POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Study and development of machine learning-based cryptocurrency trading systems

Supervisors

Prof. Luca CAGLIERO

PhD. Jacopo FIOR

PhD Giuseppe ATTANASIO

Candidate

Giuseppe D'AGOSTINO

06 2021

"Ye were not form'd to live the life of brutes,
'But virtue to pursue and knowledge high."
Dante Alighieri, Inferno, CANTO XXVI,
Translated by The Rev. Henry Francis Cary, A.M.

Table of Contents

List of Tables

List of Figures

Abstract

Since the release of Bitcoin, cryptocurrencies have gained more and more attention, becoming an important financial reality. Market capitalization exploded when the Bitcoin/USD pair reached its all-time high in December 2017, attracting investors from retail, professional and institutional markets in a novel gold run. Existing studies on Decision Support Systems (DSS) and Automated Trading Systems based show pertinence of such techniques to traditional markets, while their application to cryptocurrencies is still a study subject. This work proposes the analysis of multiple cryptocurrencies among the most popular by market capitalization between 01/2011 and 01/2019, combining daily market data with a selection of technical analysis indicators and blockchain-derived metrics to build and analyse different trading systems based on popular machine learning algorithms and ensemble methods both in terms of performance and relevant feature interactions. Results show how prediction outcomes are generally following trends, with model precision ranging between 40% and 60%. However, when analysed through metrics who take input bias into account such as index-based accuracy (IBA), very few models reach the skill threshold, implicating class imbalance in the training data affects classification results. Trading simulation shows how the proposed systems are profitable in both bear and bullish markets yet fail to identify patterns leading to high volatility events characterising the cryptocurrency markets, giving the baseline strategy a lead over longer timespans. The work also explores the reasons behind machine learning algorithms' decisions. It applies a state-of-the-art explainable model, namely SHAP, to highlight the features that mostly influence the performance of cryptocurrency price forecasting.

Chapter 1

Introduction

In the last decades, cryptographic communities made many attempts in developing a digital currency able to be anonymously transacted between peers without the need of a trusted third party (such as a bank). The first successful attempt at such task was Bitcoin, released in 2009. Since then, many other cryptocurrencies - commonly called altcoins - have either been indipendently developed or derived from Bitcoin, in a process called "Hard Fork".

At first, people started trading cryptocurrencies directly on specialized communities in a peer-to-peer fashion. However, the need for trusting the other peer when performing a trade exposed investors to risk of being scammed, thus specialized on-line marketplaces acting as a trusted third party for continuous exchange of crypto currencies began to flourish.

Transacted volume kept growing steadily, together with interest in the asset class: as exchange price of Bitcoin kept rising, more investors were drawn to the market by the ease of access, lack of regulation and incredible profit possibility, increasing demand and driving the prices further higher. The arrival of professional and institutional traders on the markets came with demand for more advanced trading tools: exchanges started offering derivative trading tools, such as leveraged futures and options contracts, unlocking margin trading against managed liquidity pools.

Nowadays market capitalization across the whole sector has surpassed 1 Trillion USD, making cryptocurrencies an important financial reality that cannot be ignored when managing an investment portfolio.

In this work both historical market data and blockchain-derived information from multiple currencies were analysed using supervised data mining techniques, aiming to build and analyse the performance of the resulting systems in terms of models performance, profitability and explainability of the outputs.

Various data sources have been evaluated, but while market data was often freely available, the high computational costs required for extraction of blockchain data and speculative activity made finding suitable sources for blockchain data hard as most of the sources were either commercial or required access to paid subscriptions for downloading the data. Finally, market data deriving from the Kraken exchange (one among the most popular in Europe) and freely available blockchain-derived metrics from the coinmetrics community were adopted, as they covered most currencies among the top by market capitalization in the analysed periods with good accuracy.

Once source data was pre-processed and imported into the databases, one dataset per currency was built by combining well-estabilished approaches such as the use of technical indicators and historical market data with candlestick chart analysis and blockchain derived information. Unlike assets in traditional markets, each cryptocurrency presents different age and characteristics, thus the resulting datasets were not uniform both in terms of included features and number of samples.

After identifying the relevant predictive classification problem, data was labeled and analysed - detecting significant class imbalance - and split into training and validation sets. A preliminary feature selection step was performed on the training set, employing feature importances as expressed by an XGBoost classifier with default parameters: this helped in both reducing input dimensionality and providing information about the market characteristics as a whole and for each currency.

The following step was model optimization, based on the well-affirmed grid search method in combination with k-fold cross validation using a stratifying splitter to avoid problems induced by class imbalance inside each of the validation folds, and an adaptive strategy choosing the number of folds k based on the number of available samples in the training set.

Once features and hyperparameters were fixed, models were fit and used for making predictions on the validation set with a sliding window approach, experimenting different window sizes. Since dealing with imbalanced data, outputs were evaluated with both tthe standard precision metric and a metric specific to imbalanced learning, namely index of balanced accuracy: their joint use allowed to determine the trade-off between bias and variance.

A custom trading agent simulating unleveraged margin trades on a real world exchange was developed based on the rules and limitations of the Kraken exchange. Orders were placed with both stop-loss and take-profit conditional close triggers mitigating the impact of mispredictions, on the base of signals generated by the model testing phase and aiming to beat a baseline buy and hold strategy.

In parallel model outputs were analysed by means of SHAP, a model-agnostic explainability framework using game-theoretic approaches to explain outputs from black-box models, aiming to find out how the factors determining model outputs changed across the testing period.

Results for the feature selection, backtesting and explainability steps were finally presented through customized visualizations and analysed.

Trading results have shown how - while highly related to trends and missing some high volatility events - the resulting trading systems are profitable beating the baseline strategy in most cases. Another important result from trading simulation was obtained by the comparison between trading results and model tests evaluation: with the best models in terms of classification metrics not corresponding to the ones by realized profit.

Feature selection results from the training set and model explainability results from the test set were finally analysed, providing insights about the characteristics of each of the analysed currencies and about how the deciding factors in model outputs changed across the testing period.

Chapter 2

Preliminaries

2.1 Cryptocurrencies

Cryptocurrencies are digital assets whose transfers and accounting are cryptographically estabilished through a distributed ledger, also known as Blockchain. Even though they are not backed by any physical asset, they can be used as a means of payment, and bearing value for their holders, they can be considered financial assets [1].

Before Bitcoin, many attempts at building such a technology have been made like "B-Money" by Wei Dai (1998), which was based on a Pub/Sub architecture requiring centralized servers, or "Bit Gold" theorized by Nick Szabo (1998) which was decentralized, and introduced the concept of Proof-of-Work (PoW) as a consensus algorithm.

This family of consensus algorithms expects miners to solve a cryptographic puzzle for each block, broadcasting the solution to the network: if it is verified, a cryptographic hash chain linking the most recent puzzle's solution to the current one is created, validating the transactions included in the block.

Although BitGold did not make it to a usable implementation because of multiple security issues such as vulnerability to double-spending and Sybill[1] attacks, the Proof-of-Work process of posting transactions to the network remained consistant in modern cryptocurrencies.

[1] A type of attack in which a party owning a huge number of nodes can perform a 51% attack independently of total hashrate

2.1.1 Bitcoin

The Bitcoin whitepaper was released in 2009 by an anonymous collective known as "Satoshi Nakamoto". It proposed the first fully distributed ledger technology called Blockchain, a chain - as the name suggests - of cryptographically linked blocks, where each block is a cluster of transactions which are validated by each peer in a process called mining.

Mining is the process by which new coins are added to the supply, while securing the network against fraudulent transactions and double-spending [2].

In exchange for the opportunity to be rewarded Bitcoin, Miners provide processing power to the network by validating and recording new transactions to the ledger in blocks, earning two types of reward: coins minted in each new block (included in a coinbase transaction[3]), and fees from the transactions included in the block.

To earn such rewards, miners compete in solving a difficult mathematical problem based on a protocol-mandated algorithm (SHA-256) including the solution in the new block as proof the miner expended computing effort. This competition is the basis of PoW-based currencies' security model.

Game-theory based mechanisms periodically readjust the mining problem's difficulty depending on the network's hashrate, with the aim of keeping the block time[4] constant: this is 10 minutes for the Bitcoin protocol.

Transactions included in a block that is added to the blockchain are considered "confirmed", and allow the receiver to spend the received coins. Further blocks increase the number of confirmations of transactions, which is directly proportional to the trust rate: exchanges often require a certain number of confirmations before considering a transaction final.

Bitcoin is a deflationary asset: every 210,000 blocks (approximately four years in block time), the amount of coins issued with coinbase transactions (Block Reward) is decreased by 50% in an event called halving. In the first four years of operation of the Bircoin network, the block reward was of 50 bitcoin per block. The first Bitcoin halving in November 2012 decreased it to 25 bitcoins per block. This amount was decreased again to 12.5 in July 2016 and 6.25 in May 2020. The next Bitcoin halving is currently expected to happen mid 2024, at block 840.000.

[2] A kind of fraudulent transaction in which the same coins are spent more than once.

[3] A coinbase transaction is a special kind of transaction where there is no "sender"

[4] A Protocol parameter indicating the approximate time it takes for a new block to be found by miners.

2.1.2 Altcoins

In the years following the advent of Bitcoin, interest in the Blockchain technology grew constantly, and many new cryptocurrencies - commonly called Altcoins - were launched on the market attempting to ride the trend. Copies of Bitcoin originated from an actual fork from the Bitcoin Network, called "hard fork": since modifications to the protocol require consensus from the network itself, the only way to implement consistent updates is minting a new coin. Hard-forked blockchains share their parent ledger's history up until the forking point, but newly added transactions will not be backwards compatible: an example of this process is Bitcoin Cash (BCH).

New cryptocurrencies, with their own ledgers yet technically similar characteristics to Bitcoin have been developed, as well as innovative currencies such as Ethereum (ETH), whose aim is to build a decentralized smart contract platform [5]

Ethereum is the first decentralized platform for creation, publishing and execution of smart contracts written in Solidity through its distributed virtual machine based on the ERC-20 protocol, using the Ether (ETH) currency. Many coins are actually tokens on the Ethereum network such as Link, or started as Ethereum tokens then moved to their own blockchain implementation.

Cardano was released in September 2017 by a co-founder of Ethereum and BitShare, and is managed by a team of researchers and professors. It is fully open source, centered on scientific research and uses an innovative Proof-of-Stake (POS) protocol. Like Ethereum, it is a smart contract platform fueled by the ADA currency.

