NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS
International College of Economics and Finance

Mikhail Mironov

Study of the Price Manipulation Schemes in Cryptocurrency Markets. Analysis of Insider Buying Volume and Post-Event Price Discovery

Term paper
38.04.01 ECONOMICS
Master's Programme **"Financial Economics"**

Moscow 2024

Contents

Abstract

Over the last couple of years, we have seen multiple pumps and dumps schemes organized on centralized exchanges. In this paper, we aim to rediscover the topic of cryptocurrency pump and dump detection and build a classification model able to label pumped tickers up to 1 hour before the pump. We apply a novel cross-sectional approach to the problem as well as cross-sectional normalization of features. We trained multiple boosting models and compared the results with other similar researches. We also propose a trading strategy which invests into potentially pumped assets based on logits of our models.

1 Introduction

1.1 What are Pumps and Dumps (P&Ds)

Pump and dumps are fraudulent schemes that involve huge price increases that do not have any fundamental backing or explanation. Inexperienced investor might consider this a great opportunity to invest, but little does he know that this price move is caused solely by a huge number of people and bots buying up the asset, all in pursuit to be the first to buy and sell for profit to others in minutes or even seconds. Essentially, this is a zero-sum game, simple redistribution of wealth between investors that happens in seconds, that is why it is very dangerous to trade such events, as you might end up being the one who is "holding the bag". The only people consistently making profits from such events are organizers, admins that orchestrate these events through Telegram channels. They buy into the token over time in order not to have a big impact on the price, then they warm up people interest by announcing the pump date and exchange and repeat the message over time, more frequently closer to the time of the pump, then they announce the token and people rush in to buy, the price increases significantly and admins or insiders sell off to people buying up the token making a quick profit in seconds.

Figure 1: VIBBTC pump on Binance

Above we can see a clear example of a pump and dump (P&D), we see little to none volume prior to the pump and then once the token is announced, there is a sudden spike in trading volume. As a result, the price increases significantly and then steadily dies off in the following seconds as true insiders (organizers) sell their positions to people participating after the announcement.

1.2 Related work

Topic of pumps and dumps in cryptocurrency has been thoroughly studied in the literature. One of the first papers to dig into P&Ds was (Kamps and Kleinberg 2018). Authors laid the foundation for the further research by explaining how these P&Ds schemes work. Authors found that mostly small market cap cryptocurrencies are chosen for manipulation as the prices could be moved with lower trading volumes. (Kamps and Kleinberg 2018) were among the first to propose automated approach to detection of P&Ds in trading data. They collected data using CCXT library for over 115 various cryptocurrency exchanges, but only 25 were left matching their criteria. They used 1h OHLCV data for their analysis. Authors focused on unsupervised anomaly detection relying on changes in prices and volumes. They employed a threshold strategy that would label price anomaly if high price of the candle is some percentage above the simple moving average computed over some predetermined window. Authors used the same idea for anomalous increases in trading volume. They used multiple combinations of window sizes and percentage thresholds. Their work sparked the interest of researchers of this topic, as it is one of the most cited papers on the topic. Following this paper, there have been multiple researches which employed supervised / semi-supervised learning to label P&Ds. Globally, these papers could be split up into 2 major groups, ones that want to predict pumps in advance (prospective) and that want to label pumps correctly among already known instances of pumps (retrospective).

1.2.1 Denial of service (DoS). Retrospective approach

The first category of papers is dedicated to performing real time pump detection, when trading data of the P&D is exposed to the model. In such researches the main objective is to correctly label the moment when the pump started, typically the imbalance in these papers comes from trading data of the same manipulated assets but at different trading time. For instance, (La Morgia et al. 2020) trained 2 types of models based on Random Forest and Logistic Regression. They manually collected 343 instances of P&Ds across 4 exchanges, but focused on Binance exchange to train the model. As a result, they were left with 104 P&Ds against BTC, for each they collected tick-level data for 7 days before and after the pump using Binance API. The data is split into chunks of s seconds and features are computed over windows of size w, chunk is labeled as 1 if the announcement time lies within the chunk, 0 otherwise. The model is trained to correctly label these chunks based on features computed. Authors were able to produce great results: their models reached f1-score varying from 0.84 to 0.92 depending on the chunk size (from 5 seconds up to 25 seconds). Later (La Morgia et al. 2023) extended their dataset of P&Ds and also applied Adaboost to the problem and as a result were able to improve evaluation metrics, they were able to reach f1-score of 0.94 on cross-validation sets. They also made a big contribution by releasing their dataset of confirmed pumps and dumps with 1111 instances of P&Ds across multiple telegram channels. (Fantazzini and Xiao 2023) followed similar methodology, authors collected 351 instances of pumps on Binance using PumpOlymp aggregator website. They also downloaded tick-level data using Binance API, as a result for each P&D they had 3 days worth of data, 1 day before the pump, the day of the pump itself and the following day. (Fantazzini and Xiao 2023) mostly focused on tackling the problem of imbalanced data as pumped chunks accounted for a measly 0.2% of overall data. They put much focus on resampling methods like ROS (random oversampling), RUS (random undersampling) and SMOTE. (Chadalapaka et al. 2022) also used the same setup of the problem but tried to apply neural networks to P&Ds detection. The authors used the same datasets as (La Morgia et al. 2020) but used different models, they applied CLSTM architecture as well as Anomaly Transformer proposed by (Xu et al. 2022). With these more complex models, they claim to slightly outperform (La Morgia et al. 2020) results. All of these papers also share the set of

variables they use for their analysis in Table (1).

Table 1: Description of variables used in these studies

Variable	Definition and explanation
StdRushOrders **AvgRushOrders**	the moving standard deviation and the average of the volume of rush orders in each chunk of the moving window
StdTrdes	the moving standard deviation of the number of trades
StdVolumes **AvgVolumes**	the moving standard deviation and the average of the volume of trades in each chunk of the moving window.
StdPrice **AvgPrice**	the moving standard deviation and average of the closing price
HourSin, HourCos **MinuteCos, MinuteSin**	the hour and minute of the first transaction in each chunk. We used the sine and cosine functions to express their cyclical nature

We can see that the features themselves are not that complicated. The main criticism that we have with the above methodology and approach to the problem is the lack of real imbalance. The above papers do not collect data on other regular tickers that were not manipulated by P&Ds organizers, therefore, they train the models using trading data of only known pumped tickers. We understand that the problem they solve is slightly different but still we believe that integration of other tickers could have helped to produce more realistic results. Therefore, we introduce the second category of papers, that, in our opinion, used a better approach to the problem.

1.2.2 Prospective approach

Papers from this category aim to predict if the ticker will be pumped in advance. The paper that we liked the most and in our opinion solves the problem in the most fair way is (Xu and Livshits 2019). Authors focused on P&Ds organized on Cryptopia. They collected 412 pumps from June 17, 2018 to February 26, 2019, out of those, 180 were organized on Cryptopia exchange. As market data they used OHLCV data downloaded from CryptoCompare.com API and they did it not only for pumped tickers as was done in previously mentioned papers but for all tickers traded on Cryptopia at the time of the pump. This way for each pump they created a cross-section where each cross-section consists of a pumped ticker labeled as 1 and other tickers labeled as 0 that were present on the Cryptopia exchange around the time of the pump. Using this approach (Xu and Livshits 2019) obtained a tabular dataset with computed features using data for all tickers including the manipulated ones and a target label indicating if the ticker is manipulated. Then, authors split 180 pumps equally into train, validation and test sets in chronological order and trained Random Forest and GLM models using these splits. As a result, they were to achieve f1-score of 0.15 on the test set using Random Forest model. This is a great result given that as they stated on average each cross-section had 296 tickers in it with only one being the manipulated ticker which implies a problem of server imbalance in target labels. As a result, they produced a model able to predict P&Ds up to 1 hour before the pump. Moreover, they proposed using this model as a trading strategy: they backtested their strategy involving buying potentially pumped tickers proportional to logits produced by Random Forest model. (Xu and Livshits 2019) assumed that other assets will not move in price during the hour before the pump, they also assumed that they are able to exit the pump perfectly at the top. With all of this, they managed to produce a return of 60% during the

backtest. In our paper, we also aim to relax some of these assumptions as well as produce the model with better predictive performance.

