LUISS

Degree Program in Finance

Course of Asset Pricing

Uncovering cryptocurrency pump-and-dumps with machine-learning

Prof. Nicola Borri

SUPERVISOR

Prof. Federico Carlini

CO-SUPERVISOR

Guglielmo Cuniato (756851)

CANDIDATE

Academic Year 2022/2023

Table of Contents

1. Introduction

The field of finance has witnessed a surge in fraudulent activities within the cryptocurrency market, particularly in the form of pump-and-dump schemes. These manipulative tactics, orchestrated by fraudsters, artificially inflate the price of a cryptocurrency before quickly selling off, resulting in substantial losses for unsuspecting investors. Pump-and-dump schemes have a long history in financial markets, but with the rise of cryptocurrencies, these manipulations have found a new breeding ground. Cryptocurrency pump-and-dump groups are primarily organized and operated through internet platforms such as Discord servers and Telegram channels. These platforms provide communication channels for group members, allowing them to coordinate and orchestrate these operations. The pump-and-dump process in the cryptocurrency market typically involves several steps. The organizers of the scheme announce the upcoming pump-and-dump operation, including the exchange where the manipulation will occur, and the exact start time. As the operation approaches, the admins repeat the announcements to create anticipation among members. When the pump starts, the target cryptocurrency is revealed to the members, and experiences a price surge almost immediately. The collective buying pressure then fades away in a few minutes and is replaced by a significant price crash during the dump phase.

The relevance of this research topic becomes evident when considering the economic impact of pump-and-dump schemes in the cryptocurrency market. The prevalence of such schemes has led to market instability and a loss of investor confidence. These manipulations create a volatile and unpredictable environment in which investors may experience huge financial losses. In this context, the first research question of this thesis aims to identify the key factors that contribute to the likelihood of success of these schemes. Analyzing these factors will shed light on the underlying dynamics and strategies employed by fraudsters. The second research question seeks to identify features that can effectively predict the cryptocurrency targeted in pump-and-dump schemes. The ability to predict the target cryptocurrencies will allow investors to make more informed trading decisions and avoid falling victim to these manipulations.

To address the research questions, this thesis follows a structured approach in three main sections. Initially, a short-term event study of pump-and-dump schemes is conducted on Kucoin Exchange to analyse the impact of these manipulations on the market. I then focus on training and comparing several machine-learning models to predict the target cryptocurrencies

of pump-and-dump schemes. Lastly, an algorithmic trading strategy based on the best machine-learning model identified before is back-tested to attempt to profit from pump-and-dump events. This analysis is conducted using market data in OHLCV (Open, High, Low, Close, Volume) format from Kucoin Exchange. Other information from third-party data providers CoinPaprika and LiveCoinWatch is also included.

The results reveal that pump-and-dump schemes exert a significant influence on cryptocurrency prices. The manipulation initiates with a substantial increase in prices, as indicated by an average abnormal return of 153.23% at the beginning of the event. The first minute fully captures the pump phase, with large positive returns and high volatility. The subsequent minutes consistently exhibit negative abnormal returns with abnormal returns up to -17.45%. Cryptocurrencies with lower market capitalization are associated to a higher magnitude of both positive and negative abnormal returns. Less liquid cryptocurrencies, instead, are related to higher abnormal return during the pump phase. The magnitude of abnormal returns decreases significantly after the tenth minute, suggesting the conclusion of the manipulation. Across the machine-learning models employed to predict the targeted cryptocurrencies in pump-and-dump schemes, the Random Forest model shows the highest performance. The age of the cryptocurrency, trading volumes, market capitalization, price level, and price momentum are identified as relevant predictors of pump-and-dump activity. Based on the Random Forest model, a trading strategy is defined as follows. The strategy involves opening long positions in the top 40 cryptocurrencies ranked by the model as potential pump-and-dump events, closing non-relevant positions upon the announcement of the targeted asset. The pumped cryptocurrency is then sold exactly one minute after the beginning of the manipulation. The findings from the back-testing process show impressive cumulative returns of 56.90% within a period of less than seven months.

Most of the results obtained during this analysis of pump-and-dump schemes agree with previous research in the cryptocurrency market. Notably, Hamrick et al. (2018) conducted a thorough examination of the factors that impact the success of pump-and-dump schemes in the cryptocurrency market. They focused on analyzing the percentage price increase of cryptocurrencies near the pump signal to assess the profitability of these schemes. One significant discovery they made was that market capitalization played a crucial role in determining the profitability of pump-and-dump schemes. Xu et al. (2019) also explored pump-and-dump processes in cryptocurrency markets and discovered that these schemes create

artificial demand and inflate buy volume. Moreover, the authors employed machine-learning techniques to predict the likelihood of a cryptocurrency being targeted in a pump-and-dump scheme. The findings of their analysis revealed that the Random Forest Model outperformed Generalized Linear Models, displaying reasonable accuracy in predicting such schemes. Furthermore, the authors identified several key features that serve as predictors of pump-and-dump activity, including low market capitalization and high returns before the manipulation. La Morgia et al. (2020) built and publicly released the first dataset of confirmed pump-and-dump events on several cryptocurrency exchanges. Based on this data, the authors focused on detecting pump-and-dump schemes in real-time on Binance exchange. Their results identified buy market orders and trading volumes as relevant features for the detection of on-going manipulations.

It is important to note that most of previous research in this area has been conducted on cryptocurrency exchanges where pump-and-dump events are no longer prevalent. Therefore, this thesis addresses this research gap by focusing on the latest available data from KuCoin Exchange, where pump-and-dump schemes continue to occur. By identifying key features and evaluating the effectiveness of different machine learning models, this research enhances our understanding for predicting and preventing these manipulations. The limitations of this work, such as data availability and other considerations, are acknowledged, and suggestions for future work are discussed in the Conclusions section. The remainder of this thesis is organized as follows: Chapter 2 provides a thorough review of the literature, presenting an overview of previous studies and highlighting their contributions to the field. Chapter 3 discusses in detail the data collected for this research and the process of feature engineering applied to extract relevant information. Chapter 4 details the methodology employed in this research, including the short-term event study to analyse the anatomy of pump-and-dump events and the implementation of machine-learning models. Chapter 5 presents and discusses the results obtained from the analysis, evaluating the performance and accuracy of the models. Chapter 6 concludes the thesis, summarizing the key findings, discussing the limitations, and suggesting avenues for future research in this area.

2. Literature Review

The phenomenon of pump-and-dump schemes has a long history predating the emergence of cryptocurrency. Consequently, a substantial portion of the existing literature focuses on pump and dumps executed in the traditional stock market. Allen et al. (1992) categorized market manipulation schemes into three types: information-based, action-based, and trade-based. Information-based manipulation consists of spreading false information about the targeted security in hope that uninformed traders will act based on such distorted information. This requires uncertainty about the fair value of the security and the existence of information asymmetry. Trade-based manipulation, instead, involves executing transactions on a security to create distortions in supply and demand dynamics. Pump-and-dump schemes in traditional markets typically involve a combination of information-based and trade-based manipulation.
Aggarwal and Wu (2006) contributed to the understanding of stock price manipulation through a combination of theoretical analysis and empirical evidence. The authors examine manipulative trading and find that throughout the manipulation period, prices tend to rise, followed by a decline after the fraudulent activity ends. Moreover, manipulation leads to increased volatility, liquidity, and returns. Bouraoui (2015) suggested that a low level of liquidity can substantially increase the risk for securities to be susceptible to pump-and-dump manipulation schemes. Austin (2018) pointed out the significant issue of lack of liquidity for markets designed to trade the securities of smaller companies. Lin (2017) discussed the detrimental effects of several manipulation techniques, such as front-running, benchmark distortion, and pump-and-dumps. He emphasized the importance of prompt regulatory intervention, particularly for those securities that are more vulnerable to these schemes: cheaply priced securities, traded in less regulated, over-the-counter markets, such as the "penny stocks".

The academic literature on pump-and-dump schemes in the cryptocurrency markets is still relatively limited, mainly because of the nascent nature of this market and the associated challenges when conducting research. However, there have been several recent studies that have made significant contributions to analysing, detecting, and predicting these schemes using both traditional and machine learning techniques. This literature review will be divided into the following three subsections, each focusing on a specific research area related to pump-and-dump schemes in the cryptocurrency markets: analysis of the event's anatomy, pump-and-dump event detection, and target coin prediction.

2.1. Anatomy of pump-and-dump events

Hamrick et al. (2018) conducted a detailed analysis of the factors influencing the success of pump-and-dump schemes in the cryptocurrency market. Their study specifically measured the percentage price increase of cryptocurrencies near the pump signal to evaluate the profitability of these schemes. One key finding of their research was that the rank of the targeted coin, evaluated based on market capitalization or trading volume, played a crucial role in determining the profitability of pump-and-dump schemes. The authors found out that targeting less prominent and more obscure coins was more profitable compared to targeting dominant coins in the cryptocurrency ecosystem. Hamrick et al.'s study focused on investigating the factors affecting the success of these schemes, providing valuable insights into the anatomy of pump-and-dump activities in the cryptocurrency market.

Xu et al. (2019) also delved into pump-and-dump frauds in cryptocurrency markets: they found out that these schemes induce fake demand and inflate buy volume. Moreover, the authors pointed out that most investors do not manage to act fast enough to sell the pumped token at a higher price, ending up selling at a loss or holding a virtually worthless coin. Small volume movements shortly before the pump-and-dump hours were also detected, and could be indicative of the organizers' pre-pump activity.

In a separate comprehensive study, Dhawan et al. (2021) examined the prevalence of pump-and-dump schemes across various cryptocurrency markets. Analysing a sample of 355 cases over a six-month period, the authors investigated the characteristics and impact of these phenomena. One significant observation was that cryptocurrency manipulators openly declare their intentions to pump specific coins rather than attempting to deceive investors covertly, as is often seen in stock market manipulation. Additionally, their study revealed that pump-and-dump events result in substantial price distortions, with an average distortion rate of 65%, along with abnormal trading volumes reaching millions of dollars. These events also facilitate significant wealth transfers among participants. Moreover, Dhawan et al. identified social media sentiment, liquidity, and trading volume as significant predictors of pump-and-dump activity. Their research primarily focused on the impact of these schemes on the market, providing valuable insights into the dynamics of these activities in cryptocurrency exchanges.

2.2. Pump-And-Dump Event Detection

Kamps et al. (2018) conducted a ground-breaking study that introduced the concept of cryptocurrency pump-and-dump schemes and defined their life cycle, setting them apart from traditional pump-and-dump events observed in stock markets. Their primary objective was to identify abnormal changes in trading volume and price, thereby enabling the detection of suspicious activities within the cryptocurrency market. To accomplish this, the authors employed statistical analysis techniques and obtained valuable findings. They found that pump-and-dump events within the cryptocurrency market exhibit distinct patterns characterized by a rapid surge in both trading volume and price, followed by a sharp decline. The insights gained from Kamps et al.'s research shed light on the behavioural characteristics of pump-and-dump schemes.

Building upon this foundation, La Morgia et al. (2020) focused on real-time detection of pump-and-dump schemes on the Binance exchange. Their study employed unsupervised learning techniques, specifically utilizing K-means clustering, to identify abnormal trading activity indicative of pump-and-dump schemes. In order to evaluate the effectiveness of their classifier, La Morgia et al. (2020) compared its performance with the detection method proposed by Kamps et al (2018). The results of their study demonstrated that the K-means clustering classifier outperformed the previous detection method, exhibiting significantly higher performance scores, particularly in terms of the F1-score. Therefore, La Morgia et al.'s research falls within the category of detecting pump-and-dump events and represents a notable advancement in this area.

More recently, Chadalapaka et al. (2022) proposed an innovative deep learning-based approach for detecting pump-and-dump schemes in cryptocurrency markets. Their study focused on the development of advanced techniques capable of accurately identifying such schemes. Specifically, they employed the C-LSTM model and the Anomaly Transformer for time-series anomaly detection. The authors reported promising results, with precision and recall metrics exceeding 90% in their study. Chadalapaka et al.'s research significantly contributes to the detection aspect of pump-and-dump schemes, aligning it with the work of Kamps et al. and La Morgia et al. By utilizing deep learning algorithms and incorporating advanced anomaly detection methods, their approach offers a valuable tool for effectively identifying and combating pump-and-dump schemes in the cryptocurrency market.

2.3. Pump-and-dump Target Coin Prediction

Xu et al. (2019) conducted an in-depth analysis of pump-and-dump schemes in the cryptocurrency market, employing machine-learning algorithms to predict the specific cryptocurrencies targeted in these events. They examined trading data of 412 fraudulent events from June 2018 to February 2019 and identified various patterns indicative of coordinated buying and selling activity. The authors found that pump-and-dump schemes are highly profitable for their organizers at the expense of unsuspecting investors. They applied machine learning techniques, namely Random Forest Algorithm and Regularized Logit Regression, to predict the likelihood of a given cryptocurrency being targeted by a pump-and-dump scheme. Specifically, the machine-learning models were trained on a dataset containing 180 pump-and-dump events organized on the Cryptopia exchange. The results of their analysis demonstrated that the Random Forest Model outperformed Generalized Linear Models and exhibited reasonable performance in accurately predicting such schemes. Additionally, they identified several key features that are predictive of pump-and-dump activity, including low market capitalization and high social media activity. Among all features related to market movement, return features seemed generally more valuable than volatility or volume variables. Instead, exchange-specific data, such as trading and withdrawal fees, did not carry relevant information to predict the cryptocurrency targeted by the manipulators. Hence, Xu et al.'s work encompasses both the prediction of coin targets and the analysis of the underlying patterns in pump-and-dump schemes.

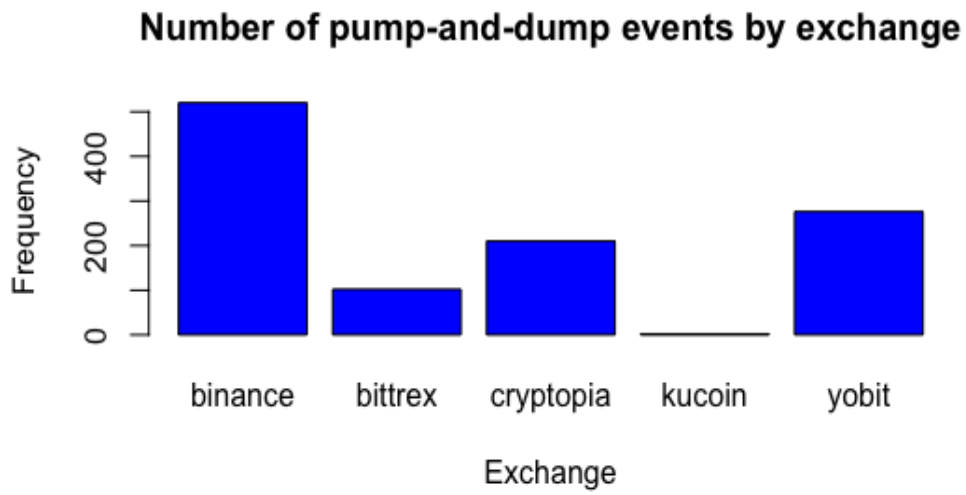
Nghiem et al. (2021) focused their study on predicting these schemes on Binance using deep learning methodologies. The researchers gathered data from various resources, including Cryptocompare.com, a large and comprehensive data repository for cryptocurrency markets. They created a dataset of market and social media data in the hours preceding each pump, and compared a cross-selection of deep learning models consisting of convolutional and recurrent neural networks for predicting the coin target of the fraud. This study highlighted the potential of utilizing deep learning techniques to predict pump and dump schemes, showcasing the effectiveness of CNN and B-LSTM models in achieving accurate predictions. Both these model specifications outperformed the baseline Logit Regression Model in terms of F1 Score.

This area of work, namely pump-and-dump target coin prediction, mainly represents the research field in which this thesis aims to be positioned.