Bitcoin Cash or BCH was born by a hard fork from Bitcoin network in December 2017, aiming to increase block size from the native 1MB to 8MB. A further split happened in November 2018, creating two distinct currencies: BCHABC (or BCHA, Bitcoin Cash ABC) and BCHSV (Bitcoin Cash SV - Satoshi Vision), with the latter aiming to further increase block size to 32MB.

Binance Coin or BNB is a digital asset launched in mid-2017 by the Binance exchange. It was initially distributed on the Ethereum Network, and moved to its own Binance chain, based on BEP-2 protocol in April 2019. It grants the trader

[5] Smart contracts are the name of the solution chosen by the Ethereum team, that allows executing software through a distributed virtual machine platform

a discount when used to pay exchange fees on the Binance exchange, which also allows converting small amounts left in the wallets to BNB. Later on, in April 2020 it received an upgrade allowing it to host smart contracts thanks to a dual-chain architecture which introduced the Binance Smart Chain.

Bitcoin Gold or BTG originated from a Bitcoin hard fork in late 2017, aiming to make mining decentralized again: the issue with Bitcoin was that huge hashpower needed to mine profitably required specialized machines known as ASICs as well as huge amounts of electricity, with the effect of centralizing mining in regions where the latter is cheaper. While keeping Bitcoin fundamentals intact, BTG could be mined by anyone with readily available graphics cards.

Digital Cash or DASH is an open source digital currency born in 2014, initially named XCoin, then DarkCoin and finally, in 2015 named Digital Cash. It focuses on anonimity thanks to its PrivateSend functionality, and confirmation speed thanks to InstantSend. Nowadays it is among the top-ranked currencies by market capitalization.

Dogecoin or DOGE is a controversial currency, born as a joke in December 2013 by forking the Litecoin codebase. It owes the name to the "Doge" meme diffused on the internet, and being based on the same algorithm as Litecoin (SCRYPT) it can be joint-mined, meaning PoW solutions are interchangeable between the two currencies. Nowadays Dogecoin is one of the most discussed currencies, among the top-ranked by marketcap.

Ethereum Classic or ETC was born as a hard fork from the Ethereum network. In 2015 Ethereum developers made some changes in the codebase, in order to repay victims of a hacker attack: some developers did not accept the change, and chose to hard fork the original platform, generating a parallel currency.

EOS.IO or EOS was initially a smart contract on the Ethereum platform, released in mid-2017. Its mainnet launched in June 2018, aiming to build a decentralized operating system for the execution of dAPPS (Decentralized Applications), based on an innovative protocol called dPOS (Delegated Proof-of-Stake) where users "vote" for a third party to stake their coins, earning dividends of the profits coming from their stake.

Chainlink or LINK is an Ethereum token aiming to ease interoperability between smart contracts and external sources from which the majority of information is retrieved. Data and services exchanges between providers and users are regulated by the LINK token.

Litecoin or LTC was forked from Bitcoin in 2011, reducing block time to 1 minute in order to speed up confirmation time for transactions. In turn, mining was made more difficult by adopting the SCRYPT algorithm, which forces a certain number of turns before reaching a solution.

NEO is the first chinese open-source and decentralized smart contract platform. Initially known as "Antshares", it is often compared to Ethereum, but it allows the use of multiple different programming language and runs on a proof of stake decentralized Byzantine fault tolerant (dBFT) consensus mechanism between a number of centrally approved nodes.

Quantum or QTUM is a hybrid solution developed in March 2016 in Singapore, aiming to build a smart contract platform on the Bitcoin blockchain model. Due to its flexibility it was targeted to big chinese businesses, but lack of adoption has caused its delisting from many of the western services.

Tronix is based on the TRON blockchain, born as a non-profit in late 2017 as a sharing platform with a focus on digital entertainment and DRM (digital rights management). TRX is the token allowing users access to the content published on the network.

Waves is a blockchain allowing token creation through provided tools and implementing a marketplace for trading of other currencies (including FIAT): the WAVES wallet is a multi-coin wallet, hosting a multitude of tokens.

XEM is the currency used on the NEM (New Economy Movement), a java-based smart contract network. Its characteristic is the Proof-of-Importance algorithm, a PoS variant weighting nodes based on their activity and not only on their stake.

Monero or XMR is a fully decentralized, Bitcoin derived currency with privacy as its main goal. It allows anonymous transactions using disposable addresses, and does not pose limits on block sizes meaning it is very scalable. It uses the CryptoNight algorithm, developed for granting privacy and incentivizing decentralization because it is only executable on CPU and GPU’s.

Ripple or XRP is not a blockchain but an open source exchange protocol based on consensus. All the circulating supply was pre-mined at platform release in 2013, and further mining is not allowed. It is a heavily centralized system regulated by the Ripple foundation, and supported by banks.

ZCash or ZEC originates from bitcoin but focuses on transparency and security. Its primary objective is privacy: it employs a Zero-Knowledge Proof strategy to validate transactions while keeping the amount and the sender private. It was released in late 2016.

0x or ZRX is an Ethereum-based protocol for fast token exchange with low fees.

2.1.3 Exchanges

With the development of more and more cryptocurrencies, many virtual platforms were created specifically for their exchange, initially facilitating spot trades while not being affiliated with the currency's creators.

Often placed outside of the western countries to avoid regulation and prosecution, exchanges are mostly strictly online businesses, providing various graphs and technical indicators and allowing customers to trade cryptocurrencies for other assets such as conventional fiat money or other digital currencies as well as withdraw of converted funds 24 hours a day / 7 days a week, which means crypto can be traded any time and any day. According to local regulations, in some cases, credit card payments, wire transfers or other forms of payment can be accepted in order to top-up fiat balances to use in trades. Binance, Coinbase, Gemini and Kraken are nowadays some of the big names in this field.

The issue with this kind of services is that they are custodial, meaning assets are held in the exchange's wallet, and incidents such as the 2014 Mt.Gox bankruptcy might happen, exposing investors to potentially unrecoverable loss. The need of trusting a third party for trades lead to the creation of Decentralized Exchanges who are non-custodial (they do not store user's funds), instead they only act as order matchers for facilitating peer-to-peer cryptocurrency trading.

Cryptocurrency exchanges can be market makers that typically take the bid–ask spreads as a transaction commission for their service or, as matching platforms, simply charge fees. Over time the range of offered services increased introducing financial instruments inspired by the ones already available on the stocks'secondary markets such as futures, allowing users to do leveraged margin and options trading.

2.2 Trading

Trading is the process of exchanging securities or shares between sellers and buyers, aiming to increase profit from speculative operations that exploit market volatility. Before internet, trading was reserved to few individuals called brokers, who acted as middlemen between investors and the market operating the exchanges in person. Nowadays, trading is more and more associated with the "Online" world, anyone can access stocks trading through a brokerage account (such as ICMarkets) or dedicated applications such as eToro or RobinHood, attracting retail investors. The same goes for cryptocurrency exchanges: until mid-2018 exchanges did not even require a KYC (Know Your Customer) filing for registration, offering retail and institutional investors alike another option for differentiating their portfolios.

Spot Trading is the process of buying and selling assets for immediate delivery. Assets (or perpetual contracts representing their ownership) are directly transferred between market participants (buyers and sellers). When spot trading cryptocurrencies, users have direct ownership (through their hosted wallet) over the traded assets: exchanges act as intermediaries for buyers and sellers to bid and ask, facilitating trades when a bid or offer is matched.

Margin Trading is the process of subscribing contracts with an agreement to buy or sell a specific asset at a future date. As such, ownership of a future contract does not grant ownership of the underlying asset. This is advantageous because users can speculate on an asset without actually buying it: if the user expects the price to go up he will buy a futures contract to go long, otherwise he will sell to go short. Profit or loss will depend on the outcome of the user's prediction. Mechanisms for managing risk (and therefore increase the risk/reward ratio) such as leverage are also available, as well as insurance funds to act as safeguards.

2.2.1 Candlestick Data

Candlestick data has been historically used in trading since the 18th Century, when a Japanese man named Homma discovered that while there was a link between supply and demand in the price of rice, this was also influenced by traders' emotions and he emphasized so by developing this kind of chart.

This data representation Figure 2.1 derives from concluded orders from an order book [6] aggregated over defined intervals, assigning to each interval a tuple of values:

[6] An orderbook entry is typically composed of four fields: timestamp, amount, price, buy/sell flag

Figure 2.1: BTCUSDT pair candlestick chart as of 2020/11. Screenshot taken from Binance <https://www.binance.com>

- Open: price at the beginning of the aggregation interval
- High: highest price in the aggregation interval
- Low: lowest price in the aggregation interval
- Close: price at the end of the aggregation interval

Candlestick data is visually represented as a sequence of bars (Figure 2.2), with each bar's color depending on the difference between closing and opening price: if the price increased, the candle will be green (or empty) and called "Bullish", otherwise it will be red (or full) and called "Bearish". High and Low prices are represented respectively as "Wicks" and "Shadows", representing selling and buying pressure on the market.

Candlesticks build patterns that predict price direction once completed, these can identify bearish or bullish trends, and have been used for centuries to predict price direction. There are various candlestick patterns used to determine price direction and momentum, with the most various names including three line strike, two black gapping, three black crows, evening star, and abandoned baby.

Volume data is often plotted along candlestick graphs, aggregated in bars representing the absolute sum of the units exchanged in a time interval, and similarly to candles, it can be obtained by aggregating orders. Trading volume can help investors in identifying momentum in a market and confirming trends: if it increases, price action should not change its direction, meaning if volume is increasing while price is in an uptrend, the uptrend should continue. Vice versa, volume can also signal when an investor should take profits and sell an asset due to low activity. No relationship between trading volume and price movement signals weakness in the current trend and a possible reversal.

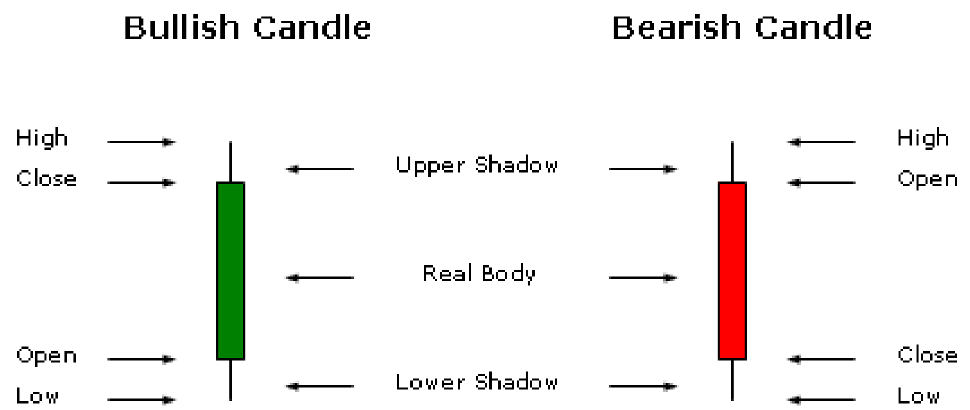

Figure 2.2: Ticmarc, 2007, Candlestick Chart Definition, Wikimedia Commons <https://commons.wikimedia.org/wiki/File:Candle_Definition.png>, accessed Jun. 2021

2.2.2 Technical Indicators

Technical indicators are time-independent statistical indicators and oscillators built on previous observations of stock data, often used for multi-day trading. Over the years, dozen of indicators have been designed by statistical economists and quantitative traders, who usually plot them on candlestick charts since they share the same x-axis.

