ORIE 5741 FINAL REPORT

May 11, 2023

Analysis of on-chain data for Ethereum

Name: Benyuan He
Student ID: bh575

Name: Litong Liu
Student ID: ll936

Contents

I Abstract

This report presents two models for developing trading strategies in Ethereum using on-chain data. Initially, the classification-based strategy demonstrates suboptimal performance. However, by utilizing feature engineering techniques such as Principal Component Analysis and boosting, the strategy is optimized, resulting in the final Majority-Vote Strategy. This strategy outperforms the benchmark with low excess return. However, to fully harness the excess information contained in on-chain data and further enhance the strategy's performance, we propose an alternative event-driven investment strategy. This novel approach identifies valuable information from cheating addresses in the Ethereum market, thereby extracting insights that enable more effective investment decisions. The event-driven strategy exhibits robust performance and generates significant returns. These findings demonstrate the potential of on-chain data for developing profitable trading strategies in Ethereum.

II Background

Cryptocurrency industry is a relatively new and rapidly growing sector that encompasses a wide range of digital assets and related technologies. It has attracted significant attention as a decentralized and secure method of conducting financial transactions. Ethereum, in particular, has emerged as a prominent cryptocurrency platform, offering a range of decentralized applications and smart contract functionality. One notable feature of Ethereum is the availability of on-chain data, which captures transactional activity and enables the exploration of new investment strategies. In this study, we focus on leveraging on-chain data to develop effective trading strategies for Ethereum, two models are proposed to solve the problem, which can potentially provide investment insights to general investors.

III Data Analysis

To explore the cryptocurrency market, our data comes from mainly two sources. The first data source is the historical market data from BINANCE, which is a cryptocurrency exchange. The dataset normally contains 7 explanatory variables: datetime, open price, high price, low price, close price, volume, and turnover. Our second data source is the historical data that are recorded on the ledger on the nodes of the Ethereum chain. From one node of the Ethereum chain, we can get the following 10 explanatory variables:

1) Number of Active Addresses: The number of unique addresses that were active in the network either as a sender or receiver. Only addresses that were active in successful transactions are counted.
2) Number of Blocks Mined: The number of blocks created and included in the main blockchain in that time period
3) Mean Block Interval: The mean time (in seconds) between mined blocks.
4) Total Block Size: The total size of all blocks created within the time period (in bytes).
5) ETH 2.0 Amount of New Value Staked: The amount of ETH transferred to the ETH2 deposit contract.
6) Total Transaction Fees: The total amount of fees paid to miners. Issued (minted) coins are not included.
7) Number of Transactions on Uniswap: The total number of transactions that contains an interaction within Uniswap contracts. Includes Mint, Burn, and Swap events on the Uniswap core contracts.
8) Mean Hash Rate: The average estimated number of hashes per second produced by the miners in the network.
9) Number of Transactions: The total amount of transactions. Only successful transactions are counted.
10) Number of Transfers per Second: The total amount of transfers per second. One transaction can trigger one or more transfers. Only successful, non-zero transfers are counted.

Figure 1 presents a visualization of the variables, while Figure 2 illustrates the correlation matrix. The high correlation coefficients observed in the matrix suggest the presence of collinearity among the features. In subsequent sections, both market and on-chain data will be employed to construct predictive models for future returns, as well as to explore the potential utility of on-chain data in other contexts to reveal hidden information.

IV Methodology

IV-A Classification Based Investment Strategy

The cryptocurrency market is known for its high volatility and unpredictability nature. When evaluating performance of an asset,investors typically first examine its market value (C_t), which is determined by multiplying the price of Ethereum per coin (p_t) by the trading volume (V_t) at day t. Based on the definition, typically, people tend to use asset price as a primary criterion when formulating their investment strategy and put weights on predicting future prices. However, the price of a

