "whale", "bots" and "top traders" characteristics to see the effect on the forecasting performance. The results obtained from these experiments are already shown in the "Evaluation and Result Analysis" part as tables.

5) Evaluation and Result Analysis:

Model Name	***RMSE***	***MAE***	***R2***
Linear Regression	0.0052	0.0034	0.9929
Random Forest	0.0063	0.0042	0.9898
XGBoost	0.0112	0.0082	0.9676
SARIMAX	0.0267	0.0228	0.8131
Prophet	1958.0558	1953.9824	-69.7868
LSTM-FCN	0.0075	0.0058	0.9855
FCN	0.0194	0.0141	0.9031
Transformer Model	0.0149	0.0115	0.94

Table 1: Models' all features included results

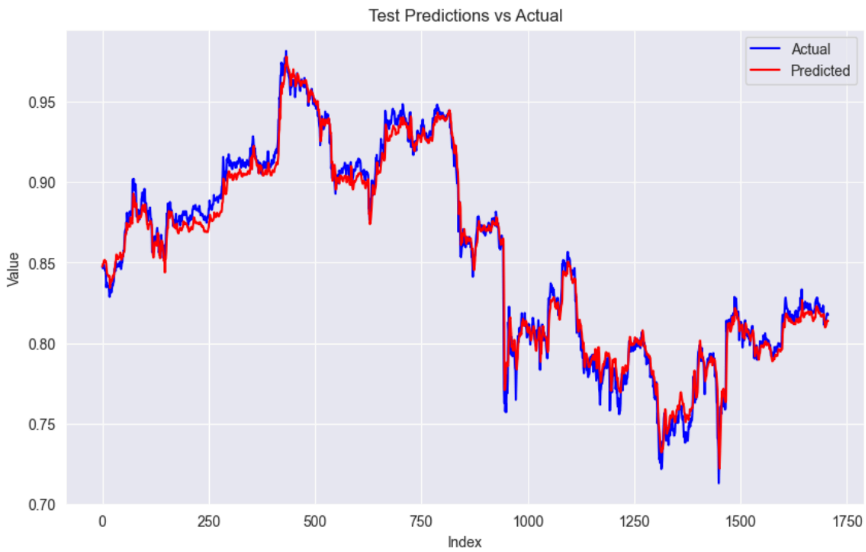


Figure 8. LSTM-FCN with all features included

Model Name	***RMSE***	***MAE***	***R2***
Linear Regression	0.0052	0.0034	0.9929
Random Forest	0.0062	0.0042	0.9901

XGBoost	0.0114	0.0084	0.9667
SARIMAX	0.0245	0.2090	0.8430
Prophet	1958.0558	96.2034	0.7997
LSTM-FCN	0.0069	0.0045	0.9878
FCN	0.0104	0.0089	0.9723
Transformer Model	0.0201	0.0173	0.8954

Table 2: Models' "whale", "bots" and "top traders" features are disabled results

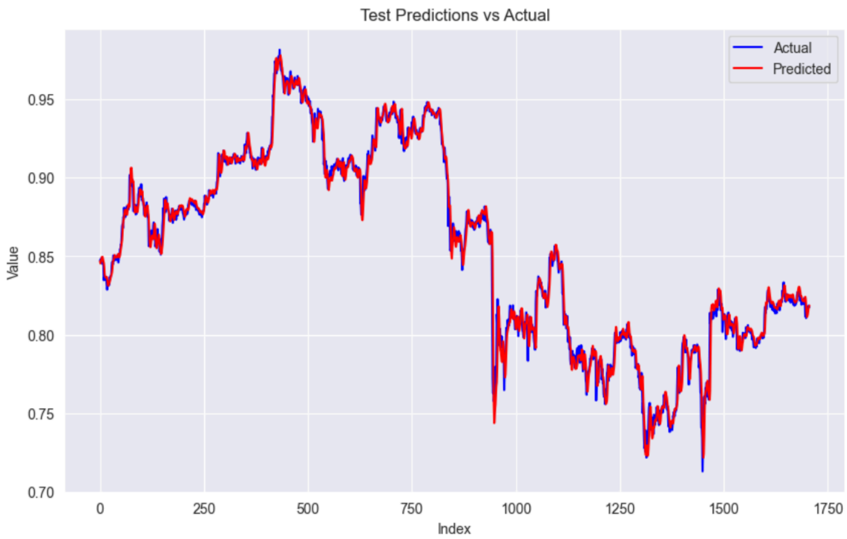


Figure 9. LSTM-FCN with "whale", "bots" and "top traders" features are disabled

Finally, we evaluate our results and analyze the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R2 Score of the models in the experiments. Our experimental results demonstrate that the 6 models (Linear Regression, Random Forest Regression, SARIMAX, Prophet, LSTM-FCN, and FCN) are affected negatively by the "whale", "bots" and "top traders" features while 2 models (XGBoost Regression, Transformer) are affected positively.
According to these results, in the field of artificial intelligence view the exogenous features which are coming from the "whale", "bots", and "top traders" are mostly ineffective for the Bitcoin stock market. On the other hand, to make a further analysis, with the increase in the capable model developments such as XGBoost and Transformer these "ineffective" features will gain importance beside the "open", "high", and "low". However, for now, the study suggests that traders give importance to feature engineering and classical features which are "open", "high", and "low".