Indicators take into account price action and exchanged volumes, giving insights in mainly four classes: trend, momentum, volatility and volume.

Indicators may be bounded or unbounded, and each of them has its own interpretation.

Bounded indicators are also called oscillators, commonly defined in the range [0, 100] or over a symmetrical range w.r.t. the origin (eg. [-100, 100]) with thresholds setting the boundaries for important areas: when the oscillator crosses the threshold line, valuable information (such as overbought or oversold conditions) can be inferred.

Unbounded indicators often employ a signal line: when the indicator crosses such a line, information such as a trend reversal can be inferred.

Indicators used in this work will be discussed in the following chapters.

2.2.3 Strategies

A trading strategy is a systematic method used for buying and selling in a security market, it is based on pre-defined rules and criteria used when making decisions. It is typically developed in advance, taking into account various factors such as investing objectives, risk tolerance, time horizon and fees. The key to a successful trading strategy is using objective data and analysis in its development phase, diligently adhering to it while periodically re-evaluating and tweaking its parameters as market conditions or investment goals change in the execution phase. Trading strategies are usually based on analysis of one or more aspects of market data:

- Technical Analysis: is the study of the market based on indicators, statistic oscillators and graphs modeling repeating behavior in historical prices of an asset
- Fundamental Analysis: is the study of aspects that can influence the market, for cryptocurrencies among these we have developer's trustworthiness, project activity and innovation, real-world applications and on-chain activity
- Sentiment Analysis: is the study of expert and analysts' opinion based on information from social media, blogs and sectorialized news

Trading strategies are often categorized by how decisions are taken:

- Discretionary trading: solely based on the trader's knowledge and capabilities in analysing graphs and market indicators to identify trading opportunities
- Quantitative trading: rely on the use of automated systems, computations and algorithms to analyze market data and identify trends and patterns the trader can use to find trading opportunities

Or by their positions' duration:

- Scalping: aims to open and close positions in small timeframes, in order of minutes or even seconds. Scalpers' aim is not to leverage a market trend but to accumulate small profits limiting losses as much as possible.
- Intraday: is the most diffused trading strategy, positions are opened and closed within the same trading day, speculating on daily price action. This allows a good control over losses, at the cost of limiting profits.
- Multiday: is a strategy in which positions are kept open for longer amounts of time, such as weeks or even months. It is based on the assumption that "history repeats itself", and is the most difficult kind of strategy because traders

have to forecast how the market will be in the future. To do so, they try to identify ongoing trends, ignoring oscillations on lower timeframes. This allows way higher gains, at the cost of exposing capitals to higher risks.

In all cases, the common objective of developing a trading strategy is reducing trading operations to an algorithm, in order to minimize behavioral/emotional biases and ensure consistent results.

Orders in a strategy can be divided by how they are executed, in:

- Market Order: open a position where entry price is exchange price at time of order execution, only specifying the position size in terms of collateral
- Limit Order: open a position where entry price and position size are fixed and specified when the order is placed

Often, when placing an order, one or more conditional exit orders are placed as well. The most used conditionals are:

- Stop Loss: places a trigger for a limit sell order, usually below (or above in case of short) the entry price. It is used to limit losses in case of misprediction.
- Take Profit: places a trigger for a limit sell order usuually above (or below in case of short) the entry price. It is used to book profits and reduce risk.

2.2.4 Trading Systems

A Trading System is a set of predefined rules, implemented by mathematicians and informatics, allowing to autonomously define entry, exit and money management strategies in order to generate a profitable strategy for the investor.

Trading can be negatively influenced by emotions: tiredness, stress, fear, greed and euphoria play an important role in how humans think and act, with potentially disruptive consequences for the trading strategy. In fact, most profitable traders are the ones who can detach themselves from their emotional sphere and only take decisions based on an objective analysis of the market.

That's where automated trading systems come to shine: their decisions are algorithmic and data-driven, can be monitored and tested on historical data (backtesting), they don't get tired and can operate at higher frequencies than their human counterparts without stops, even on multiple markets. On the other hand, automatic trading systems base their decision-making on empirical rules which are often not validated on a trustworthy data set and can overfit, therefore they can become unreliable, requiring constant monitoring.

Trading Executor is an agent executing orders and handling positions in accordance to signals generated by a strategy. It can be a human trader, using algorithmic output as support (DSS - Decision Support Systems), or a fully automated agent autonomously taking decisions based on signals.

2.3 Data Mining and Machine Learning

Data is a great source of knowledge, but with more and more data accumulated every year it becomes increasingly difficult to extract useful information from it. Data Mining is a process that allows extracting information from large amounts of data by combining traditional statistics methods with complex algorithms, aiming to identify models describing the data or predicting outcomes of future observations.

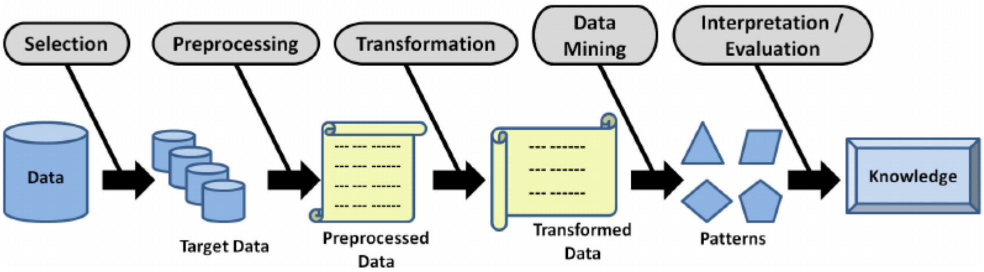

Figure 2.3: Francesco Gullo, 2015, The Knowledge Discovery in Databases (KDD) process, ResearchGate <https://www.researchgate.net/figure/The-Knowledge-Discovery-in-Databases-KDD-process_fig1_274425359>, accessed Jun. 2021

Data Mining is part of a process called Knowledge Discovery of Databases (KDD, Figure 2.3), which is composed of different phases:

- Data Collection: data is collected from various sources and stored in dedicated platforms, which can be centralized or distributed
- Data Preprocessing: is the process of integration, cleaning and transforming of data in a form allowing its use in further steps
- Data Mining: algorithms are applied to pre-processed data to extract information
- Data Postprocessing: extracted informations and patterns are checked and transformed for visualization or application

2.3.1 Machine Learning

While data mining is the process of extracting useful information from vast amounts of data, machine learning is the process of discovering algorithms that leverage past knowledge to modify their behavior, effectively allowing machines to learn without human intervention.

Machine learning tasks can be divided in two macro-categories:

- Predictive Tasks: develop models able to forecast an attribute's value from many others
- Descriptive Tasks: develop models able to explain and identify relationships between data

Predictive tasks can further be split in classification and regression tasks, by the type of their output:

- Classification tasks approximate a mapping function (f) from input variables (X) to one or more discrete output variables (y). For example, an email can be classified as belonging to one of the two classes: "Spam", "Not Spam"
- Regression Tasks approximate a mapping function (f) from input variables (X) to a continuous output variable (y). For example, a car can be predicted to sell for a specific dollar value, perhaps in the range $5000-$10000

Classification models are the most suitable for predictive tasks, many classifiers have been developed such as decision trees, rule-based classifiers, support vector machines, neural networks and so on. In the following paragraphs the classifiers used in this work will be briefly discussed.

2.3.2 Classification Models

Classification models can be categorized in unsupervised and supervised learning models. The main distinction between the two is the use of labeled datasets: supervised learning uses labeled input and output data, while unsupervised learning algorithm does not.

Unsupervised learning models discover the inherent structure of unlabeled data on their own, while in supervised learning, the algorithm "learns" from the training dataset by iteratively making predictions on the data and adjusting for the correct answer.

Since we can easily obtain labels from market data, in this work we analyzed the use of the most diffused supervised learning algorithms:

Multinomial Naive Bayes

is a classifier based on Bayes' theorem, where the adjective Naïve says that features in the input dataset are mutually independent. In other words occurrence of one feature should not affect the probability of occurrence of the other features, otherwise the resulting model will not be valid. The picked class label corresponds to the most probable given the training set, making this model highly sensitive to class imbalance.

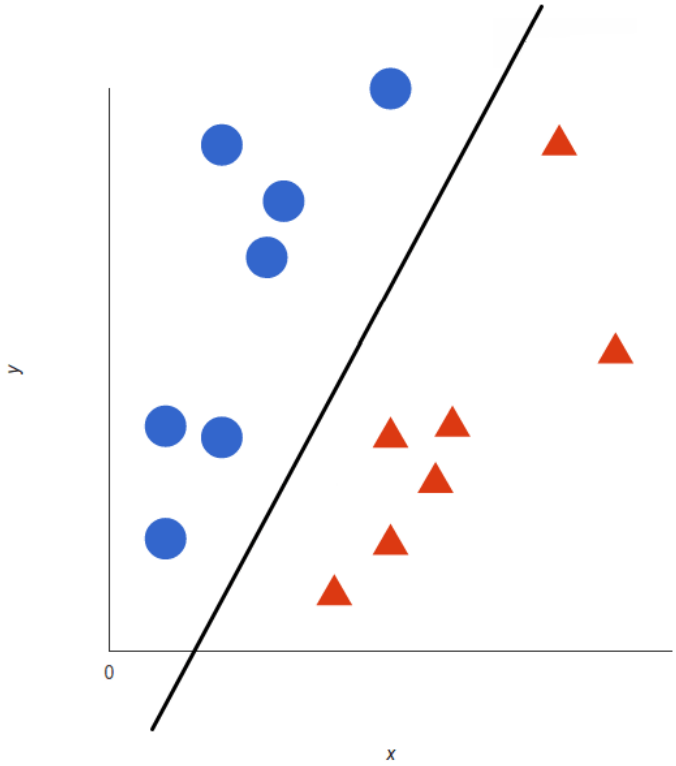

Figure 2.4: James Le, 2018, "Scatter plot representing features and the relative separating hyperplane for a linear, binary classification problem", DataCamp, <https://www.datacamp.com/community/tutorials/support-vector-machines-r> accessed Jun. 2021

Support Vector Machines

SVM are a family of classifiers aiming to find an hyperplane (Figure 2.4) separating data points in the classes of interest. If the problem is linear this can be done without further processing, otherwise the search space needs to be transformed in a multi-dimensional space, transforming input data in a set of points in the eculidean space and finding a separating hyperplane.