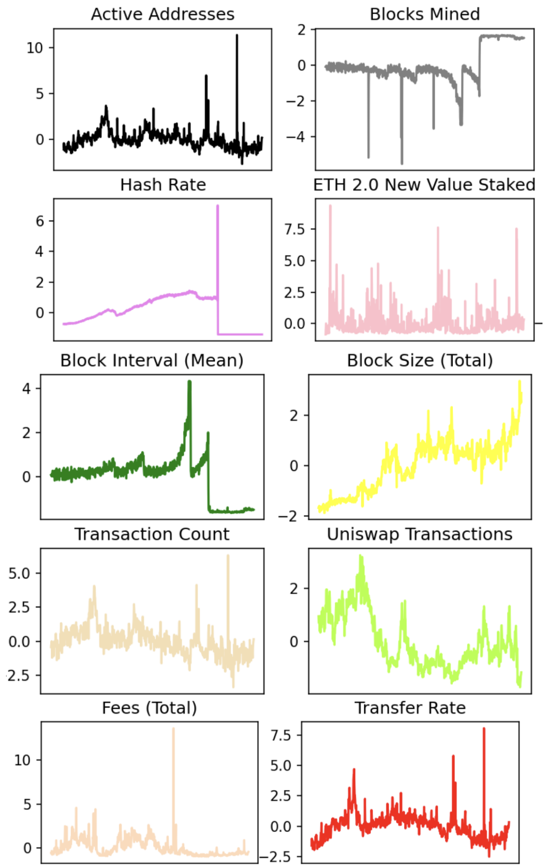

Fig. 1: Visualization of variables

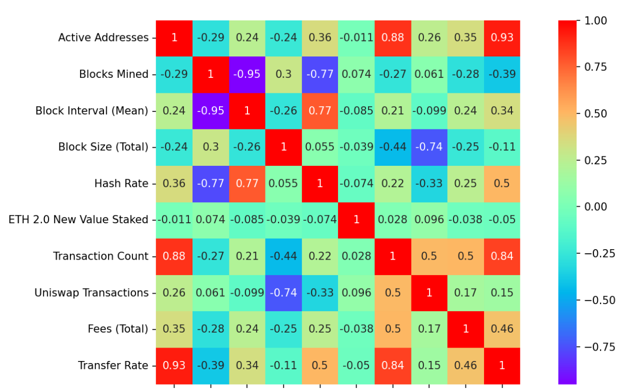

Fig. 2: The correlation matrix

financial asset is generally viewed as a stochastic process that follows a random walk, which lacks a probability distribution and hard to predict. In contrast, based on the conventional assumption of financial markets [1], the return rate of an asset is assumed to follow a lognormal distribution, making it easier and more reasonable to predict. Thus, a return based strategy is more reasonable and reliable. Define return rate at $t+1$ as

$$r_{t+1} = \frac{p_{t+1}}{p_t} - 1, \tag{1}$$

and our strategy is built based on the follows: when $r_t \geq 0$, indicating an expected increase in value of Ethereum, we adopt a long position; otherwise, we take a short position. To simplify our problem, we convert r_t to an indicator for future upward/downward trend, where

$$y_t = \begin{cases} 1 & \text{if } r_t \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

Introducing the variable y_t enables the transformation of our initial return prediction problem into a binary classification problem to predict the daily trend direction (i.e., upward or downward). Based on this prediction, a long or short position is taken. Subsequently, we describe the long-short strategy S_t for day t as follows:

$$S_t = \begin{cases} \text{long} & \text{if } y_t = 1 \\ \text{short} & \text{otherwise} \end{cases} \tag{3}$$

Thus, the implementation of the first classification based strategy can be divided into two part. First, for each day t, use some classification models to predict whether Ethereum is going upward or downward at $t+1$. Then according to the classification result, build our strategy at $t+1$ according to (3). In the following subsections, we described three methods (SVM, Decision Tree and Random Forest) to conduct the classification for Ethereum trend, and select the most accurate one to build our strategy.

Define error metric E_{mc} as

$$E_{mc} = \frac{n_y}{n} \tag{4}$$

where n_y is the number of instances that are incorrectly classified, and n is the total number of instances in the dataset. For binary classification problem, we consider to use the error metric (4) and AUC-ROC curve [2] to measure the model accuracy.

IV-B Event-driven Investment Strategy

The second investment strategy entails utilizing on-chain data and employing an event-driven approach. Firstly, we perform a comprehensive analysis of data to identify deceitful addresses. Given the prevalence of insider information in the cryptocurrency market, it is highly probable that numerous addresses exploit such information for trading purposes. Identification of these cheating addresses can facilitate further analysis to determine the correlation between their actions and the future market return. In the event that a strong correlation is observed, we can proceed to develop an event-driven investment strategy, referred to as ”Follow the smart money”, by mimicking the actions of these addresses to derive profits.

IV-C Backtesting

In finance market and trading, back-test is an important process of evaluating the performance of an investment strategy by using historical data. Back-testing involves applying a set of rules for buying and selling assets or securities to historical market data to simulate what the performance of the strategy would have been if it had been applied during that time period. The goal of back-testing is to assess the potential profitability of a trading strategy, identify its strengths and weaknesses, and refine the strategy before using it in live trading.

Generally, as [3] suggests, back testing involves several steps, including selecting data set, defining trading rules, simulating trades based on the rules, and analyzing results to evaluate the strategy's performance.

In the present study, we undertake a back-testing exercise to evaluate the performance of our proposed models. The back-testing rules are based on the trading strategies generated by the respective models. Specifically, for the Classification Based Investment models, we employ the data spanning from January 2020 to December 2022 as the training set, while the period from January 2023 to April 2023 is used as the test set and the back-testing set.

Given that the Event-driven Investment model requires a larger dataset to effectively detect ”cheating addresses” and driven events, as elaborated in subsequent sections, we employ the data from January 2018 to December 2020 as the training set and January 2021 to April 2023 as the testing set. To ensure the robustness of the Event-driven Investment model, we conduct back-testing using a longer time horizon, from 2021 to 2023, to verify its performance.

V Model & Result

V-A Classification Based Investment Strategy

Given the unique strengths of Support Vector Machine (SVM) in handling high-dimensional data, Decision Tree (DT) in providing a simple and interpretable model that facilitates visualizing the decision-making process [4], and Random Forest (RF) in ensuring model robustness and preventing overfitting with high accuracy [5], we have opted to employ these three classic machine learning models in our prediction of optimal trading strategies. Since our data is not linear separable, we consider to apply convert SVM to kernelize SVM with Gaussian kernel (rbf - SVM) [6]. When employing entropy as the impurity measure, the Decision Tree model was set to have a maximum depth of 5, while the Random Forest model was optimized with a pair of (maximum depth, number of estimators) = $(5, 80)$. These parameters were determined through a random search to find the optimal solution.