Some papers attempted to incorporate data from social media to improve performance of their models. (Nghiem et al. 2021) collected data from PumpOlymp, their collection of historical pump events dates back from June 17, 2018 to December 19, 2019. Their sample has 324 pump events transpired on Binance, followed by Yobit with 269, Cryptopia with 243, and Bittrex with 49. Authors collected hourly OHLCV data from CryptoCompare API. They collected 48 hours prior to the pump worth of data for the pumped ticker and 500 hours of random trading data that doesn't overlap with those 48 hours within 72 hours interval. They also collected market cap data from Binance at a fixed snapshot. For social data, authors also used CryptoCompare API, they were able to get Twitter, Facebook, Reddit, Github pages for each token, then they collected their follower counts and other activity measures (like commits, starts on github and etc). They calculated features over the period of 48 hours before the pump and apply min-max scaling to features, then label the observation as "1" if the token is targeted by pump and dump, 0 otherwise. They split the data into train, validation, test samples with 197, 55 and 54 pump events respectively. Train sample is populated with those random trading intervals whereas validation and test sets contain just trading hours of the pumped tokens. To tackle disbalance in the target variable, authors applied resampling methods like undersampling. Then, authors trained multiple models for the following lookback periods 6h, 12h and 24h before the pump hour using either financial data, social signal data or both. Authors trained Logistic Regression with a lookback of 6h of market data as a baseline, they also played around with the *class_weight* parameters to tackle class imbalance. Then, they trained multiple CNNs with different architectures, minimizing binary cross entropy loss. Authors also tried different LSTM-based models, like BLSTM (bidirectional LSTM) and CLSTM (CNN + LSTM). The best result they were able to achieve is f1-score of 0.16 on the test set with CNN with only financial features. They concluded that addition of social features by itself doesn't improve the quality of the models. Overall, this is a great paper, but the way (Nghiem et al. 2021) introduced imbalance to the problem is slightly confusing, depending on the number of random trading hours they include, they basically control the degree of imbalance in the data which is unfair. But results they obtained are still comparable to (Xu and Livshits 2019) which to some extent reassures that there is no evident problems with it.

Another interesting paper is (Hu et al. 2023) where authors also agree with us that retrospective approach does not actually solve any problem as investors are left with much time to make a decision as pumps start and end in the matter of seconds. Therefore, it is more valuable to predict P&Ds in advance. (Hu et al. 2023) in the similar fashion use cross-sectional approach to the problem as in (Xu and Livshits 2019) to predict the pumped token one hour before the public announcement. Authors also developed an automated pipeline to parse telegram channels and extract previous P&Ds. This way they were able to collect 709 instances of pumps with 445 that took place on Binance. (Hu et al. 2023) downloaded hourly OHLCV data from Binance API. Authors noted that they saw on average an increase in volume and price in the run up to the pump with several spikes in volume appearing between 48 hours to 1 hour before the pump which they attributed to VIP members increasin their position in the token. Authors adopted the same methodology for feature engineering from (Xu and Livshits 2019) when they take windows of size x+1 to 1 hour before the pump and compute features over such windows where $x \in [1, 3, 6, 12, 24, 48, 60, 72]$. Authors proposed using sequence based approach using SNN along with positional attention. They believe that such approach could catch the patterns in how telegram groups organizing P&Ds choose next manipulated tokens. They split the training, validation, and testing sets by the timestamp "2021-01-19 00:00:00"

and "2021-05-10 00:00:00". They trained multiple ML models and compared results to their proposed SNN model, authors used top-k accuracy as evaluation metric where top-k accuracy is the probability that the true label will be among the top-k highest logits produced by the model. The results of this paper are shown in the Table (2) below. As we can see the best result was produced by SNNs model achieving top-30 accuracy of 0.823, meaning that with probability of 82.3 % true pumped ticker will be among the top-30 highest logits. In our research we focus on results obtained by (Hu et al. 2023) and (Xu and Livshits 2019) as a baseline for our models.

Table 2: Top-K accuracy on test set by (Hu et al. 2023)

	RF	DNN	LSTM	BiLSTM	GRU	BiGRU	TCN	SNN	**SNNs**
HR@1	0.189	0.225	0.207	0.203	0.229	0.163	0.256	0.260	**0.277**
HR@3	0.348	0.278	0.339	0.344	0.339	0.335	0.348	0.383	**0.414**
HR@5	0.417	0.383	0.423	0.396	0.414	0.401	0.427	0.465	**0.513**
HR@10	0.537	0.498	0.551	0.546	0.529	0.555	0.573	0.596	**0.623**
HR@20	0.687	0.626	0.648	0.630	0.626	0.678	0.692	0.727	**0.739**
HR@30	0.731	0.727	0.696	0.696	0.714	0.709	0.770	0.797	**0.823**

2 Methodology

2.1 Pump and dump data

In this paper we collected pumps and dumps dataset by downloading chat history from 8 telegram channels, then we manually labeled pumps and dumps, namely for each we extracted ticker, pump time and exchange. Our sample covers pumps from December 18, 2018 up to April 1, 2024. This way, we were able to collect in total 175 pumps, out of those 90 organized on Binance, 66 on Kucoin and 19 on MEXC. The majority of Telegram channels that used to organize P&Ds are inactive so their chat history has been deleted by Telegram. That is why our sample is much smaller than ones of the previous researches. To extend our sample, we have taken labeled pumps and dumps dataset from authors of the paper (La Morgia et al. 2020) which they have also updated in 2021. They made a big contribution as this sample contains 1111 pumps across multiple exchanges. We have filtered out all duplicates and left pumps organized on Binance, then we removed some suspicious pumps that didn't match our criteria of the pump (price and volume didn't change a lot during the pump which could be a sign of a failed or postponed pump) which we don't want in our sample. Then, we merged these pumps to ones we have collected and obtained s sample containing 337 unique pumps and dumps.

2.2 Market data

At this point we have decided to focus on researching pumps organized only on Binance exchange, as many previous researches did the same and we would like to compare our results with them. As a result, we have downloaded all trade level data from Binance up to 31-03-2024 using Binance's Datavision website. We ended up with 500+ GB worth of compressed trade level data for all tickers ever traded on Binance. Structure of collected data is presented below in the Table (3):

	price	qty	time	isBuyerMaker	quote
0	0.000002	2437.000000	2022-07-15 15:00:02.422000	True	0.004240
1	0.000002	275.000000	2022-07-15 15:00:02.434000	False	0.000479
2	0.000002	1125.000000	2022-07-15 15:00:22.884000	True	0.001946
3	0.000002	941.000000	2022-07-15 15:00:52.277000	True	0.001628
4	0.000002	184.000000	2022-07-15 15:00:52.277000	True	0.000318

Table 3: Structure of collected data: *price* shows the price that the order was fillted at, *qty* is the amount of base asset traded, *isBuyerMaker* shows if the buyer related to this transaction was already in the orderbook, if so then someone came and matched the buyer, therefore it is a SELL, vice versa for BUY.