3. Data

Crypto-currency pump-and-dump schemes are considered relatively rare occurrences in the market. As highlighted by the work of Kamps et al. (2018), at the time it did not exist a reliable dataset of confirmed pump-and-dump events in the literature, making it challenging to gain a comprehensive understanding of these activities. La Morgia et al. (2020) addressed this gap by constructing a dataset of 1110 pump-and-dump schemes organized on 5 different cryptocurrency exchanges and releasing it to the public on GitHub. Their data was collected using Telegram APIs to perform scraping of text messages from 20 Telegram groups that organized and led these activities from 2017 to 2021. The following picture presents a visualization of their data comparing the number of confirmed pump-and-dump occurrences by cryptocurrency market.

Figure 3.1. Frequency bar-chart of pump-and-dump events by exchange in La Morgia et al.'s dataset.



Across all the exchanges included in the dataset, Binance shows the highest fraudulent activity in terms of number of organized pump-and-dump schemes, with an overall count of 520 events. Kucoin Exchange, instead, presents the lowest number of occurrences: only 2 events organized in 2019 and 2020. The fact that Binance is the market on which most pump-and-dump schemes were led is not surprising, as this exchange has a very large user base, and consistently ranks among the most famous cryptocurrency markets with relevant trading volumes according to

the data provider CoinMarketCap. Kucoin, on the other side, is a relatively newer exchange and does not reach the same levels of Binance or Bittrex in terms of trading activity.

Building upon the dataset provided by La Morgia et al. (2020), I perform an analysis of the activity of 27 Telegram groups, which included the original dataset's channels and also 7 newer Telegram groups. Our results suggest that all exchanges included in La Morgia et al. 's dataset, made exception for Kucoin Exchange, are no longer being involved in the organization of pump-and-dump schemes in the cryptocurrency market. Kucoin and HotBit seem to be the most common cryptocurrency exchanges in which these fraudulent events are nowadays conducted since 2021, and the organization of the related pump-and-dump schemes can be tracked down to these two Telegram groups: "Hotbit Crypto Pumps", and "Kucoin Binance Pumps Trading". Notably, KuCoin and HotBit offer a significantly larger selection of tradable cryptocurrencies compared to Binance, as reported by the data aggregator CoinGecko. This substantial difference in the number of available assets could potentially account for the shift of pump-and-dump organizers from Binance to these emerging markets. The increased variety of coins in these platforms expands the pool of potential targets for pump-and-dump schemes, providing organizers with more options to carry out their activities.

Compared to HotBit, Kucoin Exchange is the only cryptocurrency market offering a reliable system of APIs that allows to programmatically retrieve real-time and historical market data. Therefore, I decide to focus this research on Kucoin and expand the current available dataset of confirmed pump-and-dump events on this exchange. Precisely, I perform text message scraping on the Telegram group "Kucoin Binance Pumps Trading" through a custom Python script based on official Telegram APIs in order to identify the announcements of pump-and-dump events. The data collected allows to expand the existing pump-and-dump dataset with 55 new observations of pump-and-dump schemes organized on Kucoin Exchange from April 2021 to May 2023. Adding these observations to the 2 pump-and-dump events on Kucoin confirmed by La Morgia et al. 's dataset, I identify a total of 57 pump-and-dump schemes which constitute the starting point of the data collection process for this analysis.

This Data Chapter is divided into two subsections: the first regarding the data preparation process for the event study methodology, and the second about the data preparation and feature engineering processes involved in machine-learning models.

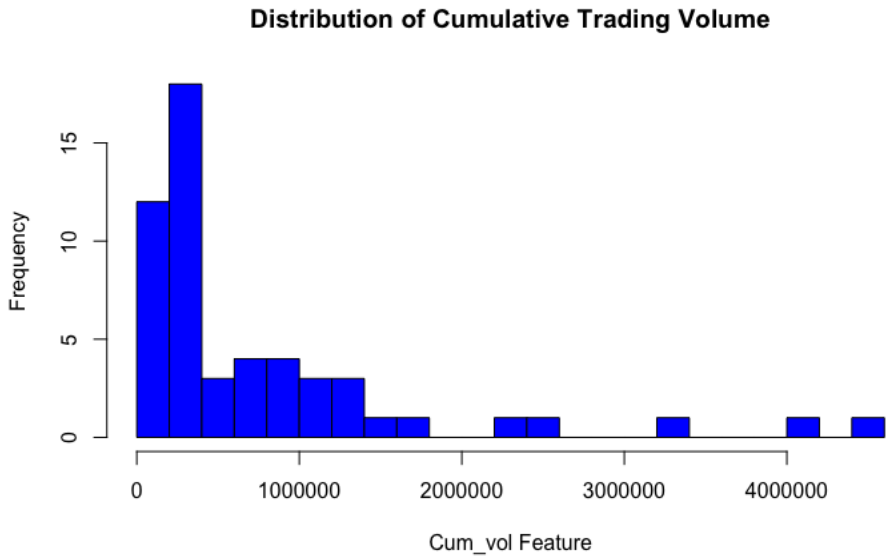
3.1. Event Study Data Preparation Process

The first step of the empirical design of this thesis consists of investigating the anatomy of pump-and-dump schemes through a short-term event study to dig into their impact on the cryptocurrency market. Through this analysis method, I am mainly interested in evaluating abnormal returns following the pump-and-dump events. This data is then related to both market and asset-specific data to analyse potential significant relationships.

Almost all features in this dataset are evaluated based on candlestick data retrieved from Kucoin Exchange APIs. Specifically, I gather candlestick data at 1-minute frequency for each fraudulent event. A higher frequency, such as 1-second frequency, would be preferable to track closer the development of these events: unfortunately, 1-minute is the highest frequency provided by Kucoin APIs. The timeframe chosen for the data goes from 14 days preceding the pump-and-dump event to 2 hours after, reaching a total of 338 hours of candlestick data per pump-and-dump event. Since the data is sampled on a per-minute basis, the event study dataset contains 20,280 observations for each scheme, and a total of 1,155,960 observations including variables in OHLCV (Open, High, Low, Close, Volume) format. All prices and volumes are quoted on the cryptocurrency Tether USDT, a stable-coin pegged to the price of US Dollar. In fact, all pump-and-dump events identified so far on Kucoin Exchange have been conducted on this trading pair. On the basis of retrieved market data, I compute 1-minute returns using opening and closing asset prices. I then build two indicators based on volumes which may carry information about the short-term liquidity and trading popularity of each asset: *Cum_vol* and *Std_vol*. Precisely, the first feature is computed as the sum of 1-minute trading volume in US dollars for each asset considering volumes from 7 days before the pump-and-dump time. According to the same timeframe, I evaluate the standard deviation of 1-minute trading volumes to build the second feature. I also build two rank features, namely *Cum_vol*_rank and *Std_vol_rank* based on values of *Cum_vol* and *Std_vol*, ranking each pump-and-dump observation in ascending order from 1 to 57. I believe these indicators may be exploited as a proxy for short-term liquidity and market interest in the targeted cryptocurrencies. Higher trading volume typically indicates a higher level of market activity, which in turn may lead to greater liquidity. At the same time, higher volume volatility indicates that trading volumes vary more wildly from the average, suggesting periods of increased trading activity or changes in investors' sentiment. The data described so far constitutes market data retrieved directly from Kucoin Exchange, and do not contain any missing value.

The final dataset for the event study analysis also includes data from third-party providers, specifically this research relies on the *Coin_rank* feature provided by the market research platform CoinPaprika. This variable is constructed by assigning a rank to each cryptocurrency out of 9,336 total assets listed on the platform for which market capitalization is known. For each cryptocurrency, higher rank indicates a lower market capitalization. I retrieve the coin rank feature for all available cryptocurrencies on CoinPaprika, and then merge this data to all pump-and-dump events' observations. This process generates missing data for observations related to 12 fraudulent schemes. In fact, there are 10 cryptocurrencies, two of which have been pumped twice, for which CoinPaprika has no available information on market capitalization and, consequently, on coin rank. The following picture reports a histogram describing the distribution of the first feature of the final event study dataset, namely *Cum_vol*:

Figure 3.2. Histogram of Cum_vol feature: cumulative trading volume from 7 days before the pump



The variable's distribution exhibits a pronounced right-skewness, indicating that a significant portion of cryptocurrencies involved in pump-and-dump events are traded with overall volumes below 500,000 US dollars during the seven days before the fraud. As a comparison, Bitcoin and Ethereum, two of the most relevant cryptocurrencies, have generated trading volumes in the last 24 hours of respectively 55,000,000 and 18,000,000 US dollars on Kucoin Exchange. Such a difference in trading volumes is remarkable and may suggest a preference for pump-

and-dump schemes to target coins that receive less trading attention and are relatively less liquid. The distribution of the other variables in the dataset strengthens such a possibility, and is reported in the following table of descriptive statistics:

Table 3.1. Descriptive statistics of features in event study dataset

Feature	***Min.***	***1st Qu.***	***Median***	***Mean***	***3rd Qu.***	***Max.***
Cum_vol	39,323	218,375	379,795	774,630	981,959	4,425,877
Std_vol	42.04	152.33	240.07	416.22	476.81	2,788.34
Coin_rank	164	1,162	1,269	1,268	1,417	1,880
Cum_vol_rank	1	15	29	29	43	57
Std_vol_rank	1	15	29	29	43	57

As depicted in the table above, the distribution of the *std_vol* feature is also highly right-skewed since the median is almost the half of the mean. This indicates that in most pump-and-dump events there was relatively not much variation in the trading volume levels in the hours preceding the fraud. Such observation is particularly noteworthy considering the low amount of trading volumes showed in Figure 3.2. In fact, this suggests that cryptocurrencies targeted in pump-and-dump schemes are not actively traded on the market before the event. Moreover, we can notice that 75% of all pumped cryptocurrencies for which coin rank feature is available are ranked in a range between positions 1162 and 1880. According to CoinPaprika data, the cryptocurrency currently ranked at position 1000 has a market capitalization slightly higher than 1 million dollars, precisely 1,075,794 US dollars. This information indicates that most pump-and-dump schemes on Kucoin have targeted low-cap coins, since higher rank corresponds to lower market capitalization.

3.2. Machine-Learning Models Data Preparation Process

The second step of the empirical design of this thesis consists of training and testing several machine and deep learning models to predict in advance the cryptocurrencies targeted in pump-and-dump schemes. Currently, Kucoin Exchange offers information for 3060 different cryptocurrencies, out of which only 762 coins are officially listed for trading on this market. Whenever a pump-and-dump event is organized on Kucoin Exchange, the manipulators choose exactly one coin out of all listed cryptocurrencies and inflate its price. Consequently, all tokens listed for trading on this exchange represent potential targets. This observation constitutes the starting point for building the machine-learning models' dataset.

For each pump-and-dump event's date confirmed in our dataset, I retrieve candlestick data at hourly frequency for all 762 listed cryptocurrencies via Kucoin Exchange APIs. The timeframe chosen for the data includes the 14 trading days preceding each event time, reaching a total of 336 hours of candlestick data per observation including variables in OHLCV (Open, High, Low, Close, Volume) format. All prices and volumes are quoted on the stable-coin Tether USDT, as all known pump-and-dumps on Kucoin Exchange have been organized on this trading pair. Since our original dataset contains 57 pump-and-dump events and there are 762 potential targets each time, the machine-learning models' dataset reaches an overall size of 43,662 observations. There are only 57 observations related to pump-and-dump events, while the rest of the observations represent ordinary trading events of cryptocurrencies which have not been targeted in these frauds.

The market data collected through Kucoin Exchange APIs does not constitute our complete dataset yet, in fact I also rely on third-party data providers CoinPaprika and LiveCoinWatch. Specifically, I include again the *Coin_rank* feature based on market capitalization provided from the first provider as in the previous section. LiveCoinWatch, instead, provides the information regarding the *Age* of the cryptocurrency measured in days, and the number of E*xchanges* in which the cryptocurrency is traded at. I believe these features may reflect asset-specific characteristics that pump-and-dump fraudsters carefully consider when choosing the target coin of their manipulation. Note that the missing data coming from *Coin_rank* feature in the whole dataset has been replaced with the third quartile of the same feature but computed only for pump-and-dump observations. In fact, I believe *Coin_rank* to be positively related to

the likelihood of pump, so this choice encourages the models to consider all data for which *Coin_rank* feature was missing as potential pumps.

The last step of the machine-learning models' dataset construction consists of a process of feature engineering based on the candlestick data retrieved via Kucoin Exchange APIs. I decide to do so to extract relevant features from market data which may be indicative of "pre-pump" activity led by manipulators or insiders. As showed in the work of Xu et al. (2019), several pump-and-dump events show some patterns during the days before the fraud: these can be unusual volume movements suggesting organizers' pre-purchase conduct, and also upward trends in the asset price which may reflect an increasing buying pressure from insiders. The figures below present as a case study the trading volumes and opening prices associated to the pump-and-dump of the cryptocurrency BiFi which was led on Kucoin on October 15th, 2022:

Figure 3.3. Hourly trading volumes of cryptocurrency BiFi in USDT during the 14 days before the pump event.

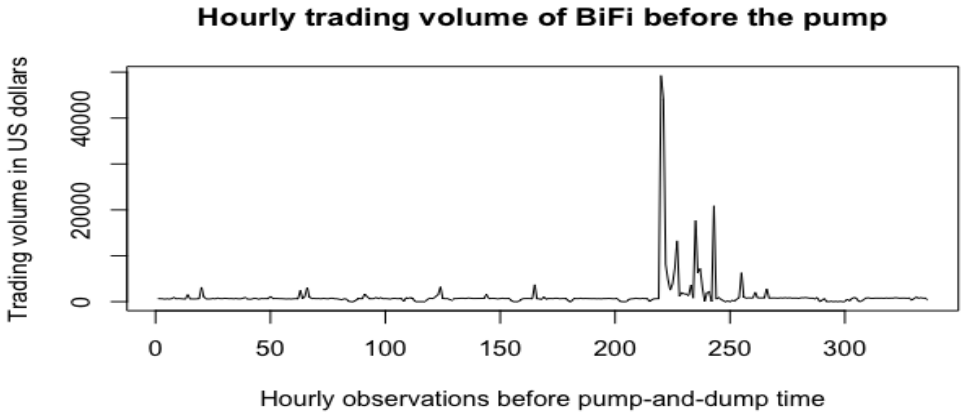


Figure 3.4. Hourly opening prices of cryptocurrency BiFi in USDT during the 14 days before the pump event. Blue dashed line represents OLS fitted line to capture the trend of the time-series.

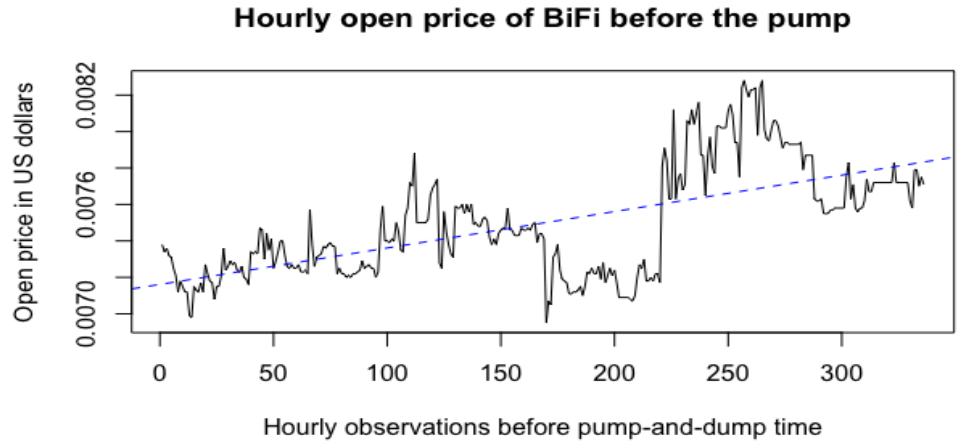


Figure 3.3 clearly shows an abnormal volume movement of almost 45,000 US dollars in correspondence of hour 219, approximately 5 days before the scheduled pump-and-dump event. The volume peak in this graph is then followed by smaller perturbations of amounts up to 20,000 US dollars until around hour 250, which represents slightly more than 3 days before the pump. These volume signals seem to be unusual given the low trading activity associated to the hours before, and also given their proximity to the pump-and-dump event. Moreover, these trading volumes are supported by an upward trend in the opening asset prices during the same timeframe as depicted in Figure 3.4. The blue dashed line in this plot is fitted on the time-series of opening prices and captures a relevant upward trend in the BiFi price during the 14 days before the pump.