Table I summarizes the prediction accuracies of our proposed models based on the error metrics, where the outputs are labels defined as (2).

Table I: Accuracy Mertics for Base Methods

	SVM	rbf - SVM	DT	RF
E_{mc}	0.551	0.449	0.472	0.427
AUC score	0.500	0.500	0.507	0.585

And the figure 3 shows the visualization of decision tree model.

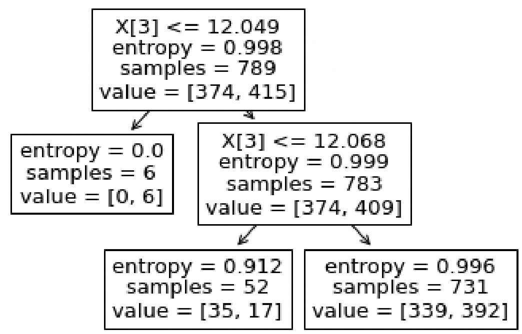

Fig. 3: Visualization of decision tree

Based on the above result, we can see that after applying the kernel trick, the performance of SVM indeed improved. The strategies suggested by the four methods and the AUC-ROC curve can be found in the Appendix VIII and IX correspondingly.

In the context of developing financial strategies, evaluating model performance based on accuracy alone may not suffice to determine financial return. Therefore, model performance is typically evaluated using multiple metrics, with the return rate being particularly significant in addition to model accuracy. In accordance with the investment principles outlined in [7], ”always holding Ethereum” is used as the benchmark strategy, i.e. the benchmark return would equal to the return of Ethereum. To evaluate the efficacy of the four classification investment strategies, a back testing experiment is conducted for the period ranging from January to April 2023, with the results compared against the benchmark strategy. The comparison is presented below.

The experiments conducted reveal that the return curves of the two SVM-based models, which recommend Ethereum to be held throughout the observed period, are consistent with the benchmark return. Conversely, the back testing results

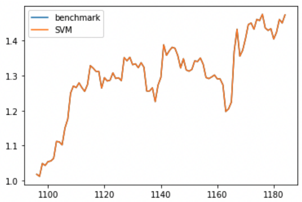

Fig. 4: Returns by SVM

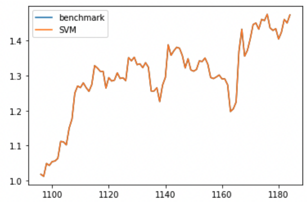

Fig. 5: Returns by rbf - SVM

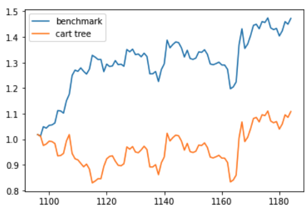

Fig. 6: Returns by Tree

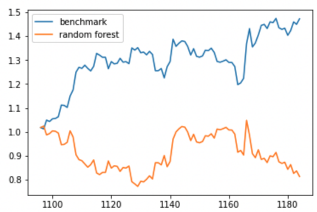

Fig. 7: Returns by RF

also indicate that although the DT and RF models exhibit greater prediction accuracy relative to the SVM-based model, their performance is suboptimal, resulting in the failure to outperform the benchmark. Thus, to enhance the performance of the classification-based strategy, we aim to investigate some potential feature engineering techniques.

V-B Improvements for Classification Based Investment Strategy

1) Principal Component Analysis

In our pursuit of improving the performance of classification-based strategies, we experimented with various feature engineering techniques. The first approach we employed was Principal Component Analysis (PCA), given that tree-based methods are susceptible to overfitting and our data, as indicated in Figure 2, exhibits high correlation among features. We were motivated by the suggestion in [8] that PCA can enhance the generalization capability of Decision Tree and Random Forest models. Through experiments, we find that when $K = 4$, our models are the most generalizable. With PCA trick, the mean error of DT (call the model with PCA: DT-PCA) can be reduced to 0.438 (from 0.472), and the AUC score can be increased to 0.572 (from 0.507), while the accuracy metrics for RF are not changed much. More details about the results of PCA models can be revealed in Appendix IX and X. The visualization of PCA-DT and confusion matrix for PCA-RF are as follows:

If we apply the new modifed model to same testing data set, we can get the following back testing return plots, refer to