We downloaded histories of all assets traded against all quotes like USDT, ETH, BTC and etc. In this paper we have a look at tickers traded only against BTC quote, which has also been done by other researchers. In the future we are planning to scale our research to other quotes by incorporating them as features converted to USDT values.

2.3 Feature engineering

Finally, we ended up with a lot of compressed zip files containing trade level data for each ticker for each month and year the ticker was traded on Binance. There are obviously different number of trades for each ticker each day, therefore we couldn't simply feed this data to any model, therefore we created a feature generation pipeline which handles decompressing data, splitting it into daily chunks, calculation of features describing the flow of trades prior to pump announcement and finally merging all features into a single crosssection associated to the pump. This pipeline is explained in details in Figure (2):

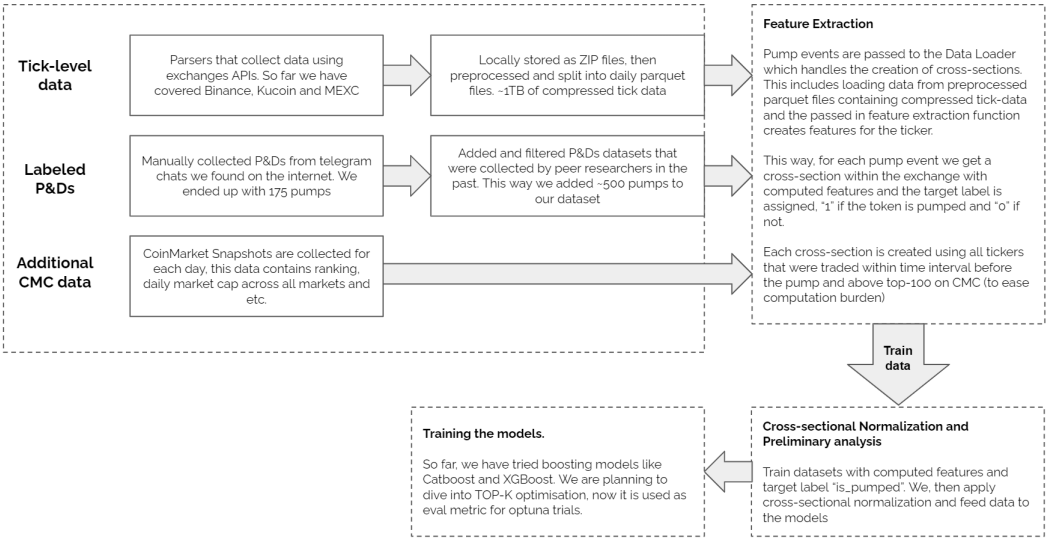

Figure 2: Feature engineering pipeline

Such flexible pipeline allowed us to go through multiple iterations of features by simply changing python file describing the logic of feature extraction. This has taken the most effort as working with data of this size required multiple optimizations. As a result, we were able to cut down feature generation time from multiple days down to 5 hours which allowed us to experiment

more freely with different features. We split features into separate groups, each will be described below.

2.3.1 Imbalance features

Imbalance is a type of metric measuring which side is more prevalent in trade data: buy or sell. For instance, volume imbalance measure skew towards buying or selling. If imbalance ratio is closer to 1, this implies that there are more buyers than sellers and vice versa.

$$\text{Volume Imbalance} = \frac{\sum_{t \in T} \text{quote\_sign}_t}{\sum_{t \in T} \text{quote\_abs}_t} \tag{1}$$

The formula (1) above calculates volume imbalance as the ratio of signed volume in BTC (negative for sell orders, positive for buy orders) over absolute volume in BTC. Below is the graph (3) of volume imbalance for VIBBTC prior and post the pump event, calculated using rolling window of size 5 minutes. We can observe some interesting patterns that prior to the pump we have some time intervals with very high volume imbalance suggesting big buying volumes whereas post the pump announcement imbalance drops to negative values suggesting the dumping phase where admins and insiders sell off their holdings to new participants. We believe that using features based on imbalances will allow us to capture buy vs short skews closer to the pump.

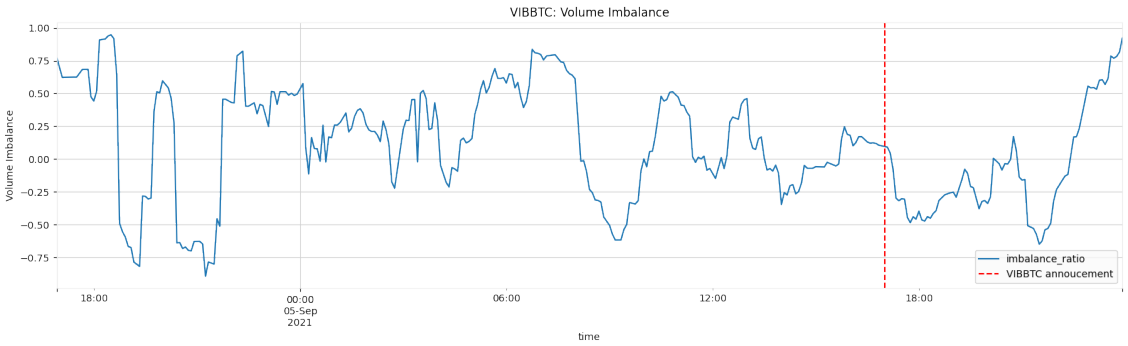

Figure 3: VIBBTC: Volume imbalance

2.3.2 Quote slippage features

We also created features based on slippages. Since we don't have information about order types, we can't see if the trade was executed by market or limit order, we believe slippage is a good proxy variable to measure market participants impatience. We follow the same methodology used in other papers where we would group ticks by execution time, saying that since we deal with highly non-liquid markets where time between trades can reach up to 15 minutes, we can safely assume aggregated ticks correspond to a single order. Therefore, we can calculate impact of the aggregated order (rush order in the literature) on the price. The idea is the following: if the market participant is ready to incur heavy slippage losses by posting a big buy order, then this person wants to fill in their position quick, maybe in anticipation of the pump. Following this idea, we have calculated slippages for each rush trade using the formula (2) below:

$$\text{Slippage Loss} = \underbrace{\sum_{i=1}^{N} \text{qty\_sign}_i \cdot P_i}_{\text{Quote actually spent}} - \underbrace{\sum_{i=1}^{N} \text{qty\_sign}_i \cdot P_0}_{\substack{\text{Quote could have been spent} \\ \text{if filled at best price}}} \tag{2}$$

This formula calculates the slippage loss as the difference between quote we actually spent and quote amount we would have paid if we were able to execute at the best bid or ask price (depending on the side of the trade). As an order is being executed it eats into the orderbook and this appears as new trades in trade level data, with this metric we want to measure how deep this order eats into the orderbook. Below we plotted how this quote slippage accumulates over time for the same ticker - VIBBTC in the run up to the pump announcement, we also split slippage incurred by buy and sell sides. We can see from the Figure (4) that closer to the pump buy side seems to be more impatient executing trades incurring more losses to slippage, perhaps, implying more aggressive buying by insiders anticipating the pump.