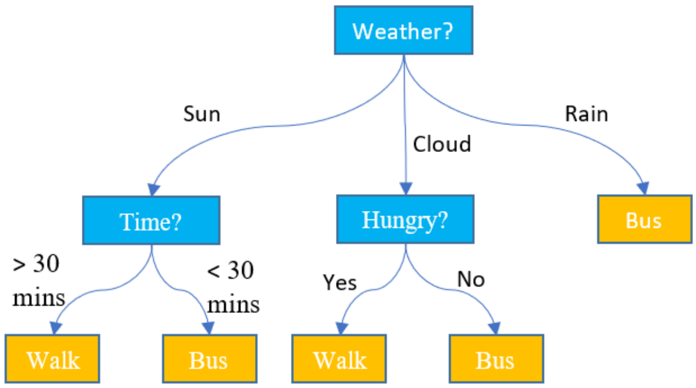

Figure 2.5: Jake Hoare, A decision tree predicting how to make the journey to work, DisplayR Blog <https://www.displayr.com/what-is-a-decision-tree/>, accessed Jun. 2021

Decision Tree

is a classifier based on a graph (Figure 2.5) that can be walked from the root to the leaf nodes, expression of possible solutions to the problem. Each node is a variable evaluation point, each of the branches stemming from a node represents an outcome of said evaluation. Walking path is determined by a "purity" evaluation, calculated differently based on the chosen loss function.

k-Nearest Neighbors

is a classifier using an unique approach: it does not use the training set to extract knowledge or patterns, instead when presented with new inputs it calculates the distance between the new data and the training data, assigning the output by majority voting among the labels of the top-k nearest instances.

Multi-Layer Perceptron

is a classifier belonging to the category of Artificial Neural Networks (or ANN), one of the most recognized examples of mimesis through which a fully connected graph of nodes (called Neurons) emulates the way biological brains work. In MLP

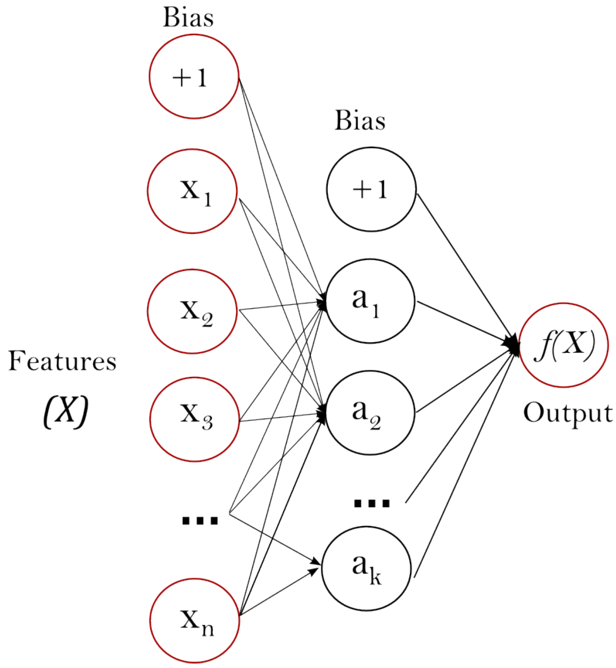

Figure 2.6: Scikit-Learn Developers, 2020, One hidden layer MLP, Scikit-Learn Documentation <https://scikit-learn.org/stable/modules/neural_networks_supervised.html>, accessed Jun. 2021

artificial neurons can be represented as an oriented graph (Figure 2.6), subdivided in layers belonging to different categories:

- Input Layer: input features are transformed through an activation function f(z) in order to reach specific target values
- Hidden Layers: contain intermediate nodes whose goal is to understand relationships between input and output data through training examples, the number of hidden layer should be proportional to the complexity of the input-output relationship
- Output Layer: inverse-transforms hidden layers' output through the activation function, reproducing the target values

The training process is iterative, pursuing to find a weight Wi for each i-th neuron through different approaches, the most common being back-propagation combined with stochastic gradient descent (SGD), in order to minimize a cost function (such as MSE).

2.3.3 Ensemble Classification

Ensemble classification techniques exploit the use of multiple learning algorithms (or "weak learners") to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. The main reasons for using an ensemble over a single model are:

- Performance: ensembles usually make better predictions and achieve better performance than any single contributing model
- Robustness: ensembles reduce the spread or dispersion of the predictions, increasing model performance

The trade-off for better performance is greater model complexity, and therefore computational cost. Among the many types of ensemble techniques, the ones used in this work are:

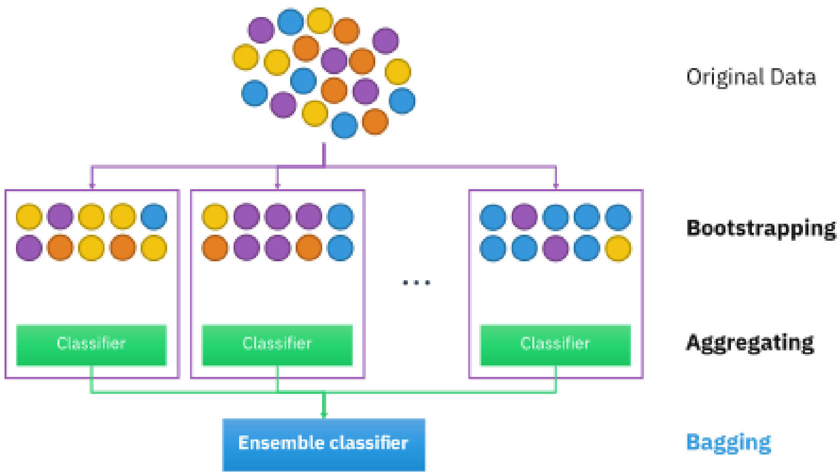

Figure 2.7: Sirakorn, 2020, Illustration of a bootstrap aggregating (bagging) method for ensemble learning, Wikimedia Commons <https://commons.wikimedia.org/wiki/File:Ensemble_Bagging.svg>, accessed Jun. 2021

- Bootstrap-aggregating (bagging) ensembles: is a parallelizable process (Figure 2.7) building an ensemble which promotes model variance by training each model with a different subset of the samples in the training set, which are randomly drawn with replacement. Results from each learner are then combined by majority voting.
- Boosting: is a process (Figure 2.8) building an ensemble starting from a naive model, iteratively training each subsequent model by emphasizing the training

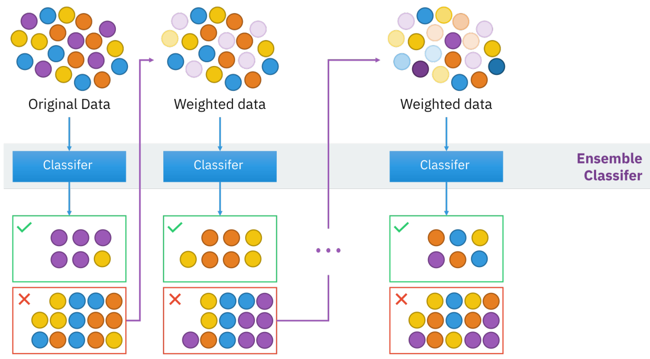

Figure 2.8: Sirakorn, 2020, Illustration of a boosting method for ensemble learning, Wikimedia Commons <https://commons.wikimedia.org/wiki/File:Ensemble_Boosting.svg>, accessed Jun. 2021

instances that the previous model mis-classified. In the training phase, an equal weight is initially given to each sample training data (uniform probability distribution). This data (D1) is then given to the first (naive) base learner (L1). The mis-classified instances are then given an higher weight than the correctly classified ones, but keeping the total probability distibution equal to 1. The resulting boosted data (D2) is then given to the next base learner adnd so on. Again, results are combined by majority voting.

Random Forest is an ensemble classification algorithm combining random decision trees with bagging. The main disadvantage of this technique is the tendency to overfit training data as the tree grows deeper, while the main advantage is that by their nature, the resulting models are highly explainable and allow for extraction of useful insights such as feature importances.

eXtreme Gradient Boosting or XGBoost is a popular ensemble classification algorithm based once again on random decision trees, using a bagging + boosting approach which can be tuned by its hyperparameters. Learners are fit using any arbitrary differentiable loss function and gradient descent optimization algorithm, naming the technique.

2.3.4 Imbalanced Classification

An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is biased or skewed: this is the case for real-world data. The challenge of working with imbalanced datasets is that most machine learning techniques were designed around the assumption of an equal number of examples for each class and will ignore the minority class, impacting performance even though performance on the minority class is typically most important.

One approach to addressing imbalanced datasets is to oversample the examples in the minority class by duplicating samples in the training dataset before fitting a model: this can balance the class distribution, but does not provide any additional information to the model.

Another approach is to synthesize new samples from the minority class: this is a well-known type of data augmentation for tabular data, and can be very effective.

Synthetic Minority Oversampling TEchnique

SMOTE is the most widely used approach for oversampling the minority class. It works by selecting examples that are close in the feature space, connecting them by a line and drawing a new synthetic sample at a point among that line. First, a random example is chosen from the minority class. Then k of the nearest neighbors for that example are found (typically employing a k-NN classifier with k=5). One of these neighbors is randomly chosen and connected to the original sample, then a new synthetic sample is drawn along the line connecting the two. The procedure can be used to create as many synthetic examples for the minority class as are required, causing the classifier to build larger decision regions containing nearby minority class points. However, a general downside to this approach is that samples are created without keeping majority classes into account, possibly resulting in ambiguous examples if there is a strong overlap between the classes.

2.3.5 Evaluation Metrics

Evaluation Metrics quantify the performance of a predictive model, playing a crucial role in data preparation and classifier modeling steps.

For classification problems, metrics involve comparing the expected class label to the predicted class label or interpreting the predicted probabilities for the class labels for the problem, influencing how the importance of different characteristics in the results are weighted and the ultimate choice of which algorithm to choose: since selecting a model and even the data preparation methods together are a search problem that is guided by the evaluation metric, choosing a wrong metric may lead to choosing the wrong model, or in the worst case being misled about its expected performance.