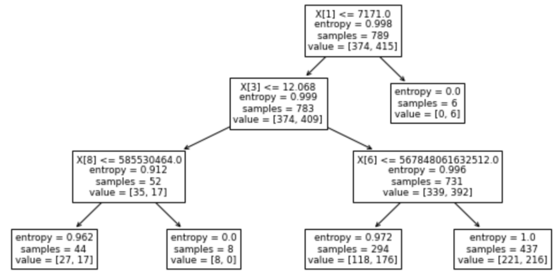

Fig. 8: Visualization of decision tree

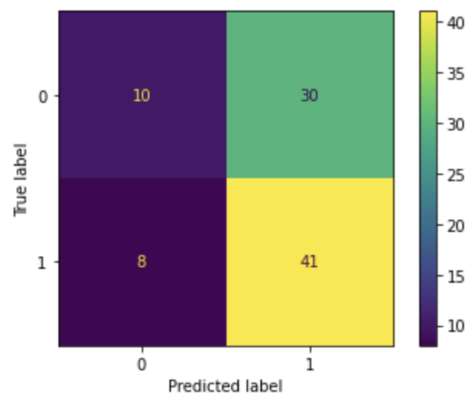

Fig. 9: Confusion matrix for random forest with PCA

figure 10 and 11, which shows that after conducting PCA, both DT and RF model perform better than the originals.

Fig. 10: Returns by PCA DT

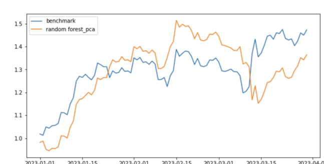

Fig. 11: Returns by PCA RF

2) Majority Vote Strategy

However, the general performance of all previous mentioned classification models are still worse than the benckmark strategy (except SVM, which coincide with benchmark). Thus, we consider the second approach to improve our performance.

The idea of Majority Vote Strategy is inspired by the Boosting and Bagging principle. As [9] discussed, gradient boosting is an algorithm that get a strong learner by training a series of weak learners. Based on the previous sections, we have already trained several weak learners. To better capture the trend of the market and improve performance, we consider the aggregation of previously trained models and use majority vote to determine our final prediction result. Mathematically speaking, for day t, define

$$y_t^{mc} = \begin{cases} 1 & \text{if mode(Y)} = 1 \\ -1 & \text{otherwise} \end{cases} \tag{5}$$

where $Y = \{y_t^{k-SVM}, \; y_t^{rf}, \; y_t^{pca-rf}, \; y_t^{pca-dt}\}$ represents the set of predictions for trend for day t by kernalize SVM, RF, RF with PCA, and DT with PCA respectively.

The error metric E_{mc} and AUC score were computed for the Majority Vote Strategy, yielding values of 0.410 and 0.54, respectively. More details results are in Appendix VIII and IX. The resulting backtesting return plot is depicted below.

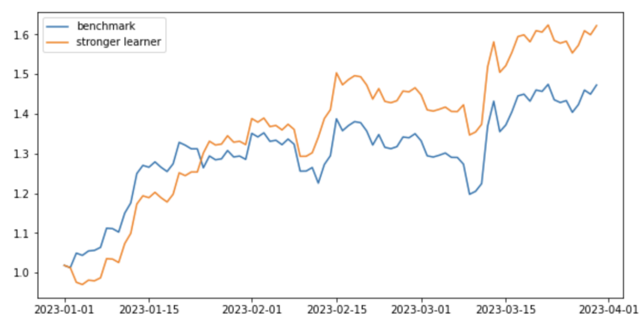

Fig. 12: Returns by Majority Vote Strategy

If we compare all the returns for Classification Investment Strategies, as the figure 13 indicates, the Majority Vote Strategy performs the best over the past four months. In conclusion, although simple machine learning classifiers cannot derive an efficient trading strategy (except also holding) for Ethereum, after doing a series of feature engineering and modification of the strategies, we can improve the performance and come up with a strategy that can outperform the benckmark for about 10% at the end.

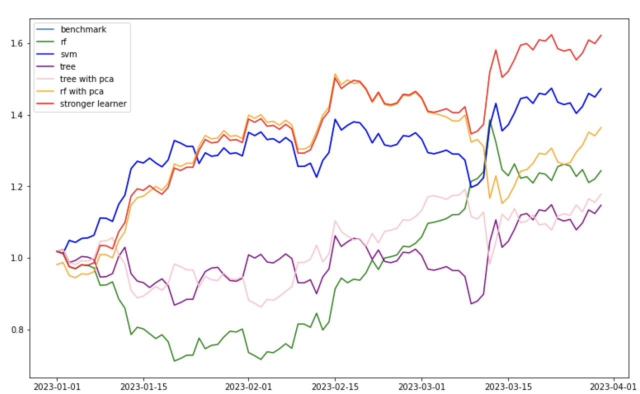

Fig. 13: Comparison of different classification strategies

Upon examining the entire testing period, it is evident that all Classification Investment Strategies lack a consistent excess return compared to the benchmark. Therefore, it is imperative to investigate alternative methods for constructing a strategy that yields higher profits and superior performance, which leads us to the second part – Event-driven Investment Strategy.

V-C Event-driven Investment Strategy

To construct an Event-driven Investment Strategy, a more thorough examination of the on-chain data is necessary to uncover relevant information. The initial step is to access an Ethereum node. Given that our price data comes from the Binance exchange, we retrieve all transactions that meet the criteria of having a transaction amount exceeding 1 million dollars and being received by the Binance exchange address, with the data range spanning the previous two years. The unique Hash value assigned to the Binance exchange is 0x3f5CE5FBFe3E9af3971dD833D26bA9b5C936f0bE.