The existence of these patterns encourages me to reshape the market data retrieved previously and build more useful features for the development of machine-learning models. Based on the candlestick data in OHLCV format I build three sets of features: the first is focused on the time-series of opening prices to capture relevant price trends; the second set of features is based on trading volumes to detect unusual volume movements; the last set includes rank features based on both price and volumes to discriminate among coins with different characteristics. All features present in the first set are based on the Relative Strength Index (RSI), a popular technical indicator in financial analysis to measure strength and momentum of price movements. Specifically, I computed the RSI indicators on the time-series of opening price considering many different timeframes before each fraudulent event. The highest correlations between the RSI and pump events are found when the RSI is evaluated on the prices over the six hours before the pump, and from 72 to 48 hours before the pump. Consequently, I insert these two features to analyse the price movements in our dataset, and respectively name them as "*Last_6_RSI"* and "*Middle_RSI"*. The second set of features is based exclusively on trading volumes, and includes two equivalently defined variables that aim at detecting unusual volumes. These indicators are constructed as the ratio of the highest hourly trading volume to the mean hourly trading volume in US dollars. Again, as in the first set, I create these variables based on many timeframes: the highest correlations between these indicators and pump events are found when the ratio is computed on data from 6 to 4 days before the pump, and from 14 to 13 days before the pump. In our dataset these two features take the name of, respectively, "*Vol_ratio1"* and "*Vol_ratio2"*. The last set of features includes coin ranks based on the opening price exactly 14 days before the pump, and on the cumulative trading volumes from 14 to 13 days before the pump. I computed these rank indicators voluntarily on the most distant

days from the pump to avoid including the effects of potential pre-pump conduct. Specifically, I performed this ranking procedure among all the coins available for trading on Kucoin during each pump-and-dump. Since there is a total of 762 cryptocurrencies officially listed on this exchange, each coin receives two ranks, one based on price and the other based on volume, with rank positions ranging from 1 to 762. The higher the rank assigned, the greater is the opening price or trading volume associated to the cryptocurrency. These two rank features are then included in the dataset with the variable names *Price_rank*, and *Vol_rank*. At this stage, the process of dataset construction for machine-learning models is complete. I discriminate between observations involved in pump-and-dumps schemes and ordinary trading observations, and report summary statistics for each of these categories in the table below:

Table 3.2. Descriptive statistics of features in machine-learning models' dataset computed by observation class.

	Pump-and-dump observations						***Ordinary trading observations***					
Feature	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**	**Min**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
Last_6_RSI	31.94	49.39	55.38	55.96	61.13	82.86	10.73	44.49	49.12	49.25	54.03	89.87
Middle_RSI	19.65	42.13	49.66	50.03	58.15	70.39	3.094	42.148	48.039	47.446	52.860	93.234
Vol_ratio1	1.387	3.444	5.339	6.187	7.671	18.260	1.051	2.460	3.519	4.306	5.293	19.980
Vol_ratio2	1.268	3.617	8.181	9.986	15.911	24.670	1.103	3.463	5.268	6.772	8.295	50
Price_rank	12	108	185	212	287	552	1	156	318	326	488	762
Vol_rank	4	43	85	160.3	232	550	1	157	318	326.9	488	762
Coin_rank	164	1,162	1,269	1,268	1,417	1,880	1	182	613	1,562.4	2,632	8,837
Age	1	576	710	711.7	816.5	2103	1	706	960	1,208	1,637	5,262
Exchanges	1	1	3	4.216	5	24	1	2	7	22.45	23	333

The table of summary statistics reported above easily allows the reader to comprehend the main characteristics of pump-and-dump events. In fact, all the descriptive figures associated to each feature have been evaluated separately for pump-and-dump observations and ordinary trading observations: this approach permits to directly compare the statistics between these two groups. The first two features *Last_6_RSI* and *Middle_RSI* capture trend and momentum in the asset prices, and both their mean and median are remarkably higher in the pump-and-dump class rather than in ordinary observations. Particularly, the mean of *Last_6_RSI* in pump-and-dump events is almost 7 points higher compared to the other class, confirming that price momentum in the six hours preceding the pump is a common pattern in pump-and-dumps belonging to our dataset. Features *Vol_ratio1* and *Vol_ratio2* aim at capturing unusual trading volume movements in the hours before the pump. As before, both their mean and median values in the pump-and-dump class are higher than the same descriptive statistics in the ordinary observations class, suggesting the existence of pre-pump trading activity. In fact, pump-and-dumps' organizers or other insiders may buy in advance the cryptocurrency targeted in the in each scheme to increase their profits. The third quartile statistics of *Price_rank* and *Vol*_rank features show how cryptocurrencies involved in pump-and-dump schemes tend to be obscure, less liquid coins compared to ordinary trading observations. Most of pump observations rank less than position 288 based on price and trading volumes, while the same proportion of ordinary observations ranks up to position 488 for both price and volume ranks. Lastly, the *Age*, *Exchange*, and *Coin_rank* features retrieved from third-party data providers suggest that the cryptocurrencies targeted in pump-and-dump schemes are generally younger and traded on much less exchange markets. On average, cryptocurrencies listed for trading on Kucoin are traded on overall 22 different exchanges. Instead, the coins targeted in pump-and-dump events exhibit an average number of 4 exchanges. Moreover, 75% of all observations in the sample of pumped cryptocurrencies have an *Age* which is less than 817 days, slightly more than 2 years. The same proportion of observations in the ordinary trading category contains coins whose *Age* reaches 1,637 days, more than 4 years and a half.

4. Methodology

The empirical design of this research work consists firstly of investigating the anatomy of pump-and-dump events and their impacts on the cryptocurrency market by applying a short-term event study around the event announcement dates. This approach allows us to dig into the dynamics of these schemes and answer the first research question of this research, which consists of identifying the key factors that contribute to the success of these schemes. According to the research led by Hamrick et al. (2018), I expect pump profitability to be significantly related to the coin rank based on market capitalization. Moreover, I also hypothesize that the likelihood of success of these frauds depend on the asset's liquidity in the specific exchange, which may be proxied through trading volumes. The empirical methodology of this thesis proceeds to train and test several machine and deep-learning models to predict the cryptocurrencies targeted in pump-and-dump schemes. Feature importance is also analyzed to find the most relevant features on which models perform their predictions and answer the second research question, which consists of identifying useful factors for predicting the occurrences of these schemes. During this stage I rely on data augmentation techniques to handle the class imbalance present in our dataset, since pump-and-dump events represent relatively rare observations in the market. Moreover, the final variables chosen to feed the machine-learning models are identified based on stepwise feature selection methods. I compare the performances of the following machine and deep-learning models: Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and Feedforward Neural Network. Particularly, I expect these models to be able to predict the occurrences of pump-and-dump events, based on the pre-pump patterns and the other features showed in the Data Chapter 3.2. Moreover, I also hypothesize that deep-learning models may outperform traditional machine-learning models in performing this task, since they have done so in the pump-and-dump prediction field according to the work of Nghiem et al. (2021). The last stage of this analysis consists of back-testing an algorithmic trading strategy which attempts to profit from pump-and-dump schemes exploiting the best machine-learning model identified before. This strategy relies on assuming long positions on all cryptocurrencies flagged by the model as potential pump-and-dumps, and then waiting for one of these assets to exhibit a surge in price before closing all open positions. I expect the performance of such a strategy to be directly related to the performance of the machine-learning models, since accurately forecasting pump-and-dump events in advance is the main driver of profitability.

4.1. Short-term Event Study

The short-term event study methodology is a widely recognized and extensively used approach in the field of finance to evaluate the impact of specific events on the financial markets. This methodology has proven to be valuable in assessing the immediate reactions of financial assets to events such as earnings announcements, mergers and acquisitions, regulatory changes, macroeconomic releases, and other significant occurrences. In the context of this study, I am mainly interested in assessing the impact of pump-and-dump events on the cryptocurrency market. This analysis is performed on returns obtained from 1-minute frequency candlestick data retrieved from Kucoin Exchange APIs. The primary objective of an event study is to evaluate the average expected price change of an asset in response to the event. Let's consider the asset return of a specific cryptocurrency, denoted as *Rj,t,* during a given period t if the event occurs, and $\widehat{Rj,t}$ as the return of the same cryptocurrency if the event does not occur. The quantity we are interested in represents the difference between the expected values of these two returns described by the following equation:

$$E\left[\delta_{j,t} \middle| Event\right] = \mathrm{Rj,t} - \mathrm{E}\left[\widehat{\mathrm{Rj,t}} \middle| Event\right] \tag{4.1}$$

Unfortunately, it is not possible to measure these two figures simultaneously for the same asset. In fact, the value of Rj, t is available for cryptocurrencies which have been involved in pump-and-dumps events, but $\widehat{Rj,t}$ is not known for the same observations. However, it is possible to evaluate the effect of pump-and-dump events on the cryptocurrencies involved using a financial model to estimate $\mathrm{E}\left[\widehat{\mathrm{Rj,t}} \middle| Event\right]$, which represents the expected asset return if the pump-and-dump had not occurred. I define as normal return the estimate of $\mathrm{E}\left[\widehat{\mathrm{Rj,t}} \middle| Event\right]$ and apply the Mean Adjusted Model to retrieve this estimate according to the relation:

$$\mathrm{E}\left[\widehat{\mathrm{Rj,t}} \middle| Event\right] = \gamma_j \tag{4.2}$$

Where γ_j denotes the average 1-minute return over the estimation window, which is a subset of our sample including dates associated to returns which are not affected from the event. Specifically, I choose an estimation window which ranges from 14 days to 7 days before the pump-and-dump event. The event's time is defined as the moment in which the targeted cryptocurrency name is released to the public on the Telegram channel "Kucoin Binance

Pumps Trading" from pump-and-dump organizers. I intentionally exclude the week before the event from the estimation window data to avoid contamination of return data from pre-pump buy activity: in fact, according to the data in Chapter 3.2, the pre-pump activity's effect on prices is quite relevant up to 72 hours before the pump. Moreover, I limit our estimation window up to 14 days to exclude the potential effects of previous pump-and-dumps on the same cryptocurrency which could bias the data. The choice of the Mean Adjusted Model for estimating the value of $\mathrm{E}\left[\widehat{\mathrm{Rj,t}} \,\middle|\, Event\right]$ assumes that the average return of the cryptocurrency would have been constant if the pump-and-dump had not happened. I believe this assumption might be useful in the cryptocurrency market, since most of other financial models require the definition of a market index, and such a choice may be particularly difficult in this context. The estimate of the value of $\mathrm{E}\left[\widehat{\mathrm{Rj,t}} \,\middle|\, Event\right]$ takes the name of normal return ($NR_{j,t}$), and this figure is then subtracted from the observed asset return in the event window to evaluate the abnormal return ($AR_{j,t}$), which is an unbiased estimator of $E\left[\delta_{j,t} \,\middle|\, Event\right]$:

$$AR_{j,t} = R_{j,t} - NR_{j,t} \tag{4.3}$$

The event window covers all the dates posterior to the pump-and-dump date on which the event influences asset returns. Specifically, I choose as timeframe for the event window all data from the pump-and-dump event's time to two hours after the event. Pump-and-dump frauds in the cryptocurrency market are known to be very fast events, lasting only a few minutes at maximum, so I believe two hours of data to be enough to assess their impact on the market. Based on the abnormal returns evaluated according to Equation 4.3 I perform inference on cross-sectional data to evaluate what factors contribute the most to define pump-and-dumps' profitability. To do so, I first assume that *abnormal returns are uncorrelated across events*. This relation is formalized below:

$$Cov\left(AR_{j1,t}, AR_{j2,t}\right) = 0. \ \forall \mathrm{t} \wedge \forall \mathrm{j1,j2} \quad \mathrm{s.t.\ j1} \neq \mathrm{j2} \tag{4.4}$$

Based on such assumption, I can initially regress the abnormal returns $AR_{j,t}$ on an intercept (μ_t) for each minute *t* in the event window to assess the impact of pump-and-dump events on the asset prices. I expect the average abnormal returns to be statistically significant only in the few minutes following the event, since the short duration of pump-and-dump manipulations in

the cryptocurrency market. This inference is conducted using t-statistics associated to the intercept μ_t through the linear model below:

$$AR_{j,t} = \mu_t + \varepsilon_{j,t} \tag{4.5}$$

Moreover, I am aware of the relatively low number of events in our dataset, so I rely on the Rank Test proposed by Corrado (1989) to draw conclusions which are robust to the presence of outliers, while still considering the magnitude of abnormal returns. In fact, the Rank Test is a non-parametric method to account for such magnitude, but without the need of the distributional assumptions needed to perform inference based on the t-statistics. The stage after assessing the average abnormal returns consists of finding relevant factors which contribute to explain the impact of pump-and-dump schemes in the market. This is performed using linear regression models, but needs further assumptions to be made:

$$E\left[\frac{1}{N}\sum_{j=1}^{N}\sum_{t=0}^{L}\delta_{j,t}\middle|\ \text{Event}\right] = \alpha + x'\beta \tag{4.6}$$

According to the relation above, I assume that the model is linear in the parameters. However, I also need to assume the following on the variance of the error term:

$$Var\left\{CAR_j - E\left[\frac{1}{N}\sum_{j=1}^{N}\sum_{t=0}^{L}\delta_{j,t}\middle|\ \text{Event}\right]\right\} = \sigma^2 \tag{4.7}$$

The equation above states that the variance of the error term is independent from the regressors and is equal across events. Finally, the linear models can be estimated regressing both the abnormal returns ($AR_{j,t}$) and cumulative abnormal returns (CAR_j) on the factors *Coin_rank, Cum_vol_rank,* and *Std_vol_rank*: these features have been accurately described in Data Chapter 3.1. These linear regressions allow us to evaluate the effect of short-term liquidity and cryptocurrency market capitalization on pump-and-dump profitability. Specifically, I am going to perform this analysis separately for the abnormal returns associated to the pump phase and for those associated to the dump phase. I distinguish between these two categories based on the sign of the average abnormal returns at each time *t* obtained from Equation 4.5. The above-mentioned linear regression models are formalized with the equations below:

$$AR_{j,t=k} = \mu_j + \beta_1\, coin\_rank_j + \beta_2\, cum\_vol\_rank_j + \beta_3 std\_vol\_rank_j + \varepsilon_j \quad (4.8)$$

$$CAR_j = \mu_j + \beta_1\, coin\_rank_j + \beta_2\, cum\_vol\_rank_j + \beta_3 std\_vol\_rank_j + \varepsilon_j \quad (4.9)$$

As already laid down at the beginning of this Methodology Chapter, I expect pump profitability to be significantly related to the coin rank based on market capitalization. Moreover, I also hypothesize that the likelihood of success of these frauds depend on the asset's liquidity in the specific exchange, which may be proxied through the features *Cum_vol_rank* and *Std_vol_rank.* These hypotheses will be assessed based on the t-statistics associated to the β_1, β_2, and β_3 coefficients.