There are standard metrics that work well on most problems and are widely used for evaluating classification predictive models, such as classification accuracy or classification error, but all of them make assumptions about the problem or what is important in the problem: the challenge is finding an evaluation metrics that best captures what is important about the model's purpose and the use of its predictions.

This challenge becomes even more difficult when there is a skew in the class distribution of the input data, because standard evaluation metrics treat all classes as equally important, while imbalanced classification problems typically rate classification errors with the minority class as more important than those with the majority class. As such performance metrics that focus on the minority class may be needed.

Confusion Matrices

		predicted class	
		Spam	Ham
true class	Spam	True Positive (TP)	False Negative (FN)
	Ham	False Positive (FP)	True Negative (TN)

Figure 2.9: Unknown, 2017, Sample confusion matrix for a Spam/Ham binary classification model, Thinbug <https://www.thinbug.com/q/25343411>, accessed Jun. 2021

Confusion matrices (Figure 2.9) summarize how successful a classification model's predictions were, in other words they represent the correlation between the model's output and the actual label using NxN tables. One axis of a confusion matrix is the label that the model predicted, and the other axis is the actual label. N represents the number of classes. In a binary classification problem, N=2.

From these matrices information giving insights in the performance of a predictive model and which classes are being predicted correctly, which incorrectly, and what type of errors are being made can be derived:

- True Positives (TP) are outcomes where the model correctly predicts the positive class.

- True Negatives (TN) are outcomes where the model correctly predicts the negative class.

- False Positives (FP) are outcomes where the model incorrectly predicts the positive class.
- False Negatives (FN) are outcomes where the model incorrectly predicts the negative class.

Classification metrics

Standard classification metrics combine insights from the confusion matrix to further emphasize certain aspects of classification performance. We will now briefly discuss the most common, then finally review them:

Accuracy is the ratio between correct outputs (both true positives and true negatives) and the total number of inputs examined.

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Error is the ratio between incorrect outputs (both false positives and false negatives) and the total number of inputs examined.

$$\frac{FP + FN}{TP + TN + FP + FN}$$

Precision is the number of true positives (i.e. the number of items correctly labelled as belonging to the positive class) divided by the total number of elements actually belonging to the positive class (i.e. the sum of true positives and false positives).

$$\frac{TP}{TP + FP}$$

Sensitivity also known as Recall or True Positive Rate is the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives).

$$\frac{TP}{TP + FN}$$

Specificity or True Negative Rate is the number of true negatives divided by the total number of elements that actually belong to the negative class (i.e. the sum of true negatives and false positives, which are items which were not labelled as belonging to the positive class but should have been).

$$\frac{TN}{TN + FP}$$

2.4 Infrastructure

The elaboration for this work was carried on various distributed computers, relying on several centralized services for features and results storage.

Metadata MongoDB is an open-source cross-platform document-oriented NoSQL database program. It uses JSON-like documents - with its own BSON standard - with optional schemas. It is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL). We used it to store experimental results and metadata such as model parameters.

Timeseries InfluxDB is an open-source time series database (TSDB) developed by InfluxData. Its data model is based on the concept of "measurement", which is a grouping of "series". A series is a temporal sequence of "points" identified by "tags", each series is composed of several key-value pairs called "fieldset". Although a MySQL database would be suitable, the need of schemas and data mapping - as well as performance - made InfluxDB the most natural choice for storing and processing features.

Data Storage MinIO is an Amazon S3 compatible server-side software storage stack, released under the Affero General Public License Version 3 License. It can handle any kind of unstructured data, with the maximum size of 5TB. We used it to store serializable data which could not be stored on databases such as trained models and plots.

Containers and Orchestration The whole infrastructure for this work is run in Docker containers, orchestrated by Docker-Compose. Containers are the lightweight equivalent of virtual machines, isolated from one another and exploiting OS-Level virtualization to achieve a lower footprint in terms of used resources, while Docker-Compose is an orchestration tool used for running multiple containerized services, allowing configuration via yaml files and allowing to perform start-up and shut-down process of said files via a single command.

Chapter 3

Input Data

3.1 Sourcing and Preprocessing

Due to the youth and non uniformity of cryptocurrency markets, data sourcing posed different challenges:

- Different age for each included cryptocurrency involve different amounts of data points
- Different implementations involve different metrics available for different currencies. One example would be Gas fees which are used for fueling Ethereum smart contracts: they would have no meaning for Bitcoin.
- Deprecation of some of the cryptocurrencies included in the study involves unavailability of some metrics. For example no blockchain data is available for the QTUM/USD pair.
- Main-net launches in some of the currencies included involve different (some times overlapping) datasets are available for the same pair, because it is in fact a different blockchain
- Localization involves different temporal labeling for different exchanges in different countries: most exchanges offer price data in their local timezone, which is not always UTC!
- Speculation and crescent interest in data-driven approaches make the most useful and valuable data (ie. lower timeframes) only available on commercial subscriptions.

Since OHLCV data was already available from previous work, crawling efforts were oriented in finding suitable on-chain and social/sentiment data from various sources.

We will now discuss each candidate source, then analyze the used data and applied transformations in the following paragraphs.

- Blockchain data would normally be freely available by using specific tools such as blockchain-ETL (an acronym for Extract, Transform and Load) which decode its binary data structure and dump it to CSV format. Unfortunately, since Cryptocurrencies are developed by different groups of people following different principles they do not share the same data structures and implementations, therefore each would need a different ETL tool which is not always available. Moreover, since blockchains are usually very large and complex, these tools take huge amounts of time and processing power to run.

- Blockchain Graph data would be available on Google Big Query, but it required a commercial Google Cloud Platform bucket subscription for download and processing, and was therefore discarded.

- Google Trends data was scraped for the cryptocurrency pairs and names, by exploiting the graph backend API. Unfortunately results were untrustworthy due to rolling normalization applied for plotting the data, and noise from coincident searches (for example, Cardano is both the name of the ADA cryptocurrency and a school in Texas)

- CryptoCompare social data included a few interesting metrics such as social statistics and developer activity, it was crawled through the available API but the free tier only allowed access to the last two years of data, masking older datapoints for commercial subscribers only.

- CryptoCompare blockchain data includes a set of metrics which is a subset of what coinmetrics provided, and was therefore discarded

- Quandl marketplace offered downloadable blockchain information such as hashrate and difficulty, but it only covered Bitcoin

- CryptoDataDownload offered downloadable aggregated market OHLCV data for most included currencies from different exchanges in various timeframes, but timezones were not uniform

- Kraken offered tick data (single trade) in UTC timezone: this was the source for our candlestick and volume data

- Coinmetrics offered aggregate blockchain data ready for download, it covered most of the analyzed currencies although offering different metrics for each.

3.1.1 Market Data

Candlestick data is composed of candles resulting from daily aggregation of tick data for USD pairs from the Kraken exchange.

The source data was provided in two tables containing data from all currencies (one for OHLC, one for volume data), stored in CSV files split by year.

The preprocessing step involved joining the source CSV files in one big DataFrame. Data (both for OHLC and volume) from each currency was then spliced and trimmed, in order to get tuples of five time series:

$$open, high, low, close, volume$$

Resulting data was imported as a measurement in the timeseries database for further processing.

3.1.2 Blockchain Data

Blockchain data was downloaded from CoinMetrics, using the free community API key (NULL). Downloaded data is already aggregated on daily granularity on end-of-day UTC timezones, and provided in tables stored in CSV files split by pair.

Downloaded data covered most of the assets, providing a good range of end-of-day aggregate metrics covering many aspects of the blockchain, while being consistent with our already acquired OHLCV data both in terms of timezone, granularity and USD exchange price.

The pre-processing step involved converting dates to the timezone-aware format, then filtering features by variance thresholding, dropping series which are stationary or contain too many null values (because stationary features don't add information to the model). Due to different blockchain implementations, each cryptocurrency expresses a different subset of features, as shown in Table 3.1.

In the case of coins hosted on a different chain that moved on their own chain such as Binance Coin (BNB) which moved from being an Ethereum (ERC-20) token to hosting its own Binance Chain (BEP-2), only the table including the most usable datapoints has been considered.

Over the course of this research provided data expanded, including many more metrics than what was initially acquired while dropping support for some deprecated currencies. Where possible, such metrics have been integrated and used, while dropped currencies were tackled by using the initially downloaded data.

Resulting preprocessed data then imported as measurement in the timeseries database for further processing.

symbol	ADA	BCH	BNB	BTC	BTG	DASH	DOGE	EOS	ETC	ETH	LINK	LTC	NEO	QTUM	TRX	WAVES	XEM	XMR	XRP	ZEC	ZRX
adractcnt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
adrbal1in{N}cnt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
adrbalcnt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
adrbalntv{N}cnt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
adrbalusd{N}cnt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
asseteodcompletiontime	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
blkcnt	Y	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	Y	N	Y	Y	Y	Y	Y	Y	N
blksizemeanbyte	Y	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	Y	N	N	Y	N	Y	N	Y	N
capact1yrusd	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
capmvrvcur	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
capmvrvff	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
capmrktcurusd	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
capmrktffusd	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y
caprealusd	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
feemeanntv	Y	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	Y	Y	Y	Y	Y	N
fee{A}usd	Y	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	Y	N	N	Y	Y	Y	Y	Y	N
feemedntv	Y	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	Y	Y	Y	Y	Y	N
feetotntv	Y	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	Y	Y	Y	Y	Y	N
ndf	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
nvtadj	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	N	Y	Y	Y
nvtadj90	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	N	Y	Y	Y
nvtadjff	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
nvtadjff90	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
pricebtc	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
priceusd	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
roi1yr	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
roi30d	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
ser	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
splyact{T}	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
splyactpct1yr	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
splyadrbal1in{N}	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
splyadrbalntv{N}	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
splyadrbalusd{N}	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
splyadrtop{N}	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
splycur	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	Y	Y	N
splyff	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y
txcnt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
txcntsec	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
txtfrcnt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
txtfrval{A}ntv	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
txtfrval{A}usd	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
velcur1yr	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	Y	Y	Y
vtydayret180d	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
vtydayret30d	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
difflast	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	N	Y	N
diffmean	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	N	Y	N
feebytemeanntv	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	N	Y	N
hashrate	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	N	Y	N
isscontntv	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	Y	N	N	Y	N	Y	N
isscontpctann	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	Y	N	N	Y	N	Y	N
isscontpctday	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	Y	N	N	Y	N	Y	N
isscontusd	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	Y	N	N	Y	N	Y	N
isstotntv	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	N	N	Y	N	Y	Y
isstotusd	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	Y	N	N	Y	N	Y	Y
revalltimeusd	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	N	Y	N
revntv	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	N	Y	N
revusd	N	Y	N	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	N	Y	N
splyexpfut10yr	N	Y	N	Y	Y	N	Y	N	Y	Y	N	Y	N	N	N	N	N	Y	Y	Y	N
blkwghtmean	N	N	N	Y	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N
blkwghttot	N	N	N	Y	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N
flowinexntv	N	N	N	Y	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
flowinexusd	N	N	N	Y	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
flowoutexntv	N	N	N	Y	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
flowoutexusd	N	N	N	Y	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
flowtfrfromexcnt	N	N	N	Y	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
hashrate30d	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
revhashntv	N	N	N	Y	N	N	N	N	Y	Y	N	N	N	N	N	N	N	Y	N	N	N
revhashratentv	N	N	N	Y	N	N	N	N	Y	Y	N	N	N	N	N	N	N	Y	N	N	N
revhashrateusd	N	N	N	Y	N	N	N	N	Y	Y	N	N	N	N	N	N	N	Y	N	N	N
revhashusd	N	N	N	Y	N	N	N	N	Y	Y	N	N	N	N	N	N	N	Y	N	N	N
splyminer0hopallntv	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
splyminer0hopallusd	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
splyminer1hopallntv	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
splyminer1hopallusd	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
gaslmtblk	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N	N
gaslmttx	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N	N
gasusedtx	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N	N
txtfr	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