In this study, we computed the Ethereum return rates of each transaction for a three-day and seven-day window after its occurrence. The addresses were then grouped as variables to derive the cumulative yield over the aforementioned periods, as well as to calculate the number of transactions with positive three-day yields and the total number of transactions. Subsequently, we employed the K-means algorithm to cluster the addresses into two categories: cheating addresses and non-cheating addresses. The average statistics of classification results is shown below:

Table II: Classification result statistics

	cumulative yield for three-day	cumulative yield for seven-day	number of positive yield	total transactions
cheating addresses	-1.52%	0.24%	56	243
non-cheating addresses	0.21%	1.08%	78	122

According to the analysis of the table, an intriguing observation can be made: the cumulative yield for the three-day period following the transactions of cheating addresses is not only negative but is also significantly smaller than that of non-cheating addresses. Notably, the number of transactions yielding a positive yield is found to be merely one-fourth of the total transactions executed by cheating addresses. This raises the possibility that these addresses may have acquired advanced information regarding short positions in the market, prompting them to transfer funds to the exchange with the intent of short selling instead of long positions. Such an inference can have profound implications for understanding the underlying dynamics of market manipulation and may offer novel insights into developing effective strategies that follow cheating activities in financial markets.

To verify our hypothesis, we plot the three-day cumulative return for the address with the largest total transaction. The plot is also the performance of ”Follow the smart money”.

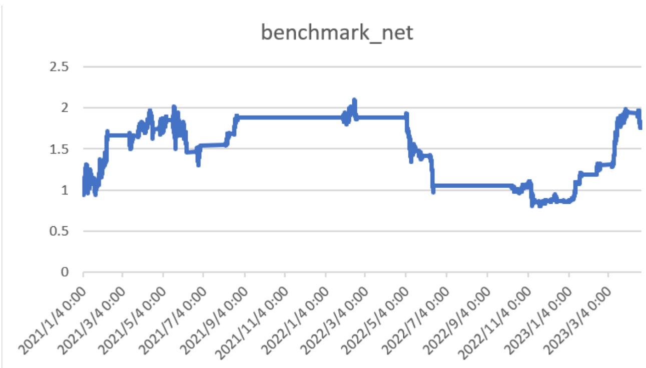

Fig. 14: Performance of smart money

Although the hypothesis was not confirmed by figure 14, it still provides valuable insights. Despite the lack of a clear directional trend in the graph, a careful observation reveals that there are several prominent unilateral market trends following the transfer of money from cheating addresses to the exchange. Subsequently, we applied ordinary linear regression to fit the benchmark return using the absolute value of the three-day return for the cheating address, which resulted in an R^2 value of 0.63. This indicates that the cheating address has some ability to forecast future trends. It can be inferred that when the cheating address engages in a transaction, the future is likely to be a trending market rather than a volatile one, although it is impossible to predict whether the market will rise or fall.

Then we could combine the actions of cheating addresses and a trend strategy to develop a timing strategy. Here, we choose to use a dual moving average trend strategy. A dual moving average trend strategy is a popular trading strategy that involves using two moving averages to identify trends in stock prices. The strategy involves calculating the short-term moving average (SMA) and the long-term moving average (LMA) of a stock's price and then watching for a crossover of the two lines as a buy or sell signal.

$$MA_t = \frac{1}{n}\sum_{i=1}^{n} P_{t-i+1} \tag{6}$$

The specific strategy implementation details are as follows: We monitor the cheating addresses with historical top 30% returns. When more than 50% of the addresses have transfer behavior, we regard the future market as a trend market. At this time, we start the dual moving average trend strategy and keep running the strategy for seven days, we used the typical parameter setting of 6 and 24 for the parameters of the dual moving average strategy. We can use the characteristics of the trend strategy to make up for the disadvantage of not being able to predict the market ups and downs from cheating address behavior. Since after conducting backtesting for the period after 2023, we get ultra-high returns, to further check the robustness of the strategy, we test it over the past two years. The backtest result of the strategy for the past two years with the benchmark returns is as follows in figure 15.

Fig. 15: Performance of the Event-driven Strategy

We could see that the excess return of the new timing strategy is as much as seven times that of the benchmark. If we consider the transaction fee, we set the transaction fee to be $8\%\%$ per transaction, and the final excess return is about 568.3%, which is far more than the benchmark.

VI Further Improvement

Due to the high correlation among features, the initial Classification Investment models exhibit low prediction accuracies. For example, SVM predicts all trends to be upward, suggesting an always-holding strategy, which is same as not conducting any transcations during the period. Although the final Majority Vote Strategy manages to outperform the benchmark over a four-month testing period, its robustness remains to be tested over longer periods. Future studies may investigate the effectiveness of the Majority Vote Strategy over extended periods and explore other feature engineering techniques to improve prediction accuracy. For instance, adding a feature that captures the previous day's return trend can provide additional information and potentially enhance the strategy's performance. What is more, since we are concentrating on exploring profitable models rather than applying for real trades, future studies can also set up some experiments to analysis the risk level of models and their practicability in real financial market.