4. CONCLUSION

In conclusion, our study investigated the intricate dynamics of the Bitcoin stock market, particularly focusing on the roles of whales, bots, and top traders alongside classical market indicators like open, high, and low prices. Through an extensive analysis employing machine learning, deep learning, and traditional time series forecasting models, we uncovered valuable insights into the effectiveness of these various trader types on Bitcoin price forecasting. Our findings indicate that while classical market indicators remain crucial, the inclusion of features representing whales, bots, and top traders did not consistently enhance forecasting performance across all models. Interestingly, models like XGBoost and Transformer demonstrated positive effects from these additional features, suggesting their potential importance in future model developments. As the field continues to evolve, likely, a nuanced understanding of both traditional market factors and emerging trader behaviors will likely become increasingly vital for accurate forecasting and informed decision-making. Ultimately, this study contributes to the ongoing discourse surrounding cryptocurrency trading, shedding light on the intricate interplay between market dynamics and trader behaviors while providing valuable insights for both researchers and practitioners in the field.

The study acknowledges two primary limitations and suggests avenues for future research. Firstly, it relies on static data gathered in a certain time interval, thereby precluding evaluation of real-time data streams. Addressing this, a proposed feature work involves testing the experimental setup on live data to yield more dynamic and applicable results. Secondly, due to computational constraints, not all models could be fully optimized. A subsequent endeavor will involve thorough optimization of all models across a broader parameter space, thereby enhancing the robustness and reliability of the study's findings. These limitations and suggested future works will make the study more robust and more clearly interpret the effect of various traders in the Bitcoin stock market.

5. REFERENCES

[1] "What is a Whale in Crypto Trading? (techopedia.com)"

[2] "What Are Crypto Trading Bots and How Do They Work? (coindesk.com)"

[3] "Who Are The Best Crypto Traders To Follow? (coinmarketcap.com)"

[4] Nakamoto, Satoshi. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System.

[5] F. Akba, I. T. Medeni, M. S. Guzel, and I. Askerzade, "Manipulator detection in cryptocurrency markets based on forecasting anomalies," *IEEE Access*, vol. 9, pp. 108819–108831, 2021, doi: 10.1109/ACCESS.2021.3101528.

[6] "What is market manipulation in cryptocurrency? (cointelegraph.com)"

[7] Liao, Kevin & Katz, Jonathan. (2017). Incentivizing Blockchain Forks via Whale Transactions. 264-279. 10.1007/978-3-319-70278-0_17.

[8] M. Azamjon, O. Sattarov and J. Cho, "Forecasting Bitcoin Volatility Through on-Chain and Whale-Alert Tweet Analysis Using the Q-Learning Algorithm," in IEEE Access, vol. 11, pp. 108092-108103, 2023,

[9] Feng, Wenjun & Wang, Yiming & Zhang, Zhengjun. (2017). Informed trading in the Bitcoin market. Finance Research Letters. 26. 10.1016/j.frl.2017.11.009.

[10] H. Jazayeriy and M. Daryani, "SPA Bot: Smart Price-Action Trading Bot for Cryptocurrency Market," 2021 12th International Conference on Information and Knowledge Technology (IKT), Babol, Iran, Islamic Republic of, 2021, pp. 178-182, doi: 10.1109/IKT54664.2021.9685662.

[11] Balcioglu, Yavuz & Özer, Gökhan. (2023). What Proportion of the Bitcoin Ecosystem Is Controlled by Bots? 10.1007/978-3-031-23416-3_3.

[12] https://arxiv.org/abs/1709.05206v1

[13] "https://github.com/titu1994/LSTM-FCN." Accessed: May 15, 2024. [Online]. Available: https://github.com/titu1994/LSTM-FCN

[14] "https://github.com/timeseriesAI/tsai/blob/main/tsai/models/FCN.py." Accessed: May 15, 2024. [Online]. Available: https://github.com/timeseriesAI/tsai/blob/main/tsai/models/FCN.py

[15] "TransformerModel ." Accessed: May 15, 2024. [Online]. Available: tsai/TransformerModel.py