Table 3.1: Feature distribution table for CoinMetrics' blockchain data. Similar features have been grouped together, replacing amounts with: {N} for numeric amounts, {A} for different aggregationss and {T} for different time intervals. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

3.2 Feature Engineering

Feature engineering involves processing done over source data and the creation of new features with the aim of expressing hidden patterns and information to the model.

Market and blockchain data have been treated separately: candles and volume data were processed to extract relevant information from candles and derive technical indicators, while blockchain data was further refined and combined to simulate popular models among the crypto communities.

3.2.1 Market Data

Candlestick Prices

Different approaches have been tested and adopted to remove trend and seasonality from price timeseries data. Other than raw prices, the following features have been developed and included in the final dataset:

- Percent variation of
$$open, high, low, close, volume$$
with the "_pct" suffix using the formula:
$$X_{t\_pct} = \frac{X_t - X_{t-1}}{X_{t-1}}$$
- Residuals resulting from STL Decomposition [1] for
$$open, high, low, close$$
series, with the "_resid" suffix
- First and second derivatives for each point of the approximated SPLINES [2] for each of
$$open, high, low, close$$
series, with the suffix "_spl_d1" and "_spl_d2" respectively.

[1]STL is a versatile and robust method for decomposing time series, implemented in the "statsmodels" library (statsmodels.org). STL is an acronym for “Seasonal and Trend decomposition using Loess,” while Loess is a method for estimating nonlinear relationships. The STL method was developed by R. B. Cleveland, Cleveland, McRae, & Terpenning (1990).

[2]Spline is a special function defined piecewise by polynomials used for interpolation. Price data was iteratively interpolated, at each iteration derivatives were calculated for the last included point

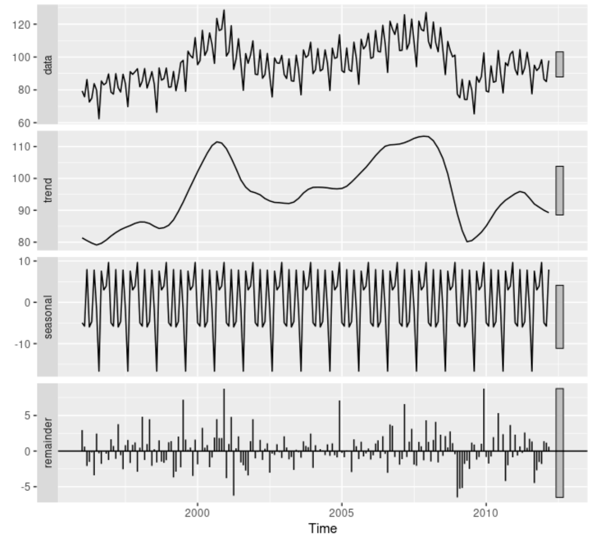

Figure 3.1: R. J. Hyndman, G. Athanasopoulos, The electrical equipment orders (top) and its three additive components obtained from a robust STL decomposition with flexible trend-cycle and fixed seasonality, OTexts <https://otexts.com/fpp2/stl.html>, accessed Jul. 2021

STL Decomposition is a versatile method for separating time series data into trend t_i, seasonal s_i and residual r_i components. It was first presented in [2], and is based on the hypothesis that at each $i-th$ instant the time series' value y_i is composed of three components in an additive relationship:

$$y_i = s_i + t_i + r_i$$

The STL Algorithm performs smoothing of the time series by using the LOESS [3] method in two cycles. The inner cycle operates on trend and seasonality components, while the outer one minimizes the effect of outliers by updating weights for the next run of the inner cycle. At each pass of the inner cycle a seasonal smoothing is performed first, updating the $i-th$ seasonal component s_i, followed by a trend smoothing updating the $i-th$ trend component t_i: remainder r_i is finally determined by subtracting seasonality and trend from input data.

$$r_i = y_i - s_i - t_i$$

Figure 3.1 shows an example of such method applied on electrical equipment orders data: as we can see the source data (in the first plot) presents a trend

[3]LOESS interpolation is a non-parametric technique using local weighted regression to fit a smooth curve through points in a scatter plot.

component causing values to fall into ascending and descending channels, and a seasonality component observable by the repeating patterns formed by its spikes. The second and third plots respectively show the trend and seasonality components, extracted from the souce data: by subtracting these from the input, the residual component (fourth plot) is left, representing the actual signal influencing changes in the source data.

Historical Prices

Historical price action has been included in each sample by considering a lagging sliding window with width

$$W = 10$$

This means each sample contains observations from the previous W samples, identified by the "_lagN" suffix, where N stands for the distance between the current observation and the lagged one.

This was done for both raw price data and percent variation candle features as well.

Candlestick Analysis

As stated in paragraph 2.2.1, candlestick data expresses more information than just price action.

To do so, we first defined candle size, candle body size, candle wick and candle shadow for each candle at timestamp t as follows:

$$candle\_size_t = high_t - low_t$$

$$candle\_body_t = close_t - open_t$$

$$candle\_wick_t = high_t - close_t$$

$$candle\_shadow_t = close_t - low_t$$

Then we applied the percent variation formula:

$$X_{pct\_t+1} = X_{t+1} - X_t / X_t$$

To both daily OHLCV data and OHLCV data resampled in 3, 7 and 30 day intervals, to obtain a broader view on the market state movements.

- "high_low_dist_pct" expresses the change in total candle size w.r.t the previous candle
- "close_open_pct" expresses the change in candle body w.r.t the previous candle

- "high_close_dist_pct" expresses the change in candle wick w.r.t the previous candle
- "low_close_dist_pct" expresses the change in candle shadow w.r.t the previous candle

Features calculated from resampled OHLCV data bear the "_dN" suffix, with N being the resample interval size.

Price Volatility

Volatility is a statistical measure indicating the dispersion of returns for a given asset, often measured as the standard deviation on returns from that same asset.

In most cases, the higher the volatility, the riskier the security, this has proven to be an efficient indicator for cryptocurrency price action.

For the purpose of this work, we calculated the 3, 7 and 30 days volatility by applying the standard deviation over a rolling mean of the corresponding period's return:

$$close\_volatility\_Td = std\_dev(rolling\_mean(percent\_change(close), T))$$

$$with\ T \ \in [3, 7, 30]$$

Candlestick Technical Patterns

Candlesticks in charts build patterns predicting price direction once completed, like technical indicators they can identify bearish or bullish trends, their use has been estabilished for centuries.

There are various candlestick patterns used to determine price direction and momentum, with the most various names including three line strike, two black gapping, three black crows, evening star, and abandoned baby.

Although pattern recognition's effectiveness is debated in this age of electronic trading, being the crypto market mostly composed of retail traders its use is justified by the fact that people would FOMO [4] in trades when seeing a bullish

[4] In the context of financial markets and trading, FOMO refers to the fear that a trader or investor feel by missing out on a potentially lucrative investment or trading opportunity.

pattern, and fear out when seeing a bearish one, contributing in expressing the market's sentiment.

The TA-Lib Python library provides a unified API for retrieval of such patterns, taking OHLC data as input and outputting a timeseries with corresponding index and values:

- -100 if a bearish pattern has been completed in the current sample
- 0 if no patterns have been completed in the current sample
- 100 if a bullish pattern has been completed in the current sample

All possible patterns have been searched for and collected in a dataframe, which has been further processed in order to summarize the results, producing two features:

- "talib_patterns_mean" containing the row-wise mean of each row
- "talib_patterns_sum" containing the row-wise sum of each row

3.2.2 Technical Analysis

Many studies on technical analysis indicators prove their correlation with price movements in the next future, therefore their predictive power.

For example, basing on the assumptions that:

- Prices follow repeating trend cycles
- Previous trends lead to similar future trends
- Any uptrend must be followed by a downtrend

Murphy [3] explains how the price of a security already contains all the information about anything that could possibly affect it.

Price data was thus augmented with a selection of technical indicators, which we will briefly discuss in this section, separately by type:

Trend Indicators

This type of indicator is also known as trend-following: their values help assess the direction and strength of a trend once it is established, but not predict it. Assessing the status of a trend can be helpful when handling multi-day positions, as a weakening bearish trend might indicate a good opportunity to enter a long trade and vice versa.

Moving averages (both simple and exponential) are the main instruments in this category[5], although their values are highly dependant on price action (and therefore present trend and seasonality components): for this reason, they are used as a means to calculate two derived indicators first used in [4], that overcome the issue by considering distances between averages instead of their values.

MACD and PPO are conceptually in the middle between an oscillator and a moving average: they are unbounded, and employ relative changes between moving averages to assess trend direction and strength, while generating overbought/oversold[6] signals when the indicator crosses its signal line.

AO translates trend recognition to a bounded oscillator in the range [-100, 100], indicating trend strength and direction, yet ignoring market conditions.

- Simple Moving Average or SMA is the average of prices in a defined time interval, with uniform weight. Traders often plot multiple moving averages with different time intervals on their charts: a shorter moving average crossing a longer one should be interpreted as a trend reversal signal.

- Exponential Moving Average or EMA similarly to SMA is the average of prices in a defined time interval, but with an exponential weight distribution granting more impact on most recent records.