As previously noted, the price of Ethereum is determined by the interplay of supply and demand. In financial markets, numerous products exhibit correlations, including complementary and substitute goods, with their prices exerting mutual influence. The identification of fluctuations in the supply and demand of a given product may facilitate the prediction of corresponding fluctuations in another. Thus, forthcoming research can further examine alternative approaches for constructing effective trading strategies by exploring the interconnections among different goods in the cryptocurrency market. It is hypothesized that these demand correlations may be easier to detect and can provide supplementary information for our second Event-Driven Investment Strategy, which rendering the second method more reliable and resilient to market conditions.

Additionally, further research can include the development of evaluation metrics to assess the accuracy of the second model, which was not addressed in this study. The Event-driven Investment model utilizes K-means as an unsupervised learning algorithm to classify "cheating" and "non-cheating" addresses, making it difficult to evaluate the accuracy of a model that does not have pre-defined labels or categories. The implementation of the Elbow Method, as suggested by Syakur and Wibowo (2018) [10], and the Silhouette Score, as proposed by Shahapure and Uplane (2020) [11], can be considered as potential methods to assess the accuracy of unsupervised learning models.

VII Conclusion

In conclusion, it is esay to see that by using on-chain dataset, which contains more transparent and rich information, investors can potentially build models with better performance than only rely on traditional market information. However, although the on-chain dataset could provide extra information to the model, simply using machine learning methods to predict future returns is not a good idea. By conducting comprehensive feature engineering and making modifications to the fundamental classification machine learning approach, the model can be tailored to be more applicable to the cryptocurrency market. However, despite these efforts, generating significant excess returns by machine learning classification remains a challenging task.

On the other hand, while the initial attempt to "Follow the smart money" using the second methodology yielded poor results, the experimentation process revealed that an effective temporal strategy can be derived by integrating the behaviors of smart money with a trend strategy. The resulting performance of the novel approach is markedly favorable, substantiating the existence of insider trading in the cryptocurrency market. Given the transparent nature of the cryptocurrency market, these transactions can be monitored, and if properly identified and leveraged, they present lucrative opportunities for profit.

Based on the presented evidence, it can be inferred that on-chain data analysis can provide insightful information, and the Event-Driven Investment Strategy can generate significant returns in the Ethereum market and is more profitable than basic classification based investment strategies. However, the development and implementation of this strategy requires rigorous experimentation to validate hypotheses and a comprehensive understanding of both the financial market and Ethereum, posing challenges for ordinary retail investors. Additionally, while classical machine learning methods are simple to execute, they often fail to produce effective strategies for Ethereum trading. Consequently, we recommend that retail investors adhere to Buffett's investment approach, which advocates for holding underlying assets rather than engaging in frequent transactions. In contrast, professional traders may consider leveraging Event-Driven strategies to capitalize on the high potential returns in the Ethereum market.

References

[1] S. Chan, J. Chu, S. Nadarajah, J. Osterrieder, A statistical analysis of cryptocurrencies, Journal of Risk and Financial Management 10 (2) (2017) 12.

[2] S. Narkhede, Understanding auc-roc curve, Towards Data Science 26 (1) (2018) 220–227.

[3] S. D. Campbell, A review of backtesting and backtesting procedures (2005).

[4] R. J. Lewis, An introduction to classification and regression tree (cart) analysis, in: Annual meeting of the society for academic emergency medicine in San Francisco, California, Vol. 14, Citeseer, 2000.

[5] G. Biau, E. Scornet, A random forest guided tour, Test 25 (2016) 197–227.

[6] T. Hofmann, B. Schölkopf, A. J. Smola, Kernel methods in machine learning (2008).

[7] R. G. Hagstrom, The Warren Buffett Way, John Wiley & Sons, 2013.

[8] H. Abdi, L. J. Williams, Principal component analysis, Wiley interdisciplinary reviews: computational statistics 2 (4) (2010) 433–459.

[9] A. Natekin, A. Knoll, Gradient boosting machines, a tutorial, Frontiers in neurorobotics 7 (2013) 21.

[10] M. Syakur, B. Khotimah, E. Rochman, B. D. Satoto, Integration k-means clustering method and elbow method for identification of the best customer profile cluster, in: IOP conference series: materials science and engineering, Vol. 336, IOP Publishing, 2018, p. 012017.

[11] K. R. Shahapure, C. Nicholas, Cluster quality analysis using silhouette score, in: 2020 IEEE 7th international conference on data science and advanced analytics (DSAA), IEEE, 2020, pp. 747–748.