4.2. Machine-Learning Models For Target Coin Prediction

The use of machine learning models to predict pump-and-dump schemes is particularly relevant in the context of the cryptocurrency market. Machine learning algorithms can analyze vast amounts of data quickly and efficiently, allowing for the identification of patterns and trends that may not be apparent through traditional analysis methods. Comparing the effectiveness of different machine learning models is an essential and innovative aspect of this research. By conducting such a comparison, this research provides valuable insights into which models are most effective at predicting these schemes, thereby informing the development of more robust and accurate prediction models. In fact, accurate machine learning models can help prevent fraudulent activities and enhance market integrity by enabling market participants to trade smarter and not incur in financial losses. In this specific context, the prediction task being modelled consists of a binary classification problem. Each time a pump-and-dump event is scheduled, only one out of all cryptocurrencies listed for trading on Kucoin is manipulated, while all the other coins continue to be ordinarily traded. Therefore, machine-learning models should be able to classify each cryptocurrency as ordinary trading observation (0) or pump-and-dump target (1) based on the data provided. This section proceeds by firstly introducing the training and test sets building procedure, also presenting the issue related to the heavy class imbalance present in our data, since pumped coins represent relatively few observations compared to the other class. Subsequently, feature selection is performed among the available variables in the dataset through stepwise regression approach to identify the best subsample of regressors for feeding the machine-learning models. The set of machine- and deep-learning models trained for this prediction task is then presented and accurately described. Furthermore, I report the fine-tuning approach employed based on training set cross-validation and the performance metrics adopted for comparing the effectiveness of the different models in predicting the target coin. The methodology used for evaluating feature importance in such models is also explored.

4.2.1. Training - Test Set Creation and Data Augmentation

To evaluate the performance and generalization capability of the machine learning models in predicting pump-and-dump schemes, it is essential to split the dataset into separate training and test sets. This division allows us to train the models on a subset of the data while evaluating their performance on unseen data. The training set constitutes the majority portion of the data and is used for model training and parameter tuning through stratified 5-folds cross validation. It enables the models to learn patterns and features present in the data, including both normal market behavior and pump-and-dump characteristics. I decide to include within the training set the first 37 pump-and-dump events of our datasets. These have been organized on Kucoin from April 25th 2021 to October 6th 2022, representing slightly more than one year of data. Overall, the size of the training set amounts to 28,194 observations. The test set, on the other hand, serves as an independent sample to assess the trained models' performance. It contains instances that the models have not encountered during training, and allows to assess the models' ability to accurately predict these fraudulent schemes. I include in the test set the remaining part of our dataset made up of 20 pump-and-dump events, ranging from October 15th 2022 to May 7th 2023. The test set contains overall 15,240 observations. This splitting procedure of the dataset across training and test set allocates 65% of total observations to training and 35% to testing. In the context of pump-and-dump schemes in the cryptocurrency market, one of the primary challenges lies in dealing with imbalanced datasets. Pump-and-dump observations are relatively rare compared to non-pump observations, resulting in a heavily imbalanced dataset. Considering the whole dataset, only 57 observations are classified as pump-and-dumps, while the other 43,605 observations refer to cryptocurrencies being traded in ordinary way. Therefore, pump-and-dump schemes' observations only constitute around 0.13% of the overall dataset size. This poses a problem when training machine learning models since they tend to learn and generalize better on the majority class, potentially leading to poor performance in predicting the pumped cryptocurrencies, which represent the minority class. This issue is addressed through a data augmentation technique called Synthetic Minority Over-sampling Technique (SMOTE), introduced by the work of Chawla et al. (2002). SMOTE is a widely used method for generating synthetic samples of the minority class by creating new instances that interpolate between existing minority samples. This technique aims to balance the dataset by oversampling the minority class and improving the model's ability to capture and classify pump and dump observations accurately. This data augmentation algorithm works by selecting a minority-class instance x_i and identifying its k-nearest neighbors. Synthetic

instances are then created by randomly selecting one of the k-neighbors x_{min} and generating new instances x_{smote} along the line segments connecting the selected instance x_i and its neighbor x_{min}. Precisely, this data generation process is described by the following equation:

$$x_{smote} = x_i + u_i \ (x_{min} - x_i) \tag{4.10}$$

$$where\ u_i\ is\ uniformly\ distributed\ as\ U\ (0,1)$$

The synthetic observations are added to the dataset, thereby increasing the representation of the minority class while maintaining the overall characteristics of the original data. By applying the SMOTE data augmentation technique, I aim at alleviating the data imbalance problem and enhancing the performance of machine learning models in predicting pump-and-dump schemes. Specifically, I rely on a combination of over-sampling of the minority class and under-sampling of the majority class, since this approach has showed successful in the research of Chawla et al. (2002). During the training phase of machine-learning models I oversample the minority class by 200 times using SMOTE, reaching a total of 7,400 pump-and-dump observations in the training set. The majority class represented by ordinary trading cryptocurrencies is then randomly undersampled to match the number of pump-and-dump observations. The overall size of the training set after the data augmentation process is of 14,800 observations. It is important to note that data augmentation is not applied on the test set, since the purpose of this data is to simulate real-world scenarios to assess the models' performance. This is valid also for the test data folders created with cross-validation during fine-tuning of models' hyperparameters.

4.2.2. Stepwise Feature Selection

Achieving accurate predictions of pump-and-dump schemes requires the identification of the most relevant features that determine the manipulated cryptocurrency. The feature selection process helps to improve the model's performance, interpretability, and generalization capability. One common approach for feature selection is the Stepwise Linear Regression, which iteratively adds or removes features based on their contribution to the adjusted R-squared. Stepwise Linear Regression is a forward-backward selection process that starts with an initial model and iteratively assesses the inclusion or exclusion of features based on their impact on the adjusted R-squared. The adjusted R-squared takes into account both the goodness of fit and the number of features in the model, providing a more reliable measure of model performance and preventing overfitting. In the context of this research, I apply Forward Stepwise Linear Regression. In forward selection, features are incrementally added to the model one at a time, starting with the feature that yields the highest improvement in the adjusted R-squared. At each step, the feature that leads to the maximum increase in the adjusted R-squared is chosen, and the process continues until no further improvement is observed or a predefined stopping criterion is met. In this case, I do not set a specific stopping rule and let the feature selection algorithm run until all variables under consideration are included in the model. This allows the researcher to determine the adjusted R-squared related to each possible number of features, and therefore assess the best combination of regressors for the machine-learning models presented in the following sections.

4.2.3. Logistic Regression

Logistic regression is a widely used statistical technique for modeling the relationship between a binary dependent variable and one or more independent variables. It is particularly suitable for situations where the dependent variable represents a categorical outcome, such as the classification in pump observation or ordinary trading observation. In Logistic Regression, the relationship between the dependent variable (Y) and the independent variables (X_1, X_2, ..., X_r) is modeled using the logistic function. The logistic function, also known as the sigmoid function, maps the linear combination of the independent variables to a probability value between 0 and 1. The logistic function is defined as:

$$p(X) = \frac{1}{(1 + e^{-z})} \tag{4.11}$$

$$where\ z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_r X_r$$

In the relation above, p(X) is the probability of the dependent variable being in the positive category given the values of the independent variables, and z is the linear combination of the independent variables weighted by their corresponding coefficients. The logistic regression model is estimated using maximum likelihood estimation (MLE). The goal of MLE is to find the set of coefficients (β_0, β_1, β_2, ..., β_r) that maximize the likelihood of observing the given data. The likelihood function (L) represents the joint probability of obtaining the observed data as function of the model parameters. Assuming that the observations are independent and identically distributed, the likelihood function is expressed as:

$$L(\beta_0, \beta_1, \beta_2, \ldots, \beta_r) = \prod_{i=1}^{N} p(X_i)^{(Y_i)} * [1 - p(X_i)]^{(1 - Y_i)} \tag{4.12}$$

The maximum likelihood estimation involves finding the set of coefficients that maximizes the logarithm of the likelihood function, known as the log-likelihood function. The estimation process involves iteratively optimizing the log-likelihood function to find the coefficients that maximize it. Several optimization algorithms, such as Newton-Raphson or Gradient Descent, can be used for this purpose. The log-likelihood function is formalized with the equation below:

$$LL(\beta_0, \beta_1, \beta_2, \ldots, \beta_r) = \sum_{i=1}^{N} Y_i * \log[p(X_i)] + (1 - Y_i) * \log[1 - p(X_i)] \quad (4.13)$$

One of the advantages of logistic regression is the ability to interpret the coefficients in terms of odds ratios. The odds ratio describes the change in the odds of the dependent variable for a one-unit change in the corresponding independent variable, while holding other variables constant. Considering an independent variable X_i with a coefficient β_i. The odds ratio (OR) for X_i can be calculated as the exponential of the coefficient's value:

$$OR = e^{\beta_i} \quad (4.14)$$

Interpreting the odds ratio involves understanding how it affects the odds of the positive outcome. It is important to note that odds ratios provide insights into the direction and magnitude of the relationship between independent variables and the odds of the positive outcome. However, they do not directly indicate the precise impact on the probability of the positive outcome, since the relationship between the dependent variable and the regressors is not linear in the Logistic Regression.

4.2.3. Decision Tree Classification

Decision tree classification is a popular machine learning technique used for solving classification problems. It is a supervised learning algorithm that builds a hierarchical structure of decision nodes and leaf nodes based on the features of the dataset. Decision trees consist of nodes that represent decisions or actions and edges that connect the nodes. The root node represents the starting point of the tree, and each subsequent node represents a decision or a test on a particular feature. The leaf nodes, also known as terminal nodes, contain the final classification outcome. The construction of a classification decision tree involves recursively partitioning the dataset based on the feature values that provide the most significant information gain based on the Gini Index (G). This measure is defined according to the following relation:

$$\mathrm{G} = \sum_{k=1}^{K} p_{m,k}\,(1 - p_{m,k}) \tag{4.15}$$

where $p_{m,k}$ is the proportion of training observations in the mth region that are from the kth class

The Gini Index represents a measure of total variance across the k classes to which each observation may belong. Moreover, it is clear that this statistic tends to assume a small value whenever all $p_{m,k}$ are close to zero or one. This property allows the Gini Index to be considered a measure of node purity: small values indicate that a node mainly includes observations from a certain class. The evaluation of the Gini Index is implemented during the recursive binary splitting of data used to grow the classification tree, allowing the model to create branches that lead to different outcomes. This process continues until a stopping criterion is met, such as reaching a maximum depth or having a minimum number of instances in the leaf nodes. Decision trees are highly interpretable and can be easily explained: the hierarchical structure and the intuitive nature of decision rules make them accessible and understandable to a wide range of individuals. Moreover, by following a series of if-then rules based on feature values, decision trees can mimic human decision-making processes more effectively than other statistical approaches.

4.2.4. Random Forest

Random Forest is a highly effective ensemble learning method that combines the predictive power of multiple decision trees to achieve robust and accurate predictions. It overcomes the limitations of individual decision trees by leveraging the concept of "wisdom of the crowd" to enhance overall performance. At its core, Random Forest constructs a large number of decision trees, each trained on a random subset of features and data samples. This technique creates diverse subsets that capture different aspects of the underlying patterns and relationships. The construction of a Random Forest involves two key steps: random feature selection and aggregation of predictions. During tree construction, only a random features subset of size m is considered, ensuring each tree focuses on different aspects of the data:

$$m = \sqrt{p} \tag{4.16}$$

where p is the total number of available regressors

A very common choice for the number of features present in each random subset is given by the equation above, according to which this figure is approximately equal to the square root of the overall number of available features. By considering only a random subset of features at each split point, Random Forest reduces correlation among the trees and improves robustness against overfitting and noisy data. Once all the decision trees are built, predictions are aggregated through voting (in classification tasks) to obtain the final prediction. This ensemble approach minimizes bias, reduces variance, and enhances prediction accuracy. Random Forest offers several advantages. Firstly, it exhibits high predictive accuracy due to the collective decision-making of multiple trees. The ensemble's ability to capture diverse perspectives and patterns in the data leads to improved overall performance. Secondly, Random Forest is robust against overfitting, thanks to random feature selection and bootstrapping. By reducing the trees' tendency to memorize noise, this model usually generalizes well to unseen data. However, it is important to acknowledge that Random Forest has its limitations. It can be computationally intensive, particularly when dealing with large datasets and a large number of trees. Additionally, the interpretability of the model may be challenging due to its ensemble nature. Careful tuning of hyperparameters is also necessary to optimize the model performance.

4.2.5. Support Vector Machine

Support Vector Machine (SVM) is a versatile and powerful supervised learning algorithm that excels in both classification and regression tasks. It offers a robust methodology for pattern recognition and prediction by identifying optimal decision boundaries or hyperplanes in high-dimensional feature spaces. At its core, SVM aims to find a hyperplane that maximally separates different classes in the input data. In binary classification, this hyperplane acts as a decision boundary, effectively separating the data points of one class from the other. SVM achieves this by defining support vectors, which are data points located closest to the decision boundary. These support vectors play a crucial role in determining the optimal hyperplane. One key advantage of SVM is its ability to handle linearly non-separable data by employing the kernel trick. Precisely, a kernel is a function which quantifies the similarity of two different observations. By transforming the input data into a higher-dimensional feature space, SVM can find a linear decision boundary that effectively separates the classes. A common kernel function which defines the linear support vector classifier is defined as:

$$\mathrm{K}(x_i, x_k) = \sum_{j=1}^{p} x_{i,j} * x_{k,j} \tag{4.17}$$

where x_i *and* x_k *represent two observations* and p is the number of features

Such a kernel function is linear and represents the inner product of two observations in the dataset. This function basically describes the similarity of two observations based on the Pearson standard correlation. The associated support vector classifier takes the following functional form:

$$\mathrm{f(x)} = \beta_0 + \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{p} x_{i,j} * x_{k,j} \tag{4.18}$$

where x_i *and* x_k *represent two observations*, p is the number of features, and n is the total number of training observations and parameters α_i

According to this model definition, all we need for representing the linear classifier f(x) and computing its coefficients are the inner products of the data observations. However, the equation reported above can be generalized to non-linear support vector classifiers changing the definition of the kernel function $\mathrm{K}(x_i, x_k)$. Other commonly used kernel functions include polynomial,

radial basis function (RBF), and sigmoid, each suited for different types of data. For instance, one could replace every instance of $\sum_{j=1}^{p} x_{i,j} * x_{k,j}$ in the linear kernel function with:

$$K(x_i, x_k) = \left(1 + \sum_{j=1}^{p} x_{i,j} * x_{k,j}\right)^d \quad (4.19)$$

where x_i and x_k represent two observations, p is the number of features, and d is the degree of the polynomial.

The relation above is also known as polynomial kernel of degree d, a positive integer. Using such a kernel with values of d greater than one allows for a more flexible decision boundary, fitting the support vector classifier in a higher dimensional space. This model can therefore handle complex decision boundaries and capture non-linear relationships by using non-linear kernel functions. This flexibility allows SVM to model intricate patterns and achieve high accuracy in various domains. Another advantage of SVM is its ability to control the trade-off between model complexity and generalization. Through the use of regularization parameters, SVM can find a balance between fitting the training data well and maintaining good generalization to unseen data. This control over model complexity helps prevent overfitting and improves the model's ability to make accurate predictions on new data. Additionally, SVM is less affected by the curse of dimensionality compared to some other machine learning algorithms. It can handle datasets with a large number of features without sacrificing performance. By focusing on the support vectors, SVM remains computationally efficient even in high-dimensional spaces. However, SVM also has some considerations. The selection of the appropriate kernel function and regularization parameters requires careful tuning, which can be a challenging task. Furthermore, SVM's interpretability is often limited due to its reliance on complex decision boundaries.