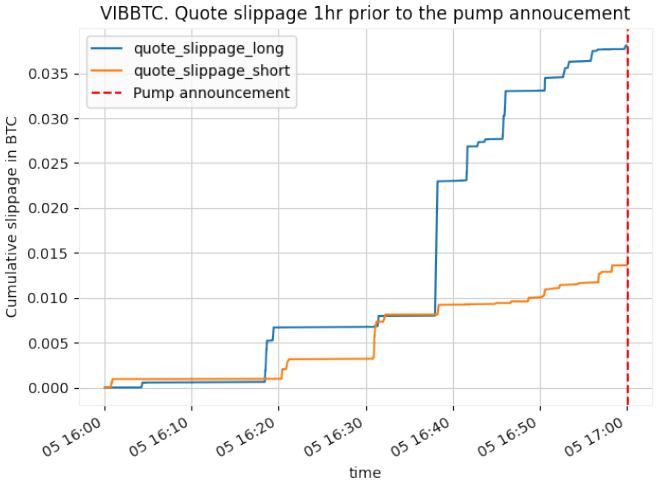

Figure 4: Cumulative quote slippage prior to the pump

2.3.3 Log returns features

We have also followed methodology of (Xu and Livshits 2019) and calculated features using a window of sizes (x+1, 1) where x is different offset in hours or days. Left bound indicates the start of the window, right bound of 1 indicates that we calculate features up to 1 hour prior to the announcement of the pump. This way we calculated mean and standard deviation of hourly log returns using windows described above. We hope that such approach could catch some anomalous hourly log returns which could point at insider buying before the pump.

2.3.4 Power law features. Quantile spread of order sizes

We also studied sizes of trades, we mostly were interested in big volume trades executed before the pump. Below we plot trades for the same ticker VIBBTC Figure (5).

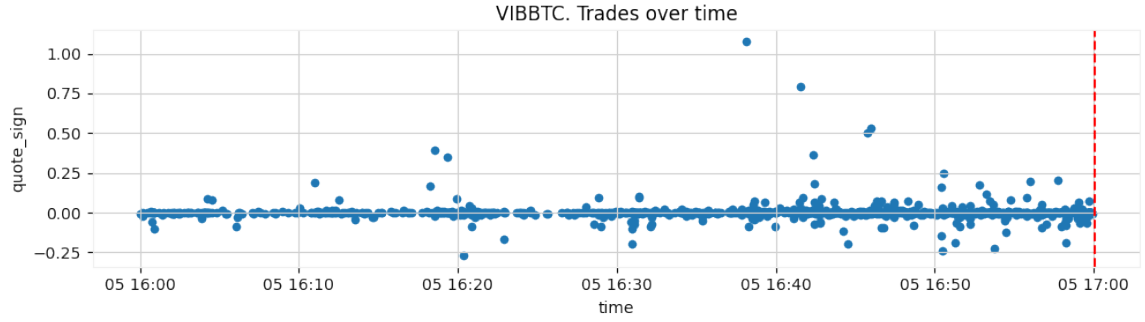

Figure 5: Order sizes in BTC before announcement

We can see that closer to the pump spread of order sizes rises with more trades significantly deviating from others in terms of trade size. In order to measure this we applied Power Law stating that data in higher quantiles has the following (3) probability density function:

$$f(x, \alpha) = \alpha x^{\alpha - 1} \tag{3}$$

Using this we estimated a value of $\hat{\alpha}_{\text{PL}}$ using samples of various sizes following the same idea of windows as described in the previous section. Since α is in the exponent of $f(x, \alpha)_{\text{pdf}}$, therefore lower values of alpha result in PDF decreasing slower which is a sign of heavier tails. This implies that there is a higher spread in upper quantiles showing the presence of abnormally big trades. In order to estimate alpha, we first filtered out all data below 95% quantile and estimated sample probability density function (PDF). Then, we regressed logarithm of estimated PDF against logarithm of *quote_ abs* feature:

$$\log \hat{f}_{\text{pdf}} = \underbrace{\log \alpha}_{\text{intercept}} + \underbrace{(\alpha - 1)}_{\text{slope}} \log \left(\text{quote\_abs}\right) \tag{4}$$

As a result of this procedure we took estimated slope which is $\alpha - 1$ and added back 1 to get $\hat{\alpha}_{\text{OLS}}$, then we used this estimate as a feature measuring spread in order sizes in upper quantiles. Below in Figure (6) we show how Powerlaw fits to observed data:

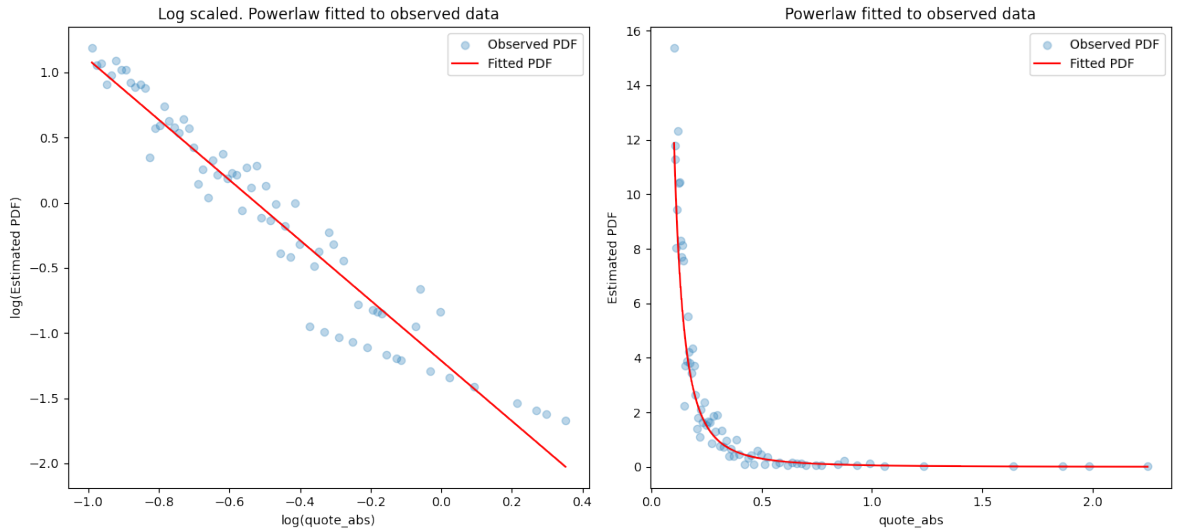

Figure 6: Powerlaw estimation

All of the features discussed are further explained in the Table (4) below:

Feature	Description	Notation
Overall return	log return of the asset calculated over the window of size (x+1, 1) hours prior to the pump announcement	$overall\_return$[x]h
Scaled average volume in BTC	average hourly volume in quote asset (BTC) calculated over the window of size (x+1, 1) hours and divided by standard deviation of hourly volumes calculated over the last 30 days	$volume\_zscore$[x]h_30d
Scaled average long volune in BTC	the same as above but everything is calculated using only long (buying) trade volumes	$volume\_long\_zscore$[x]h_30d
Scaled log return std	standard deviation of hourly log returns calculated over the window of (x+1, 1) hours and divided by 30 day standard deviation of hourly log returns	log_return_std[x]h_30d
Scaled log return mean	average hourly log return calculated over window of (x+1, 1) hours and scaled by 30-day standard deviation of hourly log returns	log_return_zscore[x]h_30d
Quote slippage share	share of overall slippages of this time window (y+1, 1) hours in total of 120 hours	$quote\_slippage\_share$[y]h_120h
Volume imbalance ratio	volume imbalance ratio computed over the window of size (y+1, 1) hours	$volume\_imbalance\_ratio$[y]h
Quote slippage imbalance ratio	imbalance ratio of slippages in BTC calculated over the same time window	$slippage\_imbalance\_ratio$[y]h
Powerlaw alpha	estimated alpha from powerlaw probability density function using the window of size (y+1, 1) hours prior to the pump	$powerlaw\_alpha$[y]h
Number of previous pumps	number of times this ticker has been pumped before within the last 90 days	num_prev_pumps
Days listed	number of days the ticker has been listed	$days\_listed$