Appendix A

VIII Trading Strategies by Classification Based Investment Models

Trading strategies for four base machine learning methods for testing period (from 2023/01 to 2023/04):

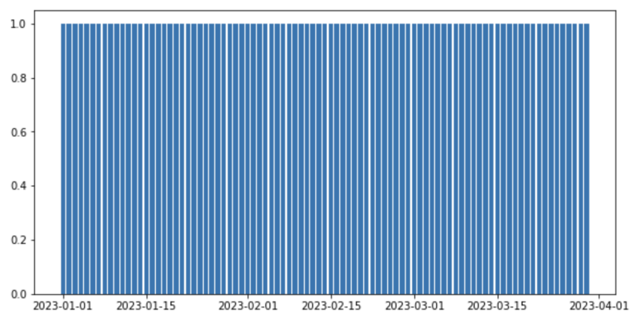

Fig. 16: Trading Strategy by SVM

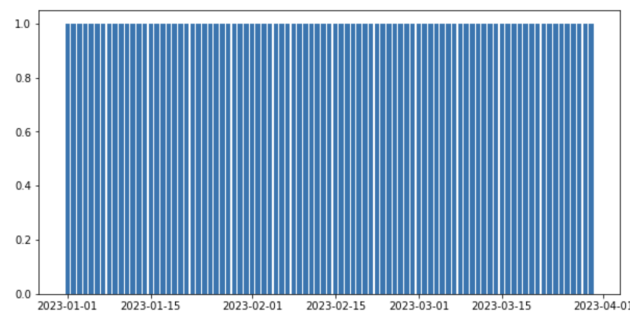

Fig. 17: Trading Strategy by Tree

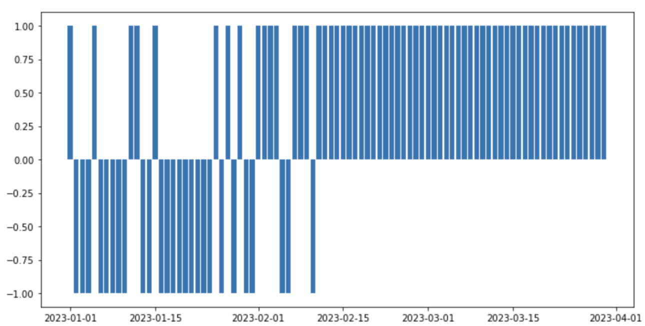

Fig. 18: Trading Strategy by RF

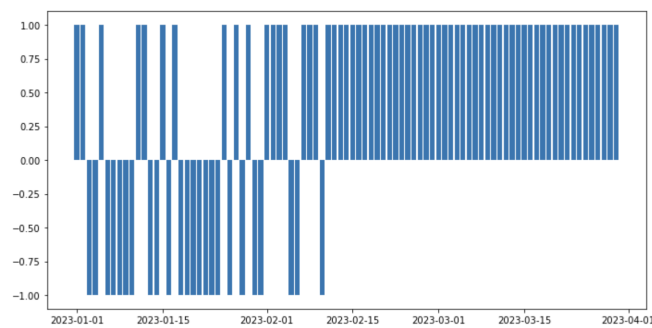

Fig. 19: Trading Strategy by RF

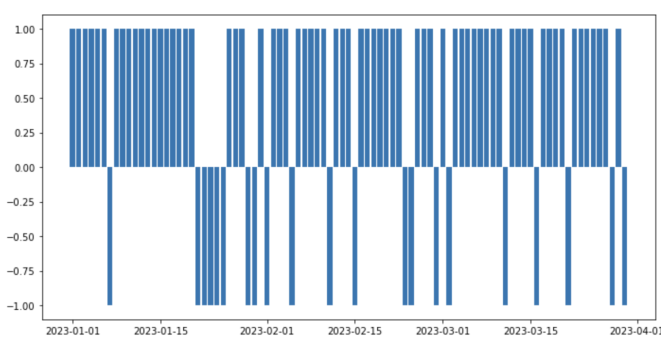

Fig. 20: Trading Strategy by DT with PCA

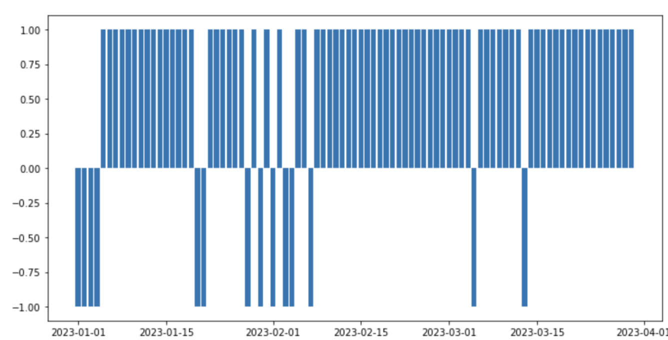

Fig. 21: Trading Strategy by RF with PCA

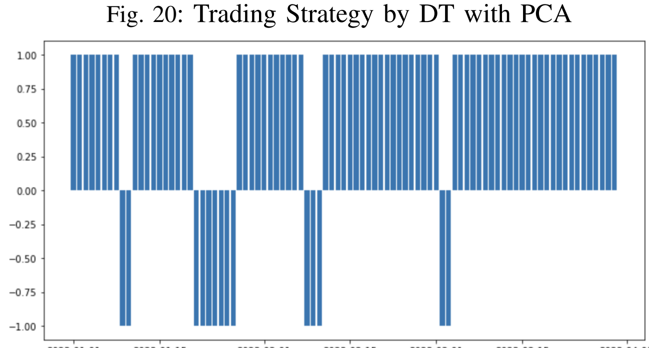

Fig. 22: Trading Strategy by RF with Majority Vote Stratedy

Here, at day t, value 1 means taking long position of Ethereum, and value -1 means taking short position.