4.2.6. Feedforward Neural Networks

Feedforward Neural Networks, also known as Multi-Layer Perceptrons (MLPs), are a fundamental class of artificial neural networks widely used for various machine learning tasks, including classification, regression, and pattern recognition. They are known for their ability to learn complex non-linear relationships and make accurate predictions. At its core, a feedforward neural network consists of an input layer, one or more hidden layers, and an output layer. Each layer comprises interconnected nodes, also known as neurons, which apply weighted transformations to the input data. The weights represent the network's parameters, which are adjusted during the training process to minimize the error between predicted and actual outputs. Feedforward neural networks use forward propagation to compute outputs. The input data flows through the network layer by layer, with each neuron often applying a non-linear activation function to its weighted sum of inputs. Common activation functions include sigmoid, hyperbolic tangent, and rectified linear unit. These non-linear functions enable the network to learn and represent complex relationships between the input and output variables. Determining the optimal structure of neural networks can be a complex task and requires to make a relevant number of arbitrary choices. Different cases and scenarios require different architectural choices. For example, the number of hidden layers, the number of neurons per layer, the type of activation functions, and the presence of regularization techniques may vary depending on the characteristics of the dataset and the desired model complexity. Since there is uncertainty regarding which network architecture may perform better in this specific case, I proceed building 3 different neural networks which mainly differ on the number of hidden layers. In all models' specifications I follow the geometric pyramid rule introduced by Masters (1993), which is a design principle that involves gradually reducing the number of neurons as we move deeper into the network. By employing the geometric pyramid rule, neural networks can effectively leverage the power of depth while keeping the computational requirements manageable. Moreover, I rely on L2 regularization to mitigate overfitting and hopefully improve the generalization performance of the models. Specifically, at the network depth level I consider a first architecture containing 3 hidden layers with respectively 32, 16, and 8 neurons in each layer as described in Figure A.1. in Appendix. I then consider a second architecture equal to the previous one, but containing only the first two hidden layers with 32 and 16 neurons each: this is visually inspected in Figure A.2. In Appendix. The last and third architecture specification only contains one hidden layer with 32 neurons and is reported in Figure A.3. in Appendix. All these architectures implement Rectified Linear Unit (ReLu) as non-linear

activation functions for the hidden layers included. Moreover, all the neural neworks built contain an output layer with only one neuron and sigmoid activation function, since our setting represents a binary classification problem. Each final model specification is accurately described and labeled in Table A.1 in Appendix. The feedforward neural networks built in this section are trained using Root Mean Squared Propagation (RMSprop), which is an extension of the Gradient Descent approach that uses a decaying average of partial gradients in the adaptation of the step size (learning rate) for each parameter. The use of a decaying moving average allows the algorithm to forget early gradients and focus on the most recently observed partial gradients seen during the process. The learning rate is a fundamental parameter of the Gradient Descent approach for the minimization of the loss function during the training phase. When this hyperparameter is too low, the training algorithm may take a long time to converge or get stuck in a local minimum. On the other hand, too high values may let the model overshoot the optimal weights and fail to converge. RMSprop algorithm aims to overcoming the challenges associated to fixed learning rates by dynamically scaling the updates for each parameter based on past behavior. This often provides much more stable convergence and faster training compared to traditional Gradient Descent methods. Lastly, the loss function chosen for minimization during the training process is the binary crossentropy, which measures the dissimilarity between predicted probabilities and the true labels.

4.2.7. Hyperparameters Tuning and Performance Evaluation

In machine learning, hyperparameters play a critical role in the performance of models. Fine-tuning these hyperparameters can significantly impact the model's predictive capability and generalization ability. Across the machine and deep-learning models employed in this research, those which require extensive tuning of their parameters are decision trees, random forest, support vector machines, and neural networks. In the case of decision trees, I consider as parameters the maximum depth, the minimum samples per split, and minimum samples per leaf. By exploring different combinations of these hyperparameters, the decision tree's accuracy can be improved and overfitting avoided, particularly limiting the decision tree depth. The random forest algorithm also can be tuned using the same parameters involved in decision trees, however this process is more computationally expensive. Therefore, I only tune the number of trees included in the forest. Fine-tuning support vector machines, instead, involves optimizing the values of parameters as the regularization parameter (C) and the kernel function. These parameters influence the trade-off between maximizing the margin and minimizing classification errors. Lastly, feedforward neural networks are tuned assessing the best values of batch size and number of training epochs for each of the network architectures described in Table A.1 in Appendix. I apply the fine-tuning process of models' parameters using stratified 5-folds cross validation on the training set. This technique divides the training dataset into five equally sized folds while preserving the class distribution. During each iteration, one different fold acts as the validation set, while the remaining folds are used for training. This ensures that the model is evaluated on different subsets of the data, capturing its ability to generalize across various samples. To assess the predictive performance of our models, I employ the AUC (Area Under the ROC Curve) metric. The ROC curve illustrates the trade-off between true positive rate (sensitivity) and false positive rate (specificity) at various classification thresholds. The AUC represents the area under this curve, providing a single summary of the model's discrimination power. Higher AUC values indicate better overall performance, with 1 being the perfect score, and 0.50 indicating that predictive ability is no better than random guessing. Moreover, I also decide to report the sensitivity and specificity performance metrics computed at the probability threshold of 50%. Sensitivity measures the proportion of actual positive instances that are correctly identified by the model. On the other hand, specificity measures the proportion of actual negative instances that are correctly identified by the model.

4.2.8. Feature Importance

Feature importance analysis is performed on the basis of the best random forest model trained during this research. Precisely, the features employed as regressors in the model can be ranked in importance using the mean decrease in the Gini coefficient. This index represents the impurity or uncertainty of a node in a decision tree. The mean decrease in Gini coefficient assesses how much the Gini coefficient decreases on average across all decision trees in the random forest when a particular feature is used for splitting. By evaluating this statistic, I can identify the features that contribute the most to reducing impurity and improving the model's predictive accuracy. Features with higher values of mean decrease in Gini coefficient are considered more important, as their presence leads to greater reductions in impurity and improves the overall performance of the random forest model. It is important to note that the mean decrease in Gini coefficient is specific to the random forest model and the dataset used. The importance scores should be interpreted within the context of the model's performance and the characteristics of the data. Nonetheless, this approach provides a useful and intuitive measure of feature importance, enabling us to uncover the key variables that drive pump-and-dump predictions and improve the understanding of the dataset.

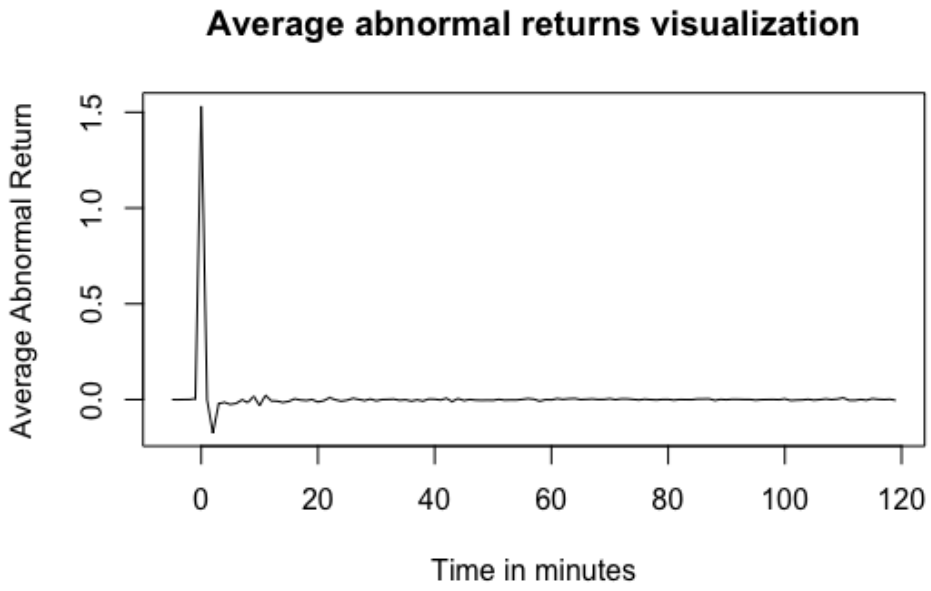
5. Results

This chapter presents the outcomes of the methodology procedures described before. I begin reporting the results associated to the short-term event study of the impact of pump-and-dump schemes on the cryptocurrency exchange Kucoin. After the comprehension of how these phenomena evolve in the market, I proceed describing and comparing the performances of the machine learning models in predicting the cryptocurrency targeted in each fraudulent event. At the end of this section, I perform back-testing of a trading strategy based on the best machine-learning model and report the results in terms of profit.

5.1. Short-term Event Study Results

The event study methodology has allowed us to compute the abnormal returns for the events present in our dataset and dig into the development of these phenomena in the market. The figure below reports the average abnormal returns computed from five minutes before each event to the end of the event window:

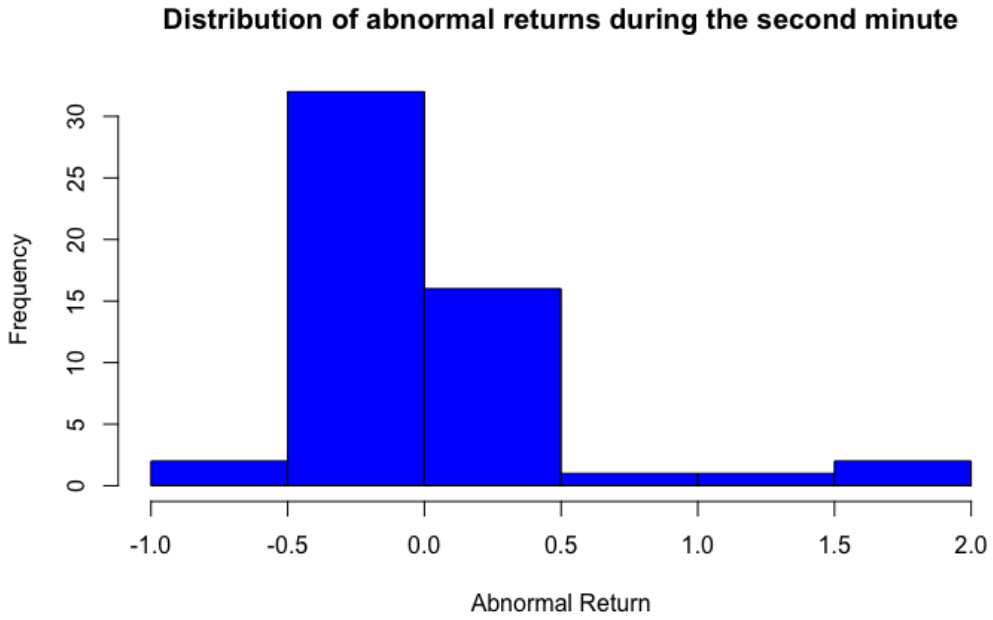
Figure 5.1. Average abnormal returns evaluated from 5 minutes before to 120 minutes after the event.



The figure above confirms that pump-and-dump schemes have an enormous impact on the assets' prices of cryptocurrencies on Kucoin. At time zero, which corresponds to the beginning

of the manipulation, the average abnormal return amounts to 153.23%. The first minute fully captures the pump phase of pump-and-dump events, given the huge positive return also considered the high volatility of cryptocurrency markets. At time one, which is related to the second minute from the beginning of the fraud, the mean abnormal return is surprisingly low and corresponds approximately to -0.94%. The negative sign of this value suggests the beginning the dump phase, however the low magnitude of this average abnormal return may imply mixed behaviours of the pump-and-dump events present in our dataset. This finding is analyzed through the visualization of the distribution of abnormal returns during the second minute, as depicted in the graph below:

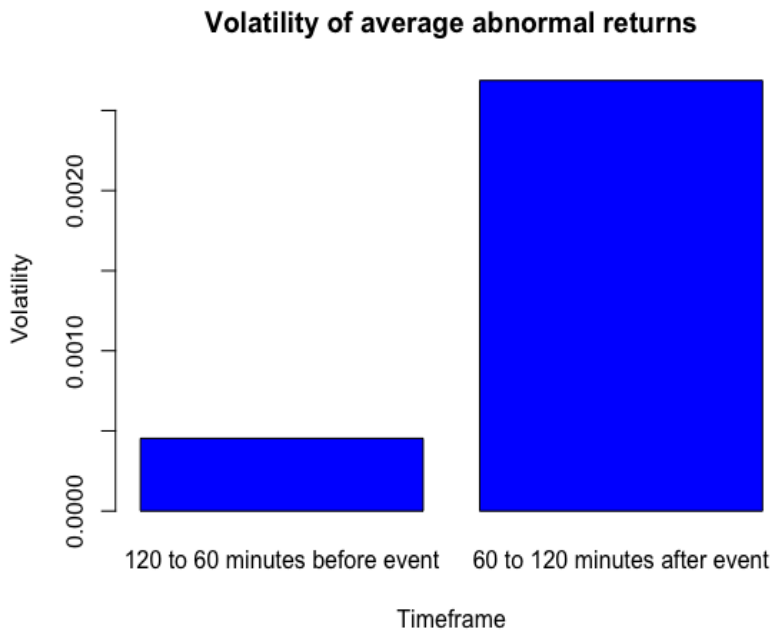
Figure 5.2. Histogram of abnormal returns computed during the second minute from the beginning of the event.



The histogram above confirms my expectations: while the majority of pump-and-dump events have already started the dump phase from the second minute, there still exist a relevant portion of manipulations with huge positive abnormal returns during this time frame. Pump-and-dump behaviours during the second minute are therefore considered uncertain given the mixed findings. Proceeding the analysis based on Figure 5.1, the average abnormal return associated to the third minute from the beginning can be fully related to the dump phase. The magnitude of the mean corresponds to -17.45%, indicating a relevant negative return related to huge sales of the cryptocurrency bought at the beginning of the fraud. The official beginning of the dump phase in correspondence of this time is also suggested from the returns following the third minute. In fact, all average abnormal returns from the fourth to the tenth minute, made

exception for minute nine, have negative signs, ranging from a minimum value of -3.13% to a maximum of -0.14%. This suggests a continuation of the price reduction process started in the second or third minute. From minute eleven onwards, the magnitude of abnormal returns decreases significantly, with the average abnormal returns being included in the range -1% ~ +1%, and with most of the values being close to zero. This pattern for the rest of abnormal returns suggests the end of the manipulation, however it is interesting to note that the presence of these abnormal return movements may indicate that pump-and-dump generate increased trading activity also in the hours following the event. This possibility is strengthened from the process of computing and comparing the standard deviation of average abnormal returns before and after the event:

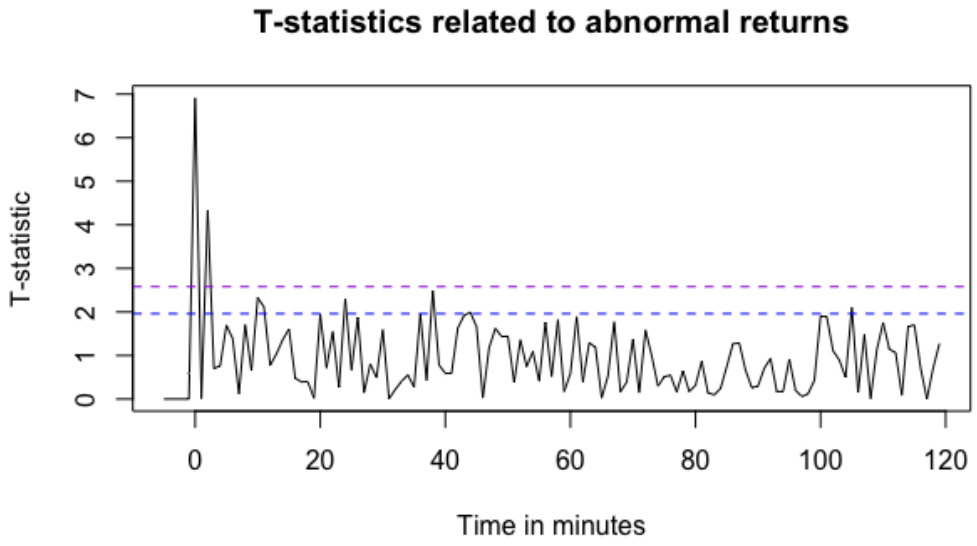
Figure 5.3. Standard deviation of average abnormal returns before and after the pump-and-dump.