Table 4: These are the features included in all models. These features are supposed to describe dynamics of price and trading volume in the run up to the pump. $x \in \{1, 6, 24, 48, 72, 168, 336\}$, $y \in \{1, 6, 24, 48, 72\}$

2.4 Cross-sectional approach

Since we deal with panel data and our data spans from early 2018 up to 2024, it is very reasonable to assume that trading activity has drastically changed over this period of time. Therefore, we propose using cross-sectional standardization in order to ensure that we remove market changes from modelling. We applied cross-sectional standardization, where for each numerical feature we subtract mean of this feature within the cross-section (within a given pump and dump cross-section) and divide by standard deviation of the same feature calculated

in the same way.

$$X_{\text{standardized}} = \frac{X - \bar{X}_{\text{within}}}{\bar{\sigma}_{X_{\text{within}}}}$$

By doing this, we are hoping to eliminate effects of market settings changes over the time period of the study which will allow classification models to better pick up patterns in the data. Below is the Figure (7) of data without and with this standardization for 5 different pumps:

Table 5: Pumps in Figure (7)

Pumped ticker	Announcement time
BRDBTC	2020-12-06 18:00:00
NEBLBTC	2020-03-15 20:00:00
BRDBTC	2018-12-22 17:00:00
NXSBTC	2020-03-03 15:59:00
KMDBTC	2018-09-20 20:11:00

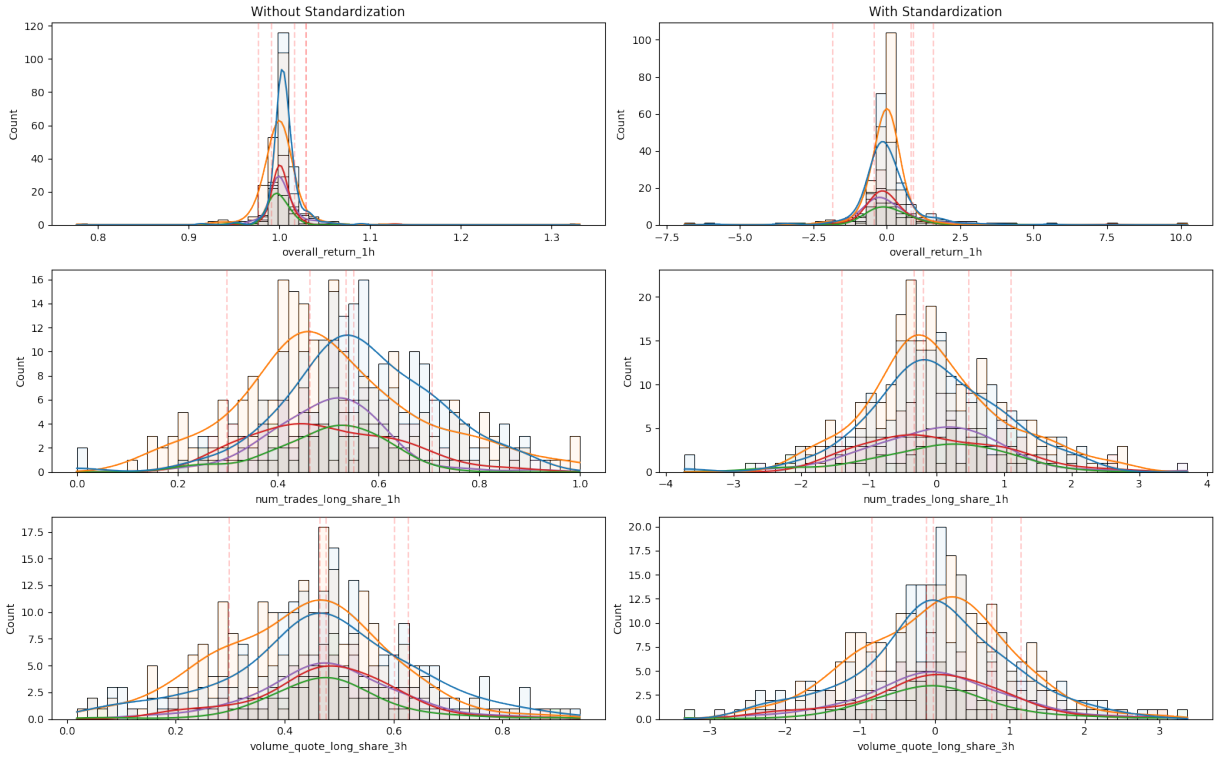

Figure 7: Features without and with cross-sectional standardization: We plotted histrograms for 5 different P&D events and also added KDE of PDF. Applying standardization helped to align distributions more closely with one another across pumps.

3 Classification models

In this section, we will attempt to build a model being able to predict pumps and dumps up to 1 hour prior to the announcement in telegram group chat. In the previous section, we have discussed data collection, aggregation and feature engineering. At this point, hopefully, we have the most sensible features that describe the pump and dump dynamics prior to the pump in the best way such that models we construct are able to pick up patterns in the data and label pumps with good enough performance. We are dealing with heavily imbalanced data, as pumps are very rare events as compared to regular trading activity. Descriptive statistics of this imbalance is shown in the Table (6) below:

Table 6: Cross-section size statistics

	mean	min	max	std
Cross-section size	113.258	34.000	231.000	41.143

This implies that on average we need to find 1 true positive among 113 alternatives. In this paper, we did not try resampling methods like Random Under/Over-Sampling, ADASYN, SMOTE like (Fantazzini and Xiao 2023) did in their paper but we will potentially look into that in the future. Prediction of pumps and dumps has become more difficult over time as exchanges are listing more and more tickers. This is also a challenge that we need to tackle which further complicates the problem of imbalance in our data. From Figure (8) we can clearly see this increasing trend:

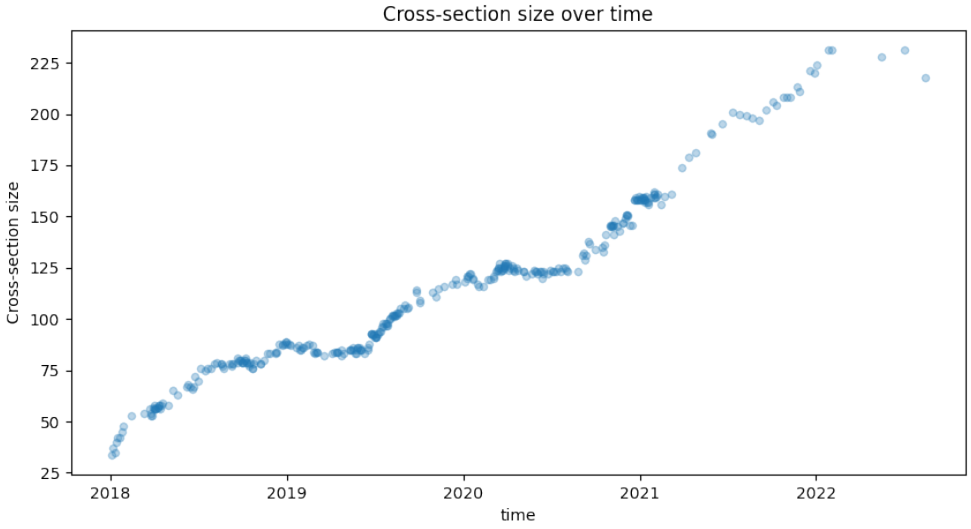

Figure 8: Binance has been increasing the number of tickers traded against BTC by listing more tokens. This tendency was likely to remain up until the end 2022 when more regulation came in with much stricter listing requirements.