- Moving Average Convergence Divergence or MACD is an indicator emphasizing the relationship between two moving averages of a security's price. It is calculated by subtracting the long EMA from the short EMA. A nine-day EMA of the MACD, called "signal line", is usually plotted on top of the MACD line, and employed as a trigger for buy and sell signals. Traders may buy when MACD crosses above its signal line and sell - or short - when MACD crosses below the signal line.

- Percent Price Oscillator or PPO is identical to the MACD indicator, except the PPO measures percentage difference between two EMAs, while the MACD measures absolute (dollar) difference.

- Aroon Oscillator or AO is a trend-following indicator that uses aspects of the Aroon Indicator (Aroon Up and Aroon Down) calculated on 14 periods to measure the strength of a current trend and the likelihood that it will continue,

[5]SMA and EMA act as support and resistances for price action: if price is above the MA then action is bullish with the MA as support, otherwise it is bearish and MA represents a resistance

[6]These are both unstable market conditions. An overbought condition indicates the security is overpriced, while an oversold condition indicates the security is under priced: in both cases, a correction to a more stable condition is expected.

assuming values in the range [-100, 100]. High oscillator values indicate an uptrend, while low oscillator values are an indication of downtrend.

SMA and EMA have been used to derive two indicators first introduced in [4]:

- Relative SMA is an indicator based on Simple Moving Averages representing the relative change between a longer and a shorter SMA: this allows tracking of crosses between moving averages while being independent of the security price.
$$RSMA_t(a,b) = \frac{SMA_t(b) - SMA_t(a)}{SMA_t(a)}$$
- Relative EMA is an indicator based on Exponential Moving Averages, representing the relative change between a longer and a shorter EMA: this allows tracking of crosses between moving averages while being independent of the security price.
$$REMA_t(a,b) = \frac{EMA_t(b) - EMA_t(a)}{EMA_t(a)}$$

Both RSMA and REMA have been calculated for the pairs (5, 20), (8, 15), (20, 50). Percent variation was additionally determined for the lower timeframes.

MACD and PPO were calculated on the pair (12, 26), deriving additional features representing indicator percent variation, signal line and difference between the indicator and its signal line.

AO was calculated on 14 periods, deriving an additional feature with a normalized representation in the range (0, 1).

Momentum Indicators

Momentum indicators are "leading" indicators, in the sense that they posses predictive power about the speed of price change. They are often bounded, with particularly defined points of interest which, when crossed, may be used to generate trading signals by identifying market conditions. Divergences between this class of indicators and price action are also of interest, as they indicate a weakening trend.

Included indicators have been picked because they are based on different principles: STOCH bases its action on historical prices, RSI is based on the speed at which price moves, MFI is similar to RSI but keeps volume into account. CMO and TSI are unbounded, thus relevance levels depend on the traded security, with the former being more sensitive than RSI and the latter being less accurate but more indicative of trend buildup.

- Stochastic or STOCH is a momentum oscillator comparing a particular closing price of a security to a range of its prices over a certain period of time. Being based on price history, sensitivity of the oscillator to market movements is reducible by adjusting the considered time period or taking a moving average of the result. It employs a 0-100 bounded range of values, and is used to generate overbought (for values > 80) and oversold (for values < 20) trading signals.

- RSI or Relative Strength Index is a widely used technical oscillator, with the objective of identifying oversold and overbought market conditions. It has a fixed [0, 100] range where values $<= 20$ indicate the asset is undervalued (oversold), while values > 80 indicate the asset is overvalued (overbought).
- MFI or Money Flow Index (MFI) is an oscillator that generates overbought or oversold signals using both prices and volume data. The oscillator moves between 0 and 100: a reading above 80 is considered overbought, while a reading below 20 is considered oversold, although levels of 90 and 10 are also used as thresholds. A divergence between the indicator and price is of interest: if the indicator is rising while price action is downwards or horizontal, price could start rising.
- CMO Chande momentum oscillator is a technical momentum assuming values in the range [-100, 100] and accounting for both up and down days without smoothing results, triggering more frequent oversold and overbought signals. Many technical traders add a 10-period moving average to this oscillator to act as a signal line, generating a bullish signal when it crosses above the moving average and a bearish one when it drops below the moving average.
- TSI The true strength index (TSI) is a price-based - thus unbounded - technical momentum oscillator used to identify trends and reversals. It may be useful for determining extreme market conditions, indicating potential trend direction changes via crossovers of the center line or signal line, and warning of trend weakness through divergence.

STOCH was calculated on a 14 periods interval. Due to high sensitivity of the indicator, a 3-periods rolling mean divided by 100 - commonly known as percent K - was also considered.

RSI, MFI and CMO were calculated on a 14 periods interval, all with an additional feature normalizing its values in range (0, 1). Additionally, the CMO signal line and the difference between the indicator and its signal line have been considered.

Volatility Indicators

Indicators in this class mostly belong to the "lagging" category, aiming to identify the range in which prices oscillate in a given day, rather than their direction. This is useful when used in conjunction with other indicators, in particular when trading volatile assets such as cryptocurrencies, as it allows the trader to determine possible reversal points.

ADX, WD and ATRP are originally designed by Welles Wilder as a part of a proprietary trading system described in [5]. These are unbounded, their interactions are used to generate trading signals.

FI is another volatility indicator designed by Alexander Elder [6], emphasizing the strength behind a price move by taking volumes into account, allowing the trader to identify reversal points.

- ADX or Average Directional Movement indicates trend strength, and is composed of two indicators: DI+ and DI- (Called positive and negative Directional Indicators, respectively) which are usually plotted together with the former, acting as signal lines. Crossovers of DI+ and DI- lines generate trading signals.
- WD Is the difference between DI+ and DI- indicators, it is used in conjunction with ADX to generate signals.
- ATRP or Average True Range Percent is a technical analysis indicator measuring changes in market volatility by decomposing the entire range of an asset price for that period. It is based on a series of "true range" values for an asset.
- FI or Force Index is a volatility indicator: strong trends - both up and down - should see the force index rise, while during sideways movement, the indicator will often fall because traded volume and/or size of each trade get smaller.

ADX, WD and ATRP were calculated on 14 days periods, while Force Index was calculated on 13 and 50 periods.

Volume Indicators

In this category are included both "lagging" and "leading" indicators, with the common objective of using averaging or smoothing of raw volume data to measure the strength of a trend - and confirm its direction. If the market scenario indicates a move, but this is not supported by volume, the move is invalidated.

All of the indicators included are unbounded, in particular OBV was first introduced by Joseph Granville [7] behind the belief that huge movements in volume act as a "compressed spring", pushing price movement.

- PVO or Percentage Volume Oscillator (PVO) measures volume surges by comparing a shorter and a longer moving average, without taking price into account. Similarly to MACD, it compares fast and slow volume moving averages by showing how short-term volume differs from the average volume over longer-term.

- ADI or Accumulation/DIstribution is a cumulative indicator using volume and price to assess whether a stock is being accumulated or distributed, identifying divergences between price and volume flow. This provides insight into how strong a trend is: If price is rising but the indicator is falling, buying or accumulation volume may not be enough to support the price rise, anticipating a price decline.

- OBV or On-Balance volume is an indicator using volume flow to predict changes in stock price. The actual individual quantitative value of this indicator is not relevant: its value is cumulative, and solely depends on the start date. Traders usually look at the nature of OBV movements over time, with the slope of the OBV line carrying all the weight of the analysis.

PVO was calculated considering the 12 and 26 periods averages, ADI and OBV do not require parameters.

3.2.3 Blockchain Data

The Blockchain technology allows building of distributed ledgers where all transaction data is publicly available, including wallet balances and coin exchanges. For example, retail interest can be measured in the form of coins held in non-custodial wallets [7], and given that for some wallets - such as exchanges - owners are well-known, asset flows can be analyzed.

Percent variation was applied to all the features included in the downloaded data, both original and resulting features were added to the final dataset, leaving the task of selecting which features worked best to the feature selection step, which will be discussed later.

For the sake of this study, included features were grouped in the following categories:

Supply metrics (Table 3.2) aim to explain token supply and its distribution among wallets. While issuance information may be helpful in determining token demand on the market, insights about wallets holding certain amounts of tokens or equity values may help assess sentiment regarding the currency: a project well supported by its community would have smaller investors buying tokens and moving them to their own custodial wallet, aiming to hold for the long run.

[7] Wallets in which the owner is the owner of the private key, opposite to custodial wallet such as exchanges' wallets where keys are held by the exchange.

Addresses metrics (Table 3.4) are an index of network activity and interest: the same wallet (private key) can have unlimited public addresses linking to the same balance. Due to the nature of Blockchain technology, all the transfers and wallet balances are publicly accessible while keeping owner information private: features in this category represent an edge over stocks trading, as they might be used to monitor off-exchange activity and anticipate price moves: for example a whale moving large amounts of tokens from a cold wallet to an hot wallet is usually interpreted as a bearish signal.

Mining metrics (Table 3.5) represent protocol-specific parameters: since PoW networks are secured by miners, hashrate could be an index of how secure the network is, with difficulty being usually periodically adjusted in its function through protocol-mandated schemes. Miner activity is incentivized by a certain revenue they receive both in minted tokens and fees from verified transactions: when difficulty is too high miner revenue decreases, so miners usually reduce energy consumption - and costs - by throttling mining power, impacting on the hashrate. Changes in hashrate usually reflect on token's exchange price.

Transactions metrics (Table 3.6) address transferred value and throughput of the network: an efficient network - thus preferable for a transfer - should be able to handle high throughput while keeping fees as low as possible. While smaller transfers of value are index of diffused day-to-day adoption, larger ones could indicate forthcoming speculative activity.

Network Usage metrics (Table 3.7) features in this class cover blockchain activity in the form of mined block and their size (or weight for segwit networks). Block count is the way blockchains measure time, as the time it takes for a block to be produced is regulated by the protocol. Protocol-mandated events such as halvings or hard forks are usually based on reaching a certain block count, therefore these features could help the model identify patterns around such events.

Fees and Revenue metrics (Table 3.8) these metrics cover the network's efficiency in terms of transfer costs, representing fees for doing operations on the blockchain such as transactions and smart contract execution. Fees are part of the reward miner receive for their work, impacting transaction times as miners could reject transactions with lower fees. Gas is the fuel used to power smart contracts: its values could indicate interest in the dAPPS published on the network, thus on the network itself.

Exchange metrics (Table 3.9) represent the currency flow for known centralized exchange addresses, for both deposit and withdrawals. This can be useful in generating signals as exchanges are where trades take place, therefore an outgoing flow of value indicates trust in the asset (for example investors moving coins to cold wallets for holding), while an in-going flow of value might anticipate a downwards move as it would mean investors readying tokens for selling.