Precisely, the volatility of average abnormal returns computed from 60 to 120 minutes before the event time amounts to 0.045%. Instead, the same statistic evaluated from 60 to 120 minutes after the pump-and-dump's beginning equals 0.27%, exactly six times more than the previous value. This clearly indicates that there is more dispersion of the returns in the hours following the event compared to before. The huge abnormal return generated in the first minute may have captured the attention of other investors in the market, leading to increased trading interest and

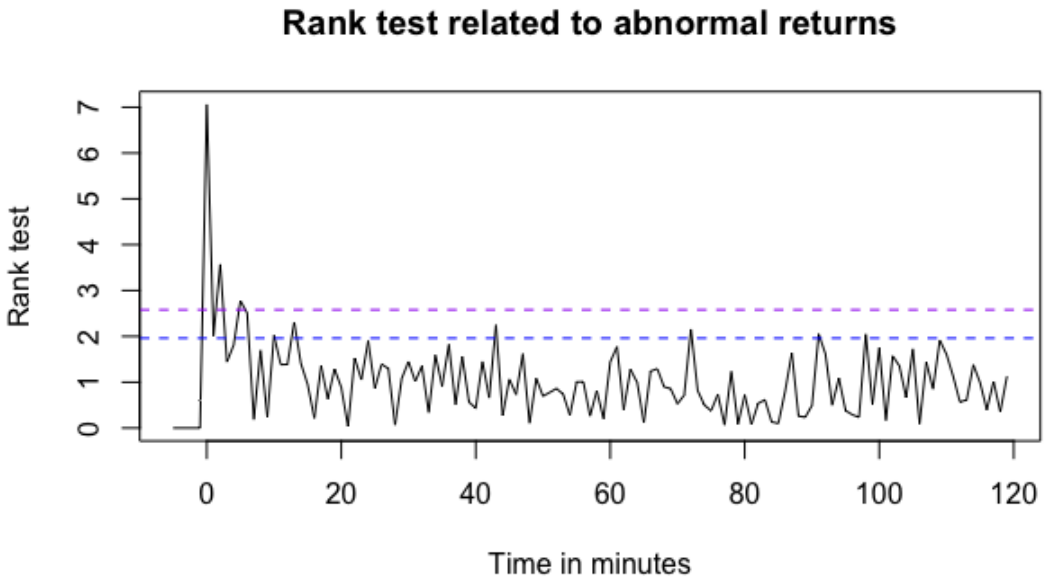
activity on the targeted cryptocurrency. The analysis proceeds exploring the statistical significance of the average abnormal returns presented above using the associated t-statistics:

Figure 5.4. T-statistics of 1-minute average abnormal returns. Blue and purple lines measure 5% and 1% statistical significance levels, respectively.



According to the t-tests showed in Figure 5.4, the average abnormal returns during the first and third minutes after the event are statistically significant at the 1% level. These are the only times that a similar significance level is reached for abnormal returns in the entire event window. Interestingly, there are several average abnormal returns significant at the 5% level distributed along the entire two hours after the pump-and-dump. This finding also encourages the idea that these phenomena influence the market up to hours after the event, even though the core of the manipulation lies within the first three minutes given the highest statistical significance. However, the average abnormal return corresponding to the second minute is not statistically significant since its t-statistic value amounts to 0.016. I strongly believe this result may be inaccurate due to the presence of outliers. This have been checked in Figure 5.2 which shows that the distribution of the average abnormal returns during the second minute is significantly right-skewed. Therefore, I proceed performing the Rank Test proposed by Corrado (1989) to draw more robust conclusions:

Figure 5.5. Rank Test of 1-minute abnormal returns. Blue and purple lines measure 5% and 1% statistical significance levels, respectively.



The results of the Rank Test validate my expectations. I can reject at the 5% level the hypothesis that pump-and-dumps have no effect on the returns during the second minute after the event. Moreover, the figure above interestingly shows a pattern of abnormal returns with significance at the 5% level that decreases more and more as time elapses from the event's beginning. This suggests that the effect of these schemes gradually fades away in the hours after the fraud. After having analyzed the impact and the timing of pump-and-dump events, I can conclude that, on average, the first minute fully includes the pump phase. Instead, I associate the dump phase to the second minute onwards until ten minutes, given the relevant number of negative abnormal returns and their magnitude. I proceed relating the abnormal returns associated to the pump phase (first minute) to the factors *Coin_rank, Cum_vol_rank,* and *Std_vol_rank* according to the following linear regression model:

$$AR_{j,t=0} = \mu_t + \beta_1\, coin\_rank_j + \beta_2\, cum\_vol\_rank_j + \beta_3 std\_vol\_rank_j + \varepsilon_j \qquad (5.1)$$

Moreover, I also fit the model on *Coin_rank* individually, and on a combination of *Coin_rank* and *Cum_vol_rank.* The results contain interesting findings, and are reported in the table below:

Table 5.1. Linear regression of abnormal returns during the first minute after the event.

Regressor	(a)	(b)	(c)
Intercept	-.7105133 (-0.78)	.0591628 (0.06)	-.3418957 (-0.32)
Coin_rank	.001843*** (2.66)	.0017619** (2.55)	.0018891*** (2.74)
Cum_vol_rank	-	-.0254854 (-1.54)	-.0407429* (-1.95)
Std_vol_rank	-	-	.0251094 (1.20)

T-stats reported in round brackets. Statistical significance levels: (***) 1% (**) 5% (*) 10%.

The parameters of the model described in Equation 6.1 are reported in column (c) of the table above. According to this model specification, *Coin_rank* is the most relevant feature in determining the abnormal returns related to the pump, with a significance level of 1%. It affects returns positively, as expected, indicating that cryptocurrencies with lower market capitalization are associated to higher returns during the pump phase. Moreover, *Cum_vol_rank* feature also is statistically significant, but only at the 10% level. The sign of the associated coefficient is negative, confirming that lower trading volumes in the days before the pump-and-dump determine greater abnormal returns during the pump. This suggests that less liquid cryptocurrencies generate higher returns in these schemes. Lastly, the feature *Std_vol_rank* is not statistically significant, so we cannot reject the hypothesis of no effect on pump abnormal returns. Overall, these findings agree with the work of Hamrick et al. (2018), who also identified market capitalization- and volume-based rank features to be strongly related to pump-and-dump profitability. I proceed to assess the effects of these factors on the abnormal returns related to the dump phase. Firstly, the cumulative abnormal returns are computed from the second minute to the tenth after the event time. These values are then regressed on the same features of before according to the model:

$$CAR_j = \mu_t + \beta_1\, coin\_rank_j + \beta_2\, cum\_vol\_rank_j + \beta_3 std\_vol\_rank_j + \varepsilon_j \qquad (5.2)$$

The estimated coefficients and associated t-statistic values are reported in the table below:

Table 5.2. Linear regression of cumulative abnormal returns from the second minute after the event to the tenth.

Regressor	(a)	(b)	(c)
Intercept	.2032601 (0.73)	.2111513 (0.66)	.2663744 (0.78)
Coin_rank	-.0003853* (-1.83)	-.0003861* (-1.81)	-.0004036* (-1.85)
Cum_vol_rank	-	-.0002613 (-0.05)	.0018396 (0.28)
Std_vol_rank	-	-	-.0034574 (-0.52)

T-stats reported in round brackets. Statistical significance levels: (***) 1% (**) 5% (*) 10%.

The feature *Coin_rank* is the only statistically significant variable in explaining cumulative abnormal returns during the dump phase. It has a negative effect on the dependent variable, but only significant at the 10% level. This finding suggests that small-cap cryptocurrencies exhibit a higher magnitude of negative abnormal returns during the dump phase, with stronger price drops. The variable *Cum_vol_rank* does not have statistical significance, but the related coefficient in model (c) has positive sign as expected. Higher trading volumes before the pump seems associated to a lower price reduction when the price starts to dump, potentially because of higher liquidity. The last feature *Std_vol_rank* also is not statistically significant, but suggests that higher dispersion of trading volumes before the event increases the magnitude of negative returns during the dump phase.

5.2. Machine-learning Models' Evaluation

This section focuses on the task of predicting the targeted cryptocurrencies in pump-and-dump schemes comparing the performances of different machine-learning models. Firstly, I apply forward stepwise linear regression to perform feature selection from the initial set of nine variables presented in Chapter Data 3.2. The best linear model is determined in terms of adjusted r-squared, and selects a total of eight features as reported in the table below:

Table 5.3. Outcome of forward stepwise linear regression for feature selection.

Regressor	Coefficient
Intercept	-0.00736530842*** (-3.835)
Last_6_RSI	0.00016635490*** (6.06)
Middle_RSI	0.00004722433* (1.947)
Vol_ratio1	0.00022432953** (2.475)
Vol_ratio2	0.00007185900 (1.494)
Price_rank	-0.00000328993*** (-2.874)
Vol_rank	-0.00000327668** (-2.54)
Coin_rank	0.00000006998** (2.23)
Age	-0.00000096902*** (-3.418)

T-stats reported in round brackets. Statistical significance levels: (***) 1% (**) 5% (*) 10%.

The outcome of the feature selection process has discarded the feature *Exchanges* from the final set of variables chosen to feed the machine-learning models. This feature selection method offers high interpretability of the relation between the features and the dependent variable, based on the coefficients reported in the table above. We can notice that *Last_6_RSI* and *Price_rank* have the strongest statistical significance, and the sign of their coefficients show that high price momentum or low price levels can increase the likelihood of being the

cryptocurrency targeted in pump-and-dump schemes. The feature *Vol_ratio1* also shows that there are useful patterns in the trading volumes before the scheduled event for predicting the manipulated coin. Moreover, the variables *Age* and *Coin_rank* suggest that young cryptocurrencies with low market capitalization are favourable targets in these frauds. After the identification of the final set of variables, I proceed training the machine- and deep- learning models for this prediction task. The assessment of their performance is performed on the test set using the Area Under the ROC Curve (AUC), which is independent from the probability threshold chosen to convert predictions into the binary categories "*pump*" (1) and "*ordinary trading observation*" (0). I also report the sensitivity and specificity of the models evaluated at the probability threshold 50%. The results are reported in the following table:

Table 5.4. Machine- and deep- learning models' performance comparison.

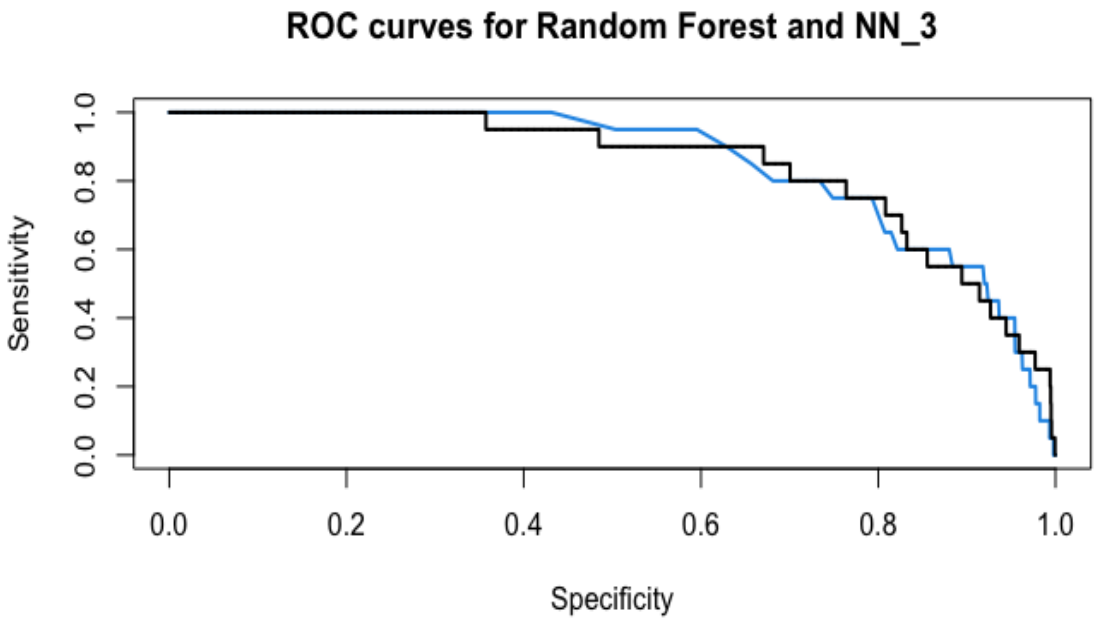
	Training Set			***Test Set***		
Feature	**AUC**	**Sensitivity**	**Specificity**	**AUC**	**Sensitivity**	**Specificity**
Logit	0.8898	0.8412	0.7761	0.8371	0.7000	0.7941
Decision Tree	0.9243	0.9733	0.7854	0.7939	0.7000	0.7537
Random Forest	0.9621	0.9733	0.9045	0.8653	0.9000	0.7822
Support Vector Machine	0.8996	0.8475	0.7812	0.8336	0.65	0.7983
Neural Network (NN_1)	0.9792	0.9769	0.9450	0.8442	0.7000	0.7965
Neural Network (NN_2)	0.9805	0.9808	0.9182	0.8253	0.8000	0.7062
Neural Network (NN_3)	0.9490	0.8856	0.8655	0.8591	0.9000	0.6845

Sensitivity and Specificity metrics have been computed at the 50% probability threshold.

According to the results described above, the best machine-learning model for predicting the cryptocurrency targeted in pump-and-dump schemes is the Random Forest. This finding agrees with the research work of Xu et al. (2019), as the authors also identify this algorithm as the

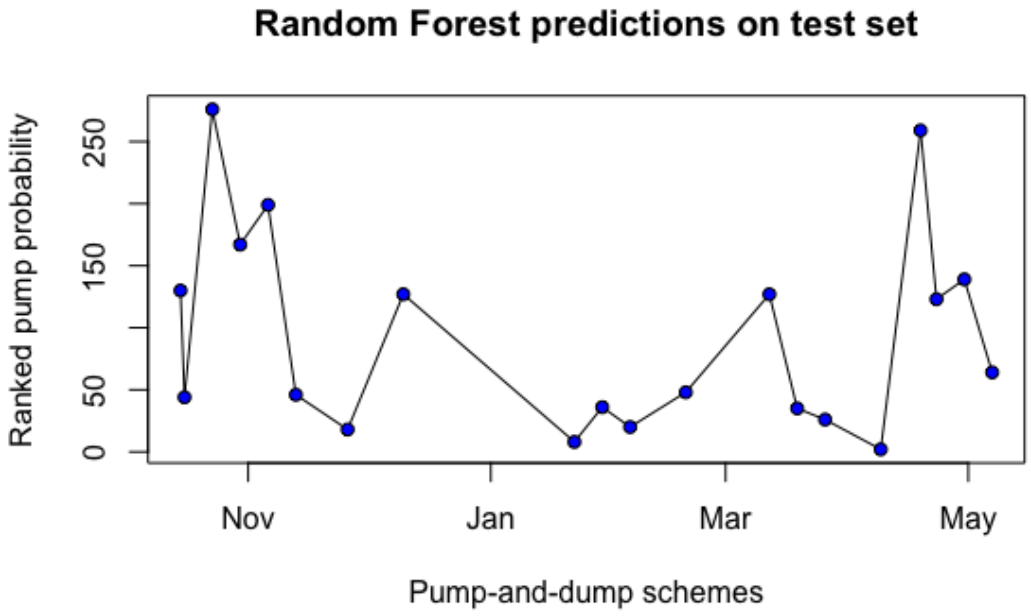
most performant in this prediction task. In this analysis, the Random Forest model achieves the highest AUC score on the test set with a value of 0.8653. Considering a probability threshold of the 50%, the model can correctly predict the 90% of cryptocurrencies involved in the test set's pump-and-dumps, showing however a relevant proportion of false positives as indicated from the specificity value of 78.22%. Across the off-the-shelf machine-learning models, the baseline Logit achieves the highest AUC of 0.8371 after the Random Forest model, confirming the relevant explanatory power of such simple models. Interestingly, none of the deep-learning methods employed in this analysis was able to outperform the Random Forest model. Across the feedforward neural networks, the highest AUC metric on the test set of 0.8591 is associated to the architecture *NN_3*, which is the model definition with the least number of hidden layers, precisely only one. This may be explained from the deeper networks overfitting the training data despite the L2 regularization employed. More hidden layers usually enhance the model's capability to learn intricate and complex patterns, however in this case they may be capturing noise or specific patterns that are not generalizable to unseen data. This idea is also suggested by the very high AUC values that the deeper neural networks reach on the training set, the highest compared to all other models. Visualizing the ROC curve allows us to study how these models behave based on the probability thresholds chosen. I proceed comparing the test set ROC curve of the best deep-learning model *NN_3* with the one of the Random Forest, which is the most performant model compared to all the others.