3.1 Baseline model

We chose Logistic Regression - the simplest model as a baseline model to see if it outperforms the dummy classifier. By doing this we wanted to check if extracted features have any actual predictive power about future pumps. We split the data into train and test samples from 2018-01-03 to 2021-01-01 and from 2021-01-01 to 2022-08-14 respectively. The train sample contains

285 unique P&Ds while test has 52 of them. We left the model as is, the only thing that we did was adjusting class weight of the minority class to penalize more heavily for mislabelling of pumps. We trained the model using the whole train sample and were able to obtain already, in our opinion, good enough results, presented in Table (7). Our primary evaluation metrics are TOP-K accuracy with the same values of K as in Table (2) and TOP-K% AUC.

TOP-K accuracy measures the probability that among K highest logits produced by the model there will be a true pumped ticker. This metric by itself is not perfect since as we discussed earlier we have cross-sections of vastly different sizes, see Table (6). This results in higher TOP-K scores for cross-sections of relatively small sizes which might artificially inflate the metric, therefore, to control for this, we propose using TOP-K% as the main evaluation metric which shows the probability of hitting the true label among K% of the cross-section. In our opinion, this is more fair way to evaluate the model and we will later use this metric for parameter tuning and comparison of the models.

3.2 Finding the best model

In this section we go over how we trained and tuned other classification models. Following the Logistic Regression we also trained RandomForest with Gini loss for binary classification. We created our own version of KFold cross-validation which also takes into account cross-sectional structure of data, meaning that we can split data on the level of cross-sections rather than observations. In order to find optimal hyperparameters we used *Optuna* package which is a package implementing Bayesian optimization. We trained the model on all combinations of train, validation folds and computed TOP-K% AUC metric with $K \in \{0, 0.01, 0.02...0.99, 1\}$ as evaluation metric, then we took average across all folds as the final evaluation metric. Then, *Optuna* used this average TOP-K% AUC across folds to train multiple models with various hyperparameter values to find the set of ones that maximize this average metric. We also wanted to reduce the hyperparameter space as much as possible to reduce the chance of overfitting hyperparameters to the train sample. By doing all of this, we obtained better hyperparameters.

3.2.1 TOP-K accuracy

Following the methodology described above, we computed TOP-K accuracy for each model on the test set. The results are presented in the Table (7) below:

Table 7: TOP-K accuracy of models on test sample

	LogisticRegression	**Logistic Regression and Regulirization**	RandomForest	LGBMClassifier
TOP-1	0.115	0.135	**0.154**	0.135
TOP-3	0.288	**0.327**	0.192	0.192
TOP-5	0.404	**0.423**	0.231	0.269
TOP-10	0.481	**0.558**	0.404	0.404
TOP-20	0.673	**0.654**	0.577	0.519
TOP-30	0.731	**0.712**	**0.712**	0.558
TOP-50	0.788	**0.923**	0.769	0.654

We can see from the table above that Baseline Logistic Regression is holding up very well if compared to other more complex models like RandomForest and LGBMClassifier. The superior

model turned out to be tuned Logistic Regression outperforming other models by quite a margin in terms of almost all TOP-Ks. We added Elastic Net regularization to the Logistic Regression and tuned *l1-ratio* which controls shares of L1 and L2 penalties in the loss function.

3.2.2 TOP-K% accuracy

We also compared the models based on TOP-K% accuracy for different values of percentage threshold K%. This is a modification of the regular TOP-K allowing us to control for the problem of increasing cross-section sizes described above. Models were evaluated on the same test sample, the results are shown below in Table (8):

Table 8: TOP-K% accuracy of models on test sample

	LogisticRegression	**Logistic Regression and Regulirization**	RandomForest	LGBMClassifier
TOP-1%	**0.154**	**0.154**	**0.154**	0.135
TOP-5%	0.462	**0.519**	0.365	0.385
TOP-10%	0.596	**0.635**	0.558	0.500
TOP-20%	0.750	**0.846**	0.750	0.596
TOP-50%	0.904	0.942	**0.962**	0.865

We see that once again Logistic Regression with Elastic Net is performing the best in terms of almost all TOP-K%s. LGBMClassifier is still dominated by other simpler models, no matter how much we tried to set up early stopping, regularization and enforce shallower trees, we still didn't manage to get some decent results more or less comparable to other models.

3.2.3 TOP-K% AUC

In order to control for the problem of various cross-section sizes described above, we would want to compute TOP-K% instead of regular TOP-K. Taking area under the TOP-K% curve allows us to evaluate performance for all values of $K \in (0, 1)$. Below in Figure (10) we have compared the models based on this metric:

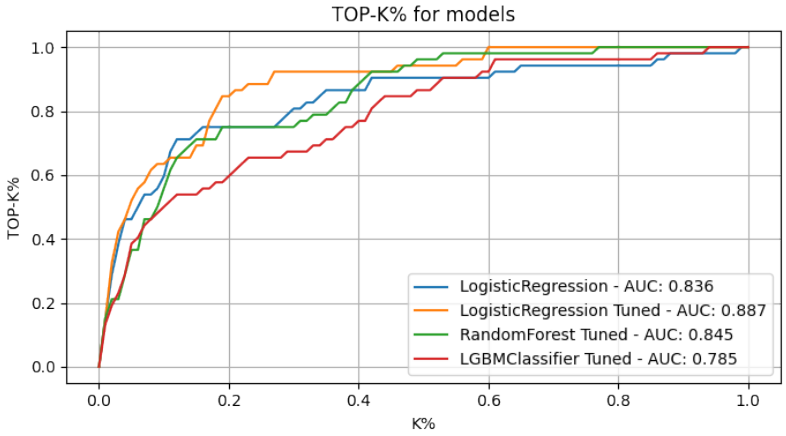

Figure 9: TOP-K% of models on test sample

We can see that the baseline Logistic Regression is not that much behind more complex models, all of the models have more or less the same performance if measured in terms of TOP-K% AUC.

3.2.4 Precision Recall Curve

We also used Precision Recall curve (PR) and PR-AUC to compare models as it allows to compare performance of classifiers without choosing any fixed probability threshold. We see that RandomForest is the best model in terms of PR-AUC with f1-score barely touching 0.2 f1-score for some threshold values. Tuned Logistic Regression is worse in terms of PR-AUC but it manages to maintain relatively high values of Precision for higher values of Recall, whereas RandomForest has very high precision for lower Recall values.