Appendix B

IX AUC-ROC curves for Classification Based Investment Models

The following figures show the AUC-ROC curve for the four base machine learning classification driven models.

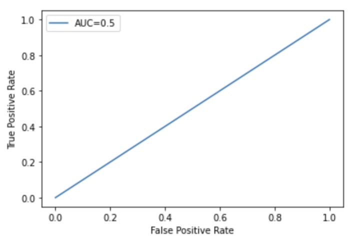

Fig. 23: Trading Strategy by SVM

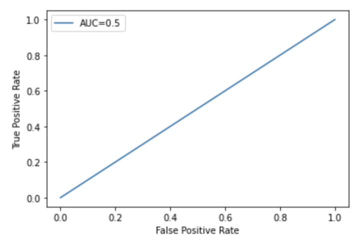

Fig. 24: Trading Strategy by Tree

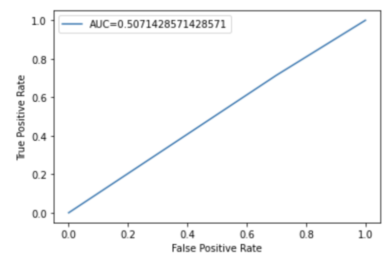

Fig. 25: Trading Strategy by RF

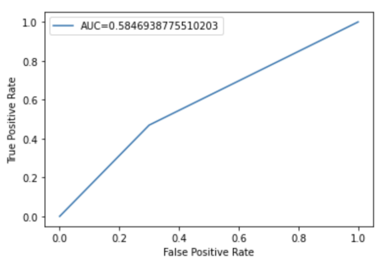

Fig. 26: Trading Strategy by RF

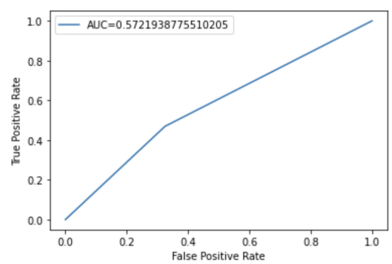

Fig. 27: AUC-ROC curve for DT with PCA

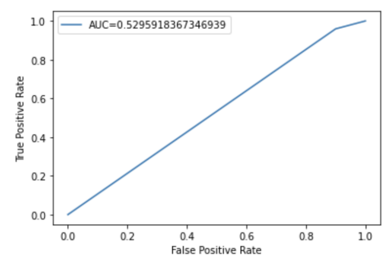

Fig. 28: AUC-ROC curve for RF with PCA

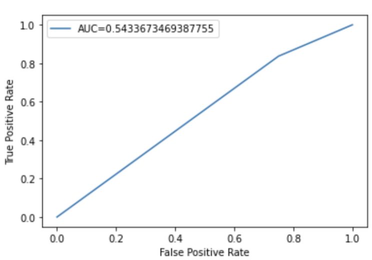

Fig. 29: AUC-ROC curve for Majority Vote Strategy

Appendix C

X Confusion Matrices

The following figures show the confusion matrix for Random Forest model with and without PCA:

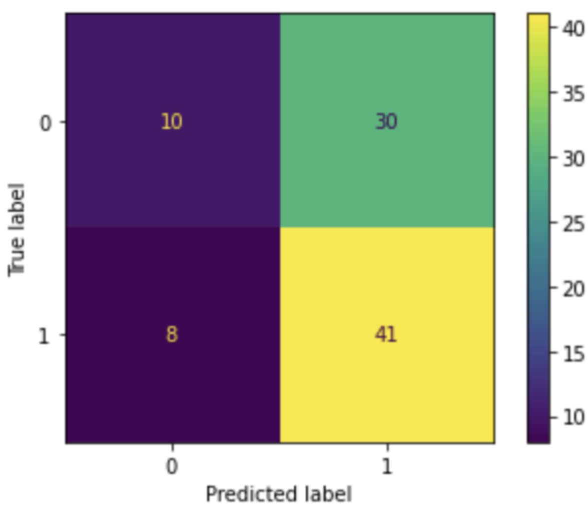

Fig. 30: Confusion matrix for RF with PCA

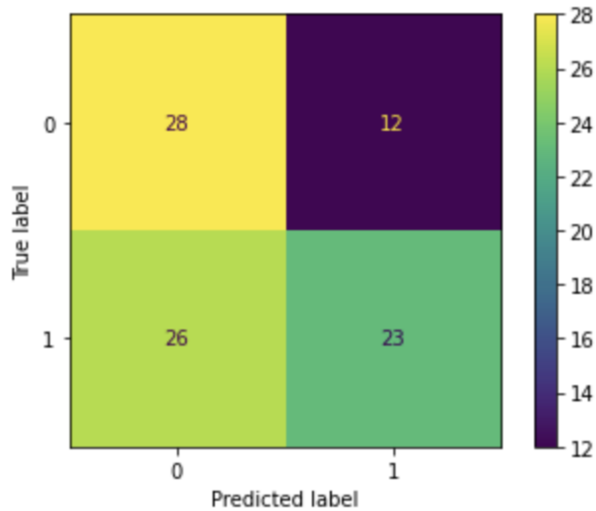

Fig. 31: Confusion matrix for RF