Figure 5.6. ROC curves of Random Forest and Neural Network NN_3. Blue line is Random Forest. Black line is NN_3.



The figure above shows that the Random Forest model and best feedforward neural network *NN_3* behave in a fairly similar way across most possible combinations of sensitivity and specificity. For very high levels of false positives, given by values of the specificity metric up to 60%, we can notice that the Random Forest outperforms the neural network always achieving sensitivity values equal or higher. For specificity values higher than 60% the behavior of these two models is mixed, with the neural network slightly achieving higher sensitivity values than the Random Forest in most cases for each given specificity. Particularly, the neural network *NN_3* seems to preserve higher sensitivity values for very low false positive rates given by specificity ranging from 95% to 99%. I now apply the most performant model, namely the Random Forest, to make predictions on the test set and graphically show the estimated likelihood of pump-and-dump for the targeted cryptocurrencies. Precisely, I estimate the probability of being targeted for all cryptocurrencies available on Kucoin during each pump event, and subsequently rank these probabilities from 1 to 762, with lowest rank indicating higher probability of being targeted. I then report these rank values for the targeted cryptocurrency in all pump-and-dump events present in the test set:

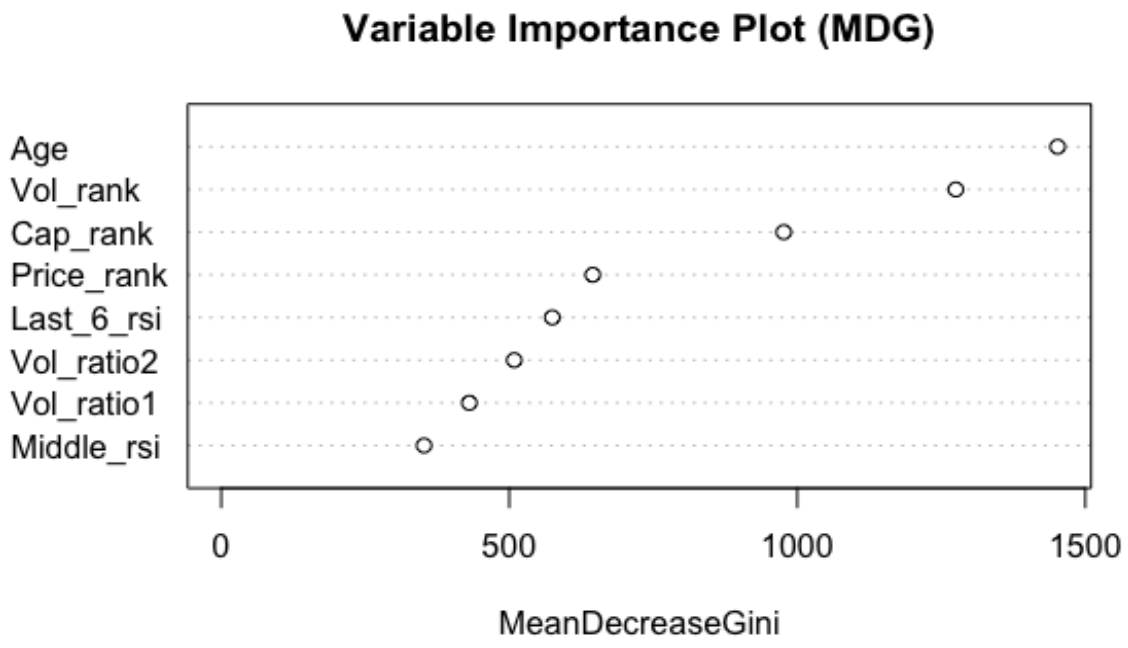
Figure 5.7. Random Forest's ranked estimated probability of pump for targeted cryptocurrencies in test set.



The graph above visually shows the interesting performance of the Random Forest model. Across the pump-and-dump events involved in the test set, more than half of all the targeted cryptocurrencies were ranked from the model below the 50th position out of 762 available coins.

The whole range of ranked predictions goes from a minimum of 1st position to a maximum of 276th position, therefore even in the worst case the model was able to filter almost two thirds of all cryptocurrencies not involved in the pump-and-dump. Very interestingly we can also note from the graph that there exist some sort of pattern in the predictions. Pump-and-dump events organized on Kucoin at the beginning, and at the end of the test set's timeframe, seem more difficult to be predicted accurately. The schemes present in the middle of this time-period, instead, are predicted from the model with better performance on average. This suggests that the organizers of pump-and-dump frauds may follow different coin selection procedures over time, potentially based on the current market condition. The features on which I have trained the Random Forest may be able to capture only some of these selection procedures, leading to poor performance over some periods of the year. This analysis proceeds focusing on the variables used to feed the machine-learning models. Particularly, I compute a measure of feature importance related to the Random Forest model based on Mean Decrease Gini index (MDG):

Figure 5.8. Feature importance computed based on Mean Decrease Gini (MDG) from Random Forest.



The picture above clearly identifies the feature *Age* as the most important predictor of cryptocurrency pump-and-dump schemes. Younger coins are apparently more likely to be targeted in these manipulations. The second and third most important predictors are,

respectively, *Vol_rank* and *Cap_rank*, describing the magnitude of trading volumes associated to the cryptocurrency and its market capitalization. The last coin-specific feature by importance is *Price_rank* which represents the price level of each cryptocurrency compared to the others. Cryptocurrencies traded at higher prices on the exchange seem less likely to be targeted. All these variables score the highest feature importance and are strictly related to the characteristics of each cryptocurrency. The features which have been created to predict pre-pump patterns, instead, are associated to the lowest importance scores. However, they still have a significant impact on the model's predictions, with the variable *Last_6_rsi* being the most important out of this category. This indicates that price momentum in the hours before the manipulation is a relevant predictor of pump-and-dump schemes.

5.3. Algorithmic Trading Strategy Back-testing

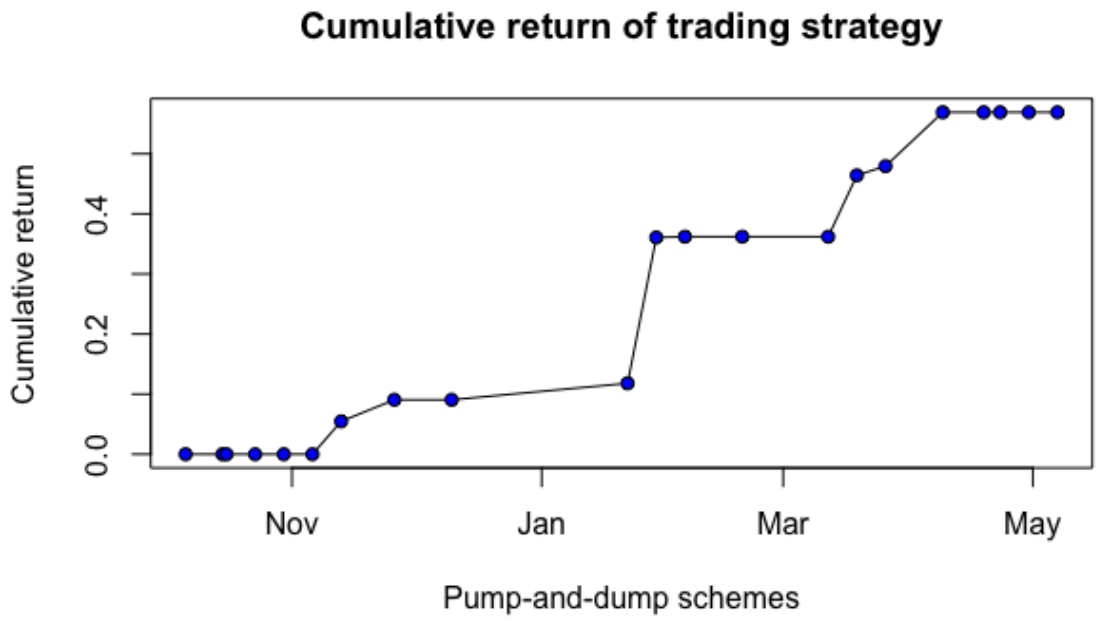
The development of accurate machine-learning models for pump-and-dump schemes' predictions can be leveraged to trade smarter and attempt to make a profit when these events happen. The impact of these schemes on the market has been previously assessed to be enormous, generating huge positive returns during the first minute from the event time. Therefore, a long position on the targeted cryptocurrency assumed before the pump-and-dump scheme would allow the investor to sell the asset at a higher price. The target coin is unknown previous to the event time, but the machine-learning models developed in the sections above can be exploited to forecast such cryptocurrency. Precisely, I rely on the Random Forest model to make this prediction, since this model has showed the best performance on the test set in terms of AUC score. The output of this machine-learning model, for a given cryptocurrency, is the likelihood of being targeted in a pump-and-dump scheme. I therefore apply the Random Forest to evaluate this probability for all cryptocurrencies available on Kucoin each pump-and-dump date in the test set. The cryptocurrencies are then ranked in descending order based on the predicted probability of being targeted in the manipulation, with lower ranks indicating higher likelihood of being targeted.

The trading strategy that I explore consists of assuming equally weighted long positions in the first 40 cryptocurrencies ranked by the model as possible upcoming pumps, and then waiting for the pumped cryptocurrency to be announced. As we know which cryptocurrency has been targeted, all non-relevant long positions opened previously should be closed. The targeted asset should be the only remaining position and is then sold on the market after exactly one minute from the beginning of the manipulation.

I rely on the idea that accurate model predictions should include the actual targeted cryptocurrencies within the first rank positions. At the same time, I assume that the cryptocurrencies not involved in the manipulation remain stable in price, allowing us to close their positions neither in profit nor loss. For this reason, the long positions should be ideally opened only a few seconds before the pump-and-dump time, since it is reasonable not to expect relevant price changes within such a short time frame. In practice, this is implementable through the Exchange APIs that allow us to algorithmically interact with the market and execute orders at very high frequency.

I perform back-testing of the presented trading strategy based on the pump-and-dump events in the test set used for machine-learning models' evaluation. The figure below reports the cumulative returns generated by this trading strategy:

Figure 5.9. Cumulative returns generated by trading strategy based on Random Forest.



The Random Forest model combined with the trading rules defined above would have allowed investors to generate cumulative returns of 56.90% from October 15th , 2022 to May 7th , 2023. This is a relevant result, since the considered time period only requires less than seven months to generate this result. Across all correctly predicted pump-and-dump events, the highest return was generated on January 29th, 2022, with the pump of cryptocurrency WAL. This manipulation inflated the asset price of WAL by 868% at the end of the first minute from the event time, generating a significant return of 21.79% in this trading strategy. I have chosen the first minute as time to sell the inflated cryptocurrency because the minute-level is the highest frequency at which Kucoin provides market data. However, the high of the first 1-minute candlestick after the event suggests for most manipulations that the price peak is reached before the end of the first minute. Back-testing the same trading strategy of above, but ideally selling the inflated asset at the highest price reached during the first minute, would have generated cumulative returns up to 115.92% within the same time period. This represents a huge number, but of course it is not possible to exactly forecast in advance when the price high comes, so this estimate should be interpreted as an upper bound for potential profits.

6. Conclusion

In this thesis, I conducted a short-term event study to analyse the impact of pump-and-dump schemes on cryptocurrency prices on Kucoin. Precisely, I computed the average abnormal returns for the events in our dataset and examined their development in the market. The findings reveal that pump-and-dump schemes have a significant impact on cryptocurrency prices. At the beginning of the manipulation, the average abnormal return amounted to 153.23%, indicating a substantial increase in prices. The first minute captured the pump phase of the events, with a large positive return and high volatility. However, in the second minute, the average abnormal return was surprisingly low in magnitude (-0.94%), suggesting the beginning of the price dump. Further analysis of the second minute abnormal returns showed uncertainty about pump-and-dumps' behaviours in terms of pump or dump phases. From the third minute onwards, the abnormal returns were consistently negative, indicating a continuation of the price reduction process. The magnitude of abnormal returns decreased significantly after the tenth minute, suggesting the end of the manipulation. I also examined the standard deviation of average abnormal returns before and after the pump-and-dump events. The results showed a significant increase in the dispersion of returns persisting for hours after the event, indicating heightened trading activity and interest in the targeted cryptocurrencies.

The statistical significance analysis using t-tests demonstrated that the first and third-minute abnormal returns were statistically significant at the 1% level. Several abnormal returns during the two hours after the pump-and-dump events were significant at the 5% level, suggesting the influence of these schemes on the market for an extended period. However, the second-minute abnormal return was not statistically significant, possibly due to the presence of outliers. To draw more robust conclusions, I performed the Rank Test proposed by Corrado (1989), which validated the effect of pump-and-dumps on returns during the second minute. The significance of abnormal returns gradually decreased as time elapsed from the event's beginning, indicating a fading effect of these schemes over time.

I then regressed the first-minute abnormal returns on a set of factors describing the market capitalization and trading volumes of the cryptocurrencies before the pump. The regression analysis revealed that rank based on market cap is the most relevant feature in determining abnormal returns during the pump phase, with a significance level of 1%. This suggested that cryptocurrencies with a lower market capitalization are associated to higher returns in the pump

phase. Moreover, rank based on trading volumes was also statistically significant, albeit at the 10% level, indicating that less liquid cryptocurrencies generate higher returns in these schemes. The same regression model was applied on cumulative abnormal returns from two to ten minutes after the event to capture the dump phase and relate its returns to the factors introduced above. This regression analysis indicated that only rank based on market capitalization was statistically significant. Small-cap coins exhibit a higher magnitude of negative abnormal returns during the dump phase. These findings align with previous research by Hamrick et al. (2018), which also identified market capitalization and volume-based rank features as strongly related to pump-and-dump profitability.

In addition to event study analysis, I employed machine-learning models to predict the targeted cryptocurrencies in pump-and-dump schemes. Forward stepwise linear regression was applied for feature selection, resulting in a set of eight features for feeding the models. The selected features showed statistical significance using a linear regression model and provided insights into the relation between coin characteristics and the likelihood of being targeted. The performance evaluation of the machine-learning models was conducted using the Area Under the ROC Curve (AUC). The results showed that the Random Forest model achieved the highest AUC score of 0.8653 on the test set, indicating its effectiveness in predicting the targeted cryptocurrencies. The sensitivity and specificity of the models varied, with the Random Forest model showing a good balance between the two. The deep-learning methodologies employed were not able to outperform off-the-shelf machine-learning models in this prediction task. These results agreed with the research work of Xu et al. (2019) who also identified the Random Forest as best predictive model for this task.

Among the selected features, the most influential predictor of pump-and-dump schemes was the age of the cryptocurrency. Younger coins are found to be more susceptible to being targeted in these fraudulent schemes. Additionally, the trading volume ranking and market capitalization ranking of the cryptocurrency were identified as important factors. Cryptocurrencies with higher trading volumes and market capitalization are less likely to be targeted in pump-and-dump activities. Another significant predictor was the price ranking, which indicates that cryptocurrencies traded at higher price levels are less likely to be manipulated. Furthermore, the model highlights the significance of price momentum in the hours preceding the manipulation, with the RSI technical indicator standing out as a relevant predictor.