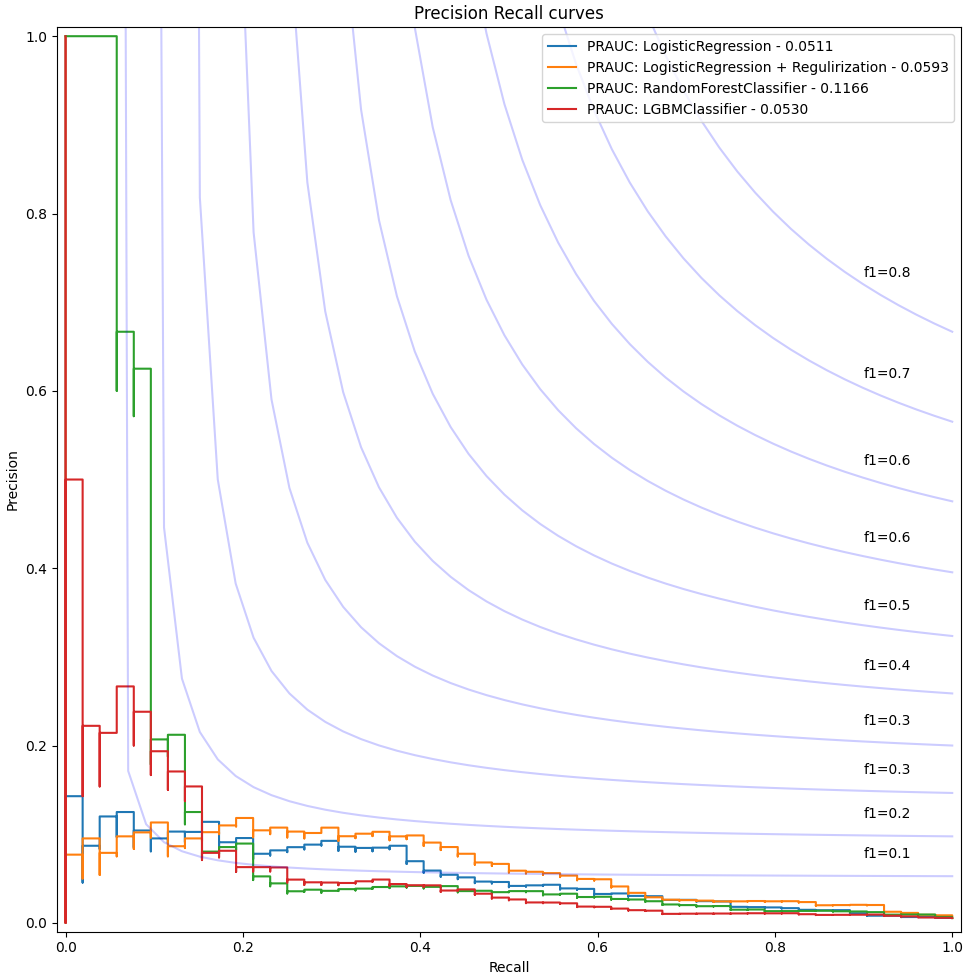

Figure 10: Precision Recall (PR) curve for models calculated on the test sample. PR-AUC - area under the Precision Recall curve.

3.3 Interpretation of the models

In this section we will move from comparison of the models and focus on interpretation of the obtained models. We will see which features had the most impact and try to explain why this makes sense in the context of prediction of P&Ds.

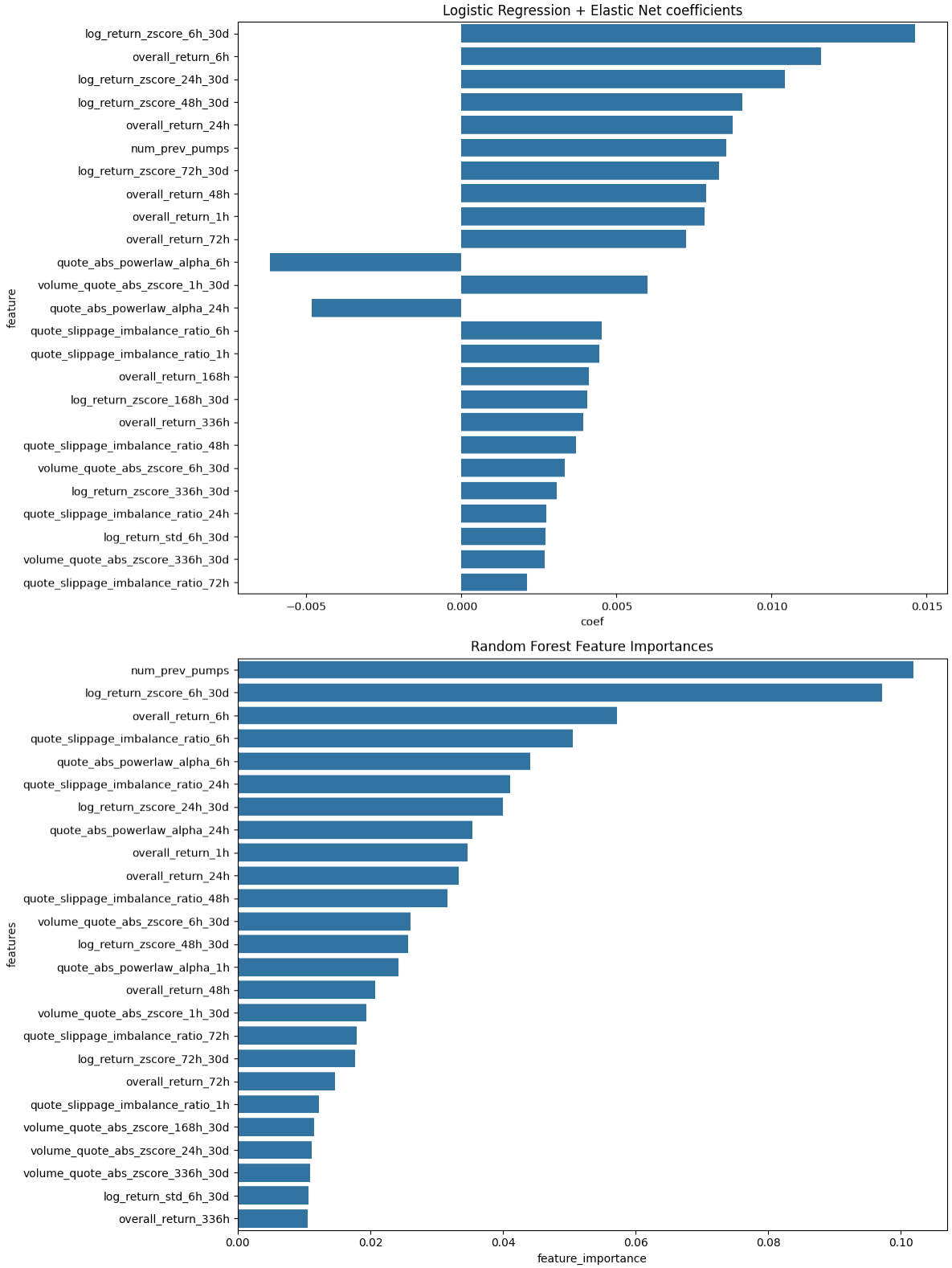

Figure 11: Features of the best models

We plotted highest in absolute value coefficients of the tuned Logistic Regression along with feature importances for Random Forest model. We see that in both models among the most important features are features like *num_prev_pumps*, features describing overall returns in the run up to the pump like *overall_return*[x]*h*, log returns features - *log_return_zscore*[x]*h*

4 Portfolio strategy

In this section we will propose a portfolio based strategy similar to the one proposed by (Xu and Livshits 2019), but we will differ in the way portfolios are created. (Xu and Livshits 2019) proposed using trained classification models to get proba for each asset in the cross-section, then construct a portfolio of assets that have proba of being pumped above some threshold. They also proposed using weights of the portfolio proportional to the logits produced by the model. (Xu and Livshits 2019) backtested their strategy using the following assumptions: assets that are not pumped do not change in price over the pump, authors buy at open price of the last hourly candle prior to the pump and are able to sell at pump's highest price (which coincides with high price of the next hourly candle). They backtested their strategy using the test sample from October 29, 2018 to January 11, 2019 and were able to achieve a return of 60% in measly 2 months. We would like to further work on this approach and apply our trained models to create portfolios that hopefully are able to capture pumps consistently.

We propose using models tuned for TOP-K% AUC which will ensure that the models are optimized not only for TOP-1 accuracy but for any arbitrary values of K. We relaxed some assumptions of (Xu and Livshits 2019) in the way we trade our portfolios, the timeline of how we trade each pump is described below:

Algorithm 1 Portfolio Methodology

1: Get logits from the classification model and sort them in descending order
2: Take the TOP-K logits or TOP-K% percentage of logits. These will be the tickers in the portfolio.
3: For each ticker, get buying (entry) price 5 minutes prior to the pump, buy assets based on weights of the portfolio (in our case equally weighted portfolio)
4: Hold the portfolio until the pump announcement. If the ticker is not pumped, then sell at the closest to pump price.

We decided to use tuned Logistic Regression as the primary model to create portfolios as it has superior performance as compared to other models in terms of TOP-K accuracy for lower values of K. We backtested the strategy using the steps described above for various portfolio sizes of K. Moments of portfolio return distributions are presented in the Table (9) below:

Table 9: Tuned Logistic Regression. Portfolio returns for various sizes K

Portfolio size (K)	min	mean	std	max
1	-0.2008	0.0360	0.2148	1.4217
3	-0.0538	**0.0637**	0.2607	**1.7293**
5	-0.0360	0.0498	0.1538	1.0112
10	-0.0149	0.0345	0.0795	0.5104
20	-0.0081	0.0197	0.0412	0.2670
30	-0.0042	0.0142	0.0276	0.1789
50	**-0.0021**	0.0102	**0.0172**	0.1141

We can see that smaller portfolios tend to outperform larger portfolios in terms of mean returns, this is because we spread out our position across less assets and therefore we are able to take advantage of correctly labelling the pumped ticker if it is in our portfolio. But this still increases the risk that we take on, as returns become more volatile and maximum draw-down also tends to increase with lower values of K.

Below, in Figure (12), we also plotted cumulative returns for each of the model for portfolios of size K=3. It follows from Table (7), that Logistic Regression is the best model as it boasts the return of 331.3 % over the period from 2021-01-01 to 2022-08-16. The lines are computed as the cumulative gain from investment into portfolios (without reinvestment of previous profits). We see that all models gained the most returns during the first part of 2021, this could perhaps be attributed to the bull run when bitcoin reached 65000$ for the first time in late April.

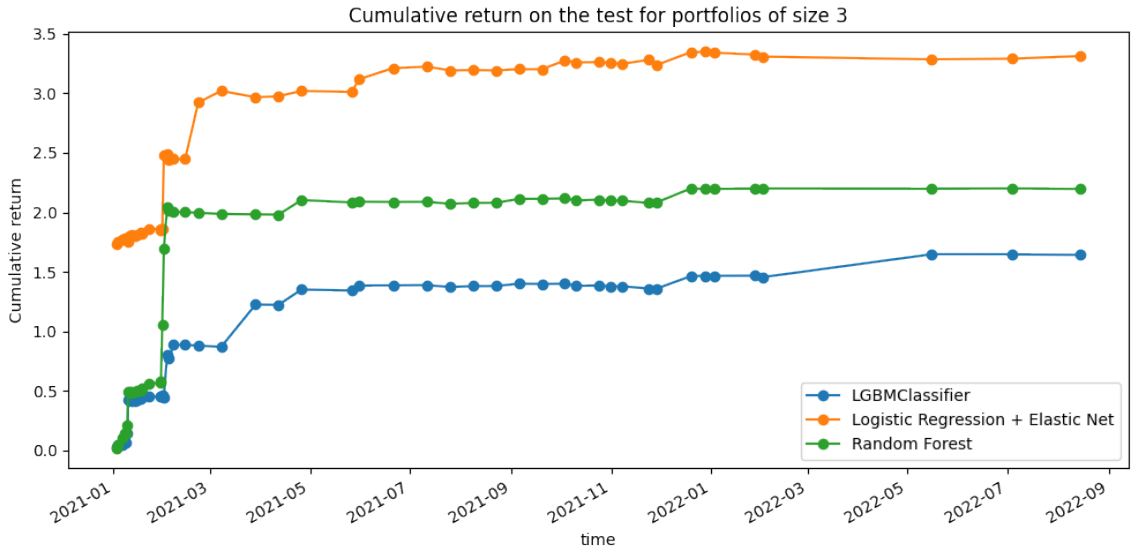

Figure 12: Cumulative returns on test test sample

5 Conclusions

In this paper we managed to achieve comparable results to papers by (Xu and Livshits 2019) and (Hu et al. 2023). We revisited the topic of P&Ds and showed that prediction of pumps and dumps in advance is possible and could be actually turned into a portfolio-based trading strategy. We backtested this proposed strategy and were able to obtain decent returns over the period of 1.5 years. The most interesting takeaway was that Logistic Regression turned out to be the best model in terms of TOP-K accuracy beating more complex models like Random Forest and LGBMClassifier by quite a margin. We attempted to use multiple boosting models but all of them no matter how hard we tried to prevent overfitting completely failed on the test sample. In the future we are planning to investigate cases of pumps and dumps on other exchanges, in 2024 admins of telegram groups still organize such fraudulent events but they migrated from Binance to new less regulated exchanges like Kucoin, Latoken and MEXC. In the future we would like to apply similar methodology laid out in this paper to pumps on other exchanges listed above. This will be much more difficult as the above exchanges have much more tickers traded at each time, therefore, it will be much more challenging to build classification models predicting pumps with the same TOP-K accuracy.

6 References

[1] V. Chadalapaka et al. *Crypto Pump and Dump Detection via Deep Learning Techniques*. May 2022. DOI: 10.48550/arXiv.2205.04646.

[2] D. Fantazzini and Y. Xiao. "Detecting Pump-and-Dumps with Crypto-Assets: Dealing with Imbalanced Datasets and Insiders' Anticipated Purchases". In: *Econometrics* 11.3 (2023). ISSN: 2225-1146. DOI: 10.3390/econometrics11030022.

[3] S. Hu et al. "Sequence-based target coin prediction for cryptocurrency pump-and-dump". In: *Proceedings of the ACM on Management of Data* 1.1 (2023), pp. 1–19.

[4] J. Kamps and B. Kleinberg. "To the moon: defining and detecting cryptocurrency pump-and-dumps". en. In: *Crime Science* 7.1 (Nov. 2018), p. 18. ISSN: 2193-7680. DOI: 10.1186/s40163-018-0093-5.

[5] M. La Morgia et al. "Pump and Dumps in the Bitcoin Era: Real Time Detection of Cryptocurrency Market Manipulations". In: *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. ISSN: 2637-9430. Aug. 2020, pp. 1–9. DOI: 10.1109/ICCCN49398.2020.9209660.

[6] M. La Morgia et al. "The doge of wall street: Analysis and detection of pump and dump cryptocurrency manipulations". In: *ACM Transactions on Internet Technology* 23.1 (2023), pp. 1–28.

[7] H. Nghiem et al. "Detecting cryptocurrency pump-and-dump frauds using market and social signals". In: *Expert Systems with Applications* 182 (Nov. 2021), p. 115284. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2021.115284.

[8] F. Victor and T. Hagemann. "Cryptocurrency Pump and Dump Schemes: Quantification and Detection". In: *2019 International Conference on Data Mining Workshops (ICDMW)*. ISSN: 2375-9259. Nov. 2019, pp. 244–251. DOI: 10.1109/ICDMW.2019.00045.

[9] J. Xu and B. Livshits. "The Anatomy of a Cryptocurrency Pump-and-Dump Scheme". In: *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 1609–1625. ISBN: 978-1-939133-06-9.

[10] J. Xu et al. *Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy*. 2022. arXiv: 2110.02642 [cs.LG].