I have then back-tested an algorithmic trading strategy based on the Random Forest model to profit from pump-and-dump schemes. By leveraging machine-learning, I devised a trading approach that involved opening long positions in the top 40 cryptocurrencies ranked by the model as potential pump-and-dump events, closing non-relevant positions upon the announcement of the targeted asset. The pumped cryptocurrency position is then closed selling the asset exactly one minute after the event. Back-testing this strategy using historical pump-and-dump events resulted in impressive cumulative returns of 56.90% within a period of less than seven months. This confirms that machine-learning can be leveraged to trade smarter and generate profits out of these events.

Overall, this thesis contributes to the academic literature by providing insights into the short-term effects of pump-and-dump schemes on cryptocurrency prices. Moreover, this study contributes to the field of machine learning by evaluating and comparing the performance of different models in predicting the cryptocurrencies targeted in pump-and-dump schemes. However, the results obtained in this research should be interpreted in light of the data on which this analysis was based. This work focused on pump-and-dump schemes on Kucoin and may not be generalizable to other cryptocurrency exchanges or markets. The presence of outliers and the small size of the pump-and-dump events' dataset could have also affected some of the results. Moreover, the trading strategy back-testing's figures rely on the assumption of price stability for the cryptocurrencies not involved in the manipulation. Also, transaction costs have not been accounted for. Future research can explore other markets and exchanges, and further investigate these phenomena collecting more data for both inference and prediction tasks.

References

1. Kamps, J., & Kleinberg, B. (2018). To the moon: Defining and detecting cryptocurrency pump-and-dumps. Crime Science Journal. https://doi.org/10.1186/s40163-018-0093-5

2. Hamrick, J. T., Rouhi, F., Mukherjee, A., Feder, A., Gandal, N., Moore, T., & Vasek, M. (2018). The economics of cryptocurrency pump-and-dump schemes. CEPR Discussion Paper No. DP13404. https://doi.org/10.2139/ssrn.3303365

3. Dhawan, A., & Putnins, T. J. (2021). A new wolf in town? Pump-and-dump manipulation in cryptocurrency markets. Review of Finance. https://doi.org/10.1093/rof/rfac051

4. Xu, J., & Livshits, B. (2019). The anatomy of a cryptocurrency pump-and-dump scheme. Proceedings of the 28th USENIX Security Symposium. https://doi.org/10.48550/arXiv.1811.10109

5. La Morgia, M., Mei, A., Sassi, F., & Stefa, J. (2020). Pump and dumps in the Bitcoin era: real-time detection of cryptocurrency market manipulations. 29th International Conference on Computer Communications and Networks (ICCCN). https://doi.org/10.1109/ICCCN49398.2020.9209660

6. Chadalapaka, V., Chang, K., Mahajan, G., & Vasil, A. (2022). Crypto pump-and-dump detection via deep learning techniques. ArXiv preprint server. https://doi.org/10.48550/arXiv.2205.04646

7. C. J. Corrado (1989). A nonparametric test for abnormal security-price performance in event studies. Journal of Financial Economics, 23, 385 – 395.

8. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.

9. Masters, T. (1993). Practical neural network recipes in C++. San Diego, CA: Academic Press Professional, Inc. https://doi.org/10.1016/B978-0-08-051433-8. 50001-X

10. Nghiem, H., Muric G., Morstatter F., Ferrara E. (2021). Detecting cryptocurrency pump-and-dump frauds using market and social signals. Expert Systems with Applications,182. https://doi.org/10.1016/j.eswa.2021.115284

11. Aggarwal, R. K., & Wu, G. (2006). Stock Market Manipulations. The Journal of Business, 79(4), 1915–1953. https://doi.org/10.1086/503652

12. Bouraoui, T. (2015). Does 'Pump and Dump' Affect Stock Markets? International Journal of Trade, Economics and Finance, 6.

13. F. Allen and D. Gale (1992). Stock-price manipulation. The Review of Financial Studies, 5, 503–529.

14. Janet Austin (2018). How Do I Sell My Crowdfunded Shares? Developing Exchanges and Markets to Trade Securities Issued by Start-Ups and Small Companies. Harvard Business Law Review, 8, 21–35.

Webliography

1. https://github.com/SystemsLab-Sapienza/pump-and-dump-dataset

2. https://coinmarketcap.com/rankings/exchanges/

3. https://www.coingecko.com/en/exchanges/kucoin

4. https://api.coinpaprika.com/v1/coins/

5. https://www.livecoinwatch.com/

6. https://www.kucoin.com/

7. https://www.binance.com/

Appendix

Figure A.1. Feedforward Neural Network with 3 hidden layers each with 64, 32, and 16 neurons. (NN_1)

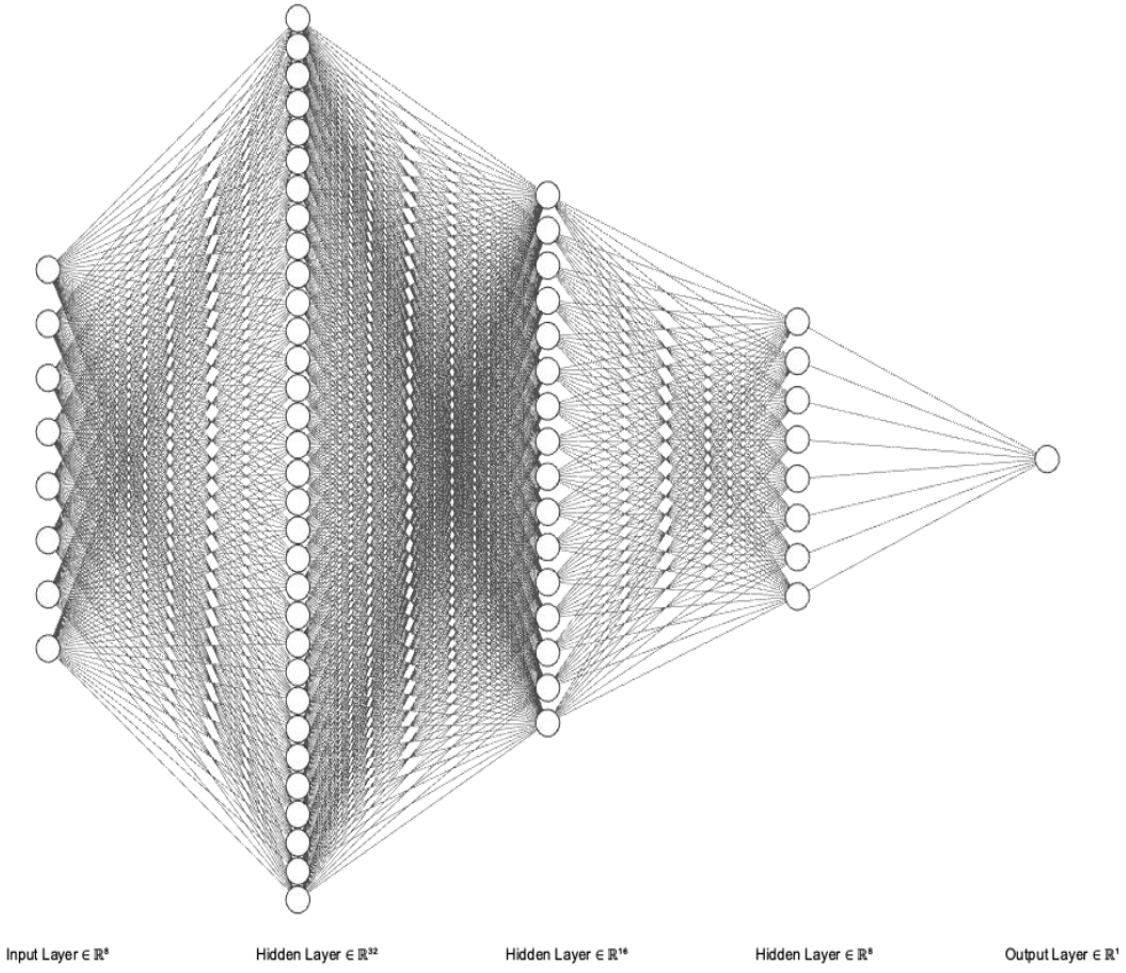


Figure A.2. Feedforward Neural Network with 2 hidden layers each with 32, and 16 neurons. (NN_2)

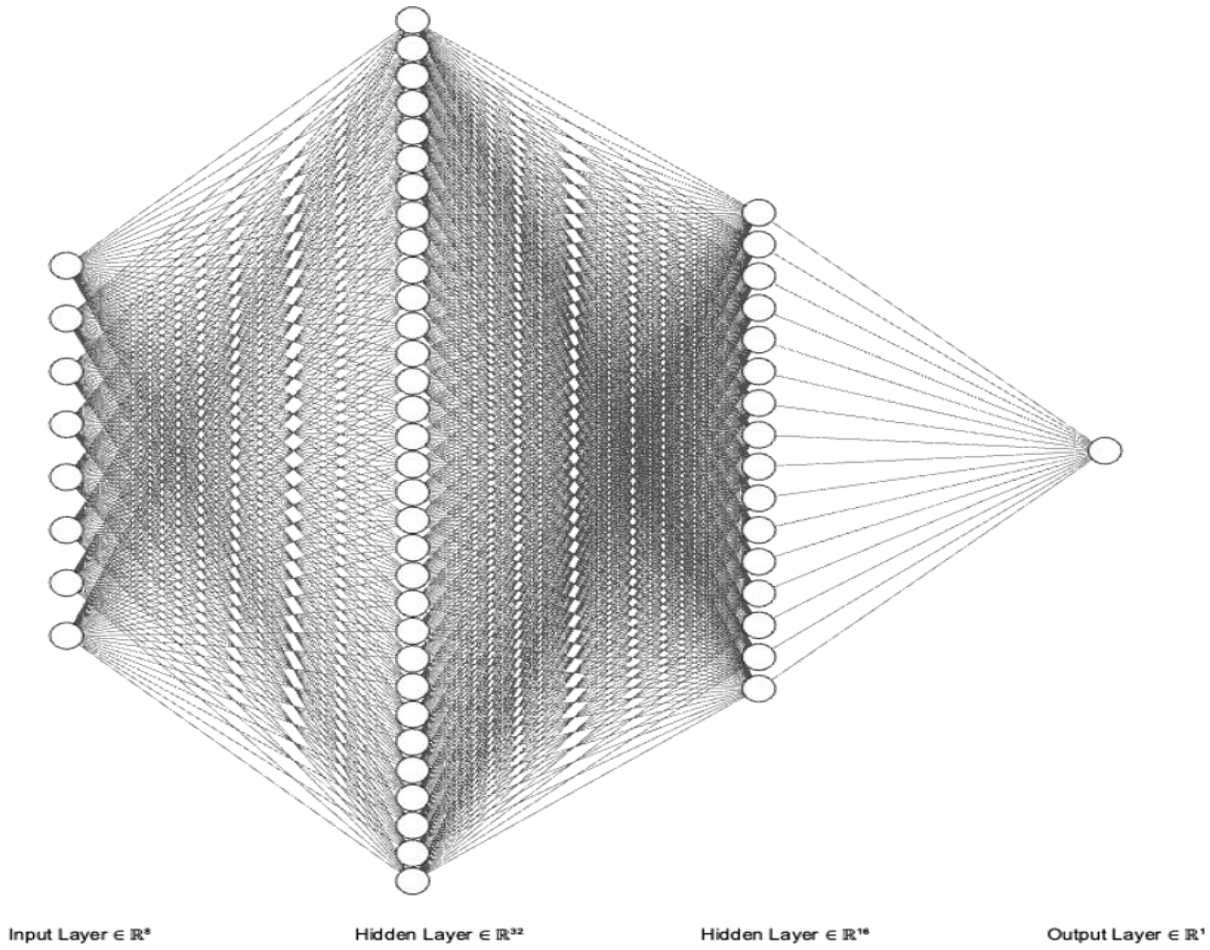


Figure A.3. Feedforward Neural Network with 1 hidden layer with 32 neurons. (NN_3)

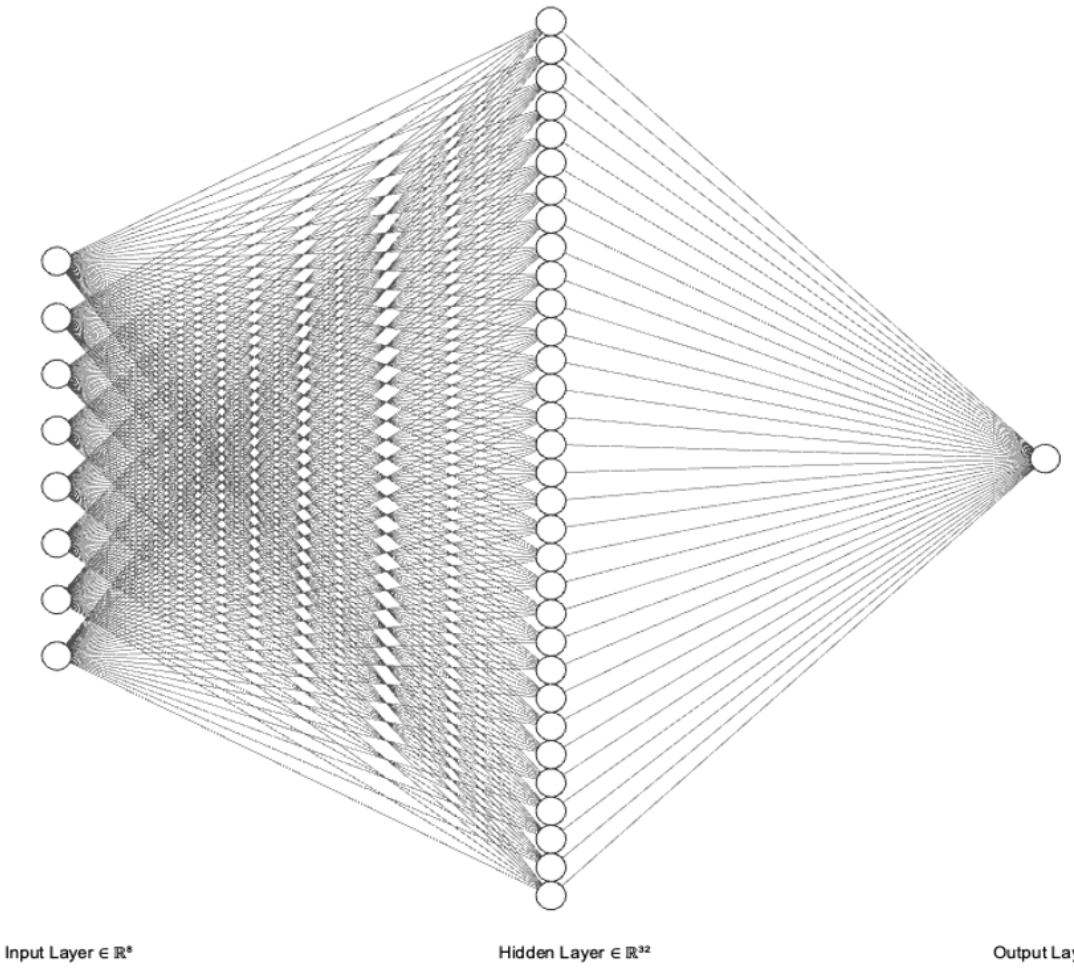


Table A.1. Feedforward Neural Network architectures description.

Architecture	***Hidden Layer 1***	***Hidden Layer 2***	***Hidden Layer 3***	***Activ. Function (H)***	***Output Layer***	***Activ. Function (O)***
NN_1	32	16	8	ReLu	1	Sigmoid
NN_2	32	16	-	ReLu	1	Sigmoid
NN_3	32	-	-	ReLu	1	Sigmoid

The column Activ. Function (H) refers to the activation function employed in all the hidden layers.

The column Activ. Function (O) refers to the one employed in the output layer.

The columns Hidden Layer and Output Layer report the number of included neurons.

Symbol – indicates that the hidden layer is not present in this architecture.