Market metrics (Table 3.10) cover the economic aspects of cryptocurrency markets such as capitalization, BTC exchange price, ROI and volatility returns. In general this class of features adds information about market liquidity, also considering other investors' activity (and their profits): this could be helpful as a high realized value could trigger investors taking profits over their positions, affecting price action.

Economics metrics (Table 3.11) include the ratio of the USD network value (or market capitalization, current supply - both in float and free-float) divided by the adjusted transfer value (in USD). Also referred to as NVT, this class of features could be seen as an indicator for overbought/oversold market conditions: an high NVT value means the network is overvalued in relation to the value it is able to transfer (in terms of volume).

Picks features have been developed by combining existing features, and placed in their own category. Such features are:

- earned_vs_transacted representing miners earnings versus transfered value in the time interval, aiming to spot liquidity injections in the market:

$$earned\_vs\_transacted = \frac{isstotntv + feetotntv}{txtfrvaladjntv}$$

- isstot1_isstot365_pct representing changes in the ratio between newly issued tokens and tokens issued in an expanding window W of at least one week and up to 1 year:

$$total\_mined = \sum_{i=0}^{W} isstotntv_{t-i}$$

$$isstot1\_isstot365\_pct = pct\_change(\frac{isstotntv}{total\_mined})$$

- splycur_isstot1_pct representing changes in the ratio between newly issued tokens and current supply:

$$splycur\_isstot1\_pct = pct\_change(\frac{isstotntv}{splycur})$$

Feature Name	Aggregation	Unit	Description
isscontntv	SUM	NTV	Sum of new native units issued that interval.
isscontpctann	PERCENTAGE	N/A	Percentage of new native units (continuous) issued over that interval, extrapolated to one year (i.e., multiplied by 365), and divided by the current supply at the end of that interval. Also referred to as the annual inflation rate.
isscontpctday	PERCENTAGE	N/A	Percentage of new native units (continuous) issued over that interval divided by the current supply at the end of that interval. Also referred to as the daily inflation rate.
isscontusd	SUM	USD	Sum USD value of new native units issued that interval.
isstotntv	SUM	NTV	Sum of all new native units issued that interval.
isstotusd	SUM	NTV	Sum USD value of all new native units issued that interval.
ndf	RATIO	N/A	Ratio of supply held by addresses with at least one ten-thousandth of the current supply of native units to the current supply.
splyact{T}	SUM	N/A	Sum of unique native units that transacted at least once in the trailing T periods up to the interval.
splyactever	SUM	N/A	Sum of unique native units held by accounts that transacted at least once up to that interval.

Table 3.2: CoinMetrics' blockchain features belonging to the Supply category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Feature Name	Aggregation	Unit	Description
splyactpct1yr	PERCENTAGE	N/A	Percentage of the current supply that has been active in the trailing 1 year up to the end of that interval.
splyadrbal1in{N}	SUM	N/A	Sum of all native units being held in addresses whose balance was at least 1/N of the current supply of native units as the end of that interval.
splyadrbalntv{N}	SUM	N/A	The sum of all native units being held in addresses whose balance was N native units or greater at the end of the interval.
splyadrbalusd{N}	SUM	N/A	The sum of all native units being held in addresses whose balance was USD N or greater at the end of the interval.
splycur	SUM	NTV	Sum of all native units ever created and visible on the ledger (i.e., issued) at the end of that interval.
splyexpfut10yr	SUM	NTV	Sum of all native units counting current supply and those expected to be issued over the next 10 years if the current issuance schedule is followed. Future expected hard-forks are not considered until the day they are activated/enforced.
splyff	SUM	NTV	Number of native units readily available to trade in open markets at the end of the time interval.

Table 3.3: CoinMetrics' blockchain features belonging to the Supply category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Feature Name	Aggregation	Unit	Description
adractcnt	SUM	N/A	Sum count of unique addresses active in the network either as a recipient or originator of a transaction.
adrbal1in{N}cnt	SUM	N/A	Sum count of unique addresses holding at least 1/N fraction of the total supply of native units.
adrbalcnt	SUM	N/A	Sum count of unique addresses holding any amount of native units.
adrbalntv{N}cnt	SUM	N/A	Sum count of unique addresses holding at least N native units.
adrbalusd{N}cnt	SUM	N/A	Sum count of unique addresses holding an amount of native units corresponding to N USD.

Table 3.4: CoinMetrics' blockchain features belonging to the Addresses category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Feature Name	Aggregation	Unit	Description
difflast	N/A	N/A	The difficulty [8] of the last block in the interval.
diffmean	N/A	N/A	Mean difficulty of finding a hash that meets the protocol-designated requirement in the interval.
hashrate	MEAN	Varies	Mean rate at which miners are solving hashes that interval. Hash rate is the speed at which computations are being completed across all miners in the network. The unit of measurement varies depending on the protocol.
hashrate30d	MEAN	Varies	Mean rate at which miners are solving hashes over the last 30 days. The unit of measurement varies depending on the protocol.
revhashntv	MEAN	NTV	Mean miner reward per estimated hash unit performed during the period, in native units. The unit of hashpower measurement depends on the protocol.
revhashusd	MEAN	USD	USD value of the mean miner reward per estimated hash unit performed during the period, also known as hashprice.
revhashratentv	MEAN	NTV	Mean daily miner reward per estimated hash unit per second performed during the period, in native units.
revhashrateusd	MEAN	USD	USD value of the mean daily miner reward per estimated hash unit per second performed during the period, also known as hashprice.

Table 3.5: CoinMetrics' blockchain features belonging to the Mining category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Feature Name	Aggregation	Unit	Description
txcnt	SUM	TX	The sum count of transactions [9] that interval.
txcntsec	SUM	TX/s	The sum count of transactions divided by the number of seconds that interval.
txtfrcnt	SUM	TFR	The sum count of transfers [10] that interval. Only transfers that are the result of a transaction and that have a positive (non-zero) value are counted.
txtfradjntv	SUM	NTV	The sum of native units transferred between distinct addresses that interval removing noise and certain artifacts.
txtfradjusd	SUM	USD	The USD value of the sum of native units transferred between distinct addresses that interval removing noise and certain artifacts.
txtfrvalmeanntv	MEAN	NTV	Sum value of native units transferred divided by the count of transfers between distinct addresses that interval.
txtfrvalmeanusd	MEAN	USD	Sum USD value of native units transferred divided by the count of transfers between distinct addresses that interval.
txtfrvalmedntv	MEDIAN	NTV	Median count of native units transferred per transfer between distinct addresses that interval.
txtfrvalmedusd	MEDIAN	USD	The median USD value transferred per transfer between distinct addresses that interval.
velcur1yr	RATIO	N/A	Ratio of the value transferred (i.e., the aggregate size of all transfers) in the trailing 1 year divided by the current supply up to the end of that interval.

Table 3.6: CoinMetrics' blockchain features belonging to the Transactions category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Feature Name	Aggregation	Unit	Description
blkcnt	SUM	N/A	Sum count of blocks included in the main chain in the time interval.
blksizemeanbyte	MEAN	Bytes	Mean size (in bytes) of blocks created in the time interval.
blkwghtmean	MEAN	Weight	Mean weight [11] of blocks created in the time interval.
blkwghttot	SUM	N/A	Sum of the weights of blocks created in the time interval.

Table 3.7: CoinMetrics' blockchain features belonging to the Network Usage category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Feature Name	Aggregation	Unit	Description
feebytemeanntv	MEAN	NTV	Mean transaction fee per byte of all blocks that interval in native units.
feemeanntv	MEAN	NTV	Mean fee per transaction in native units in the interval.
feemeanusd	MEAN	USD	USD value of the mean fee per transaction that interval.
feemedntv	MEDIAN	NTV	Median fee per transaction in native units that interval.
feemedusd	MEDIAN	USD	USD value of the median fee per transaction that interval.
feetotntv	SUM	NTV	Sum of all fees paid to miners that interval.
feetotusd	SUM	USD	Sum USD value of all fees paid to miners that interval.
gaslmtblk	SUM	GAS	Sum gas limit of all blocks that interval.
gaslmtblkmean	MEAN	GAS	Mean gas limit per block that interval.
gaslmttx	SUM	GAS	Sum gas limit of all transactions that interval.
gaslmttxmean	MEAN	GAS	Mean gas limit per transaction that interval.
gasusedtx	SUM	GAS	Total gas used across all transactions that interval.
gasusedtxmean	MEAN	GAS	Mean gas used per transaction that interval.
revalltimeusd	SUM	USD	Sum USD value of all miner revenue (fees plus issued tokens) from all time up to the end of the interval.
revntv	SUM	NTV	Sum native units of all miner revenue (fees plus issued tokens) in the interval.
revusd	SUM	USD	Sum value of all miner revenue (fees plus issued tokens) in the interval, in USD.

Table 3.8: CoinMetrics' blockchain features belonging to the Fees and Revenue category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Feature Name	Aggregation	Unit	Description
flowinexntv	SUM	NTV	Sum number of native units sent to exchanges that interval, excluding exchange to exchange activity.
flowinexusd	SUM	USD	Sum USD value sent to exchanges that interval, excluding exchange to exchange activity.
flowoutexntv	SUM	NTV	Sum number of native units withdrawn from exchanges that interval, excluding exchange to exchange activity.
flowoutexusd	SUM	USD	Sum USD value withdrawn from exchanges that interval, excluding exchange to exchange activity.
flowtfrfromexcnt	SUM	N/A	Sum count of transfers from any address belonging to an exchange in that interval, excluding exchange to exchange activity.

Table 3.9: CoinMetrics' blockchain features belonging to the Exchange category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021.

Feature Name	Aggregation	Unit	Description
capact1yrusd	PRODUCT	USD	Sum USD value of all active native units in the previous year.
capmvrvcur	RATIO	N/A	Ratio of the sum USD value of the current supply to the sum realized USD value of the current supply.
capmvrvff	RATIO	N/A	Ratio of the free float market capitalization to the sum realized USD value of the current supply.
capmrktcurusd	PRODUCT	USD	Sum USD value of the current supply. Also referred to as network value or market capitalization.
capmrktffusd	PRODUCT	USD	Sum USD value of the current free float supply. Also referred to as free float market capitalization.
caprealusd	PRODUCT	USD	Sum USD value based on the USD closing price on the day that a native unit last moved (i.e., last transacted) for all native units.
pricebtc	N/A	BTC	Fixed closing price of the asset for the BTC pair, as reported by CoinMetrics' reference rate service. Not valid for BTC.
priceusd	N/A	USD	Fixed closing price of the asset for the USD pair, as reported by CoinMetrics' reference rate service.
roi1yr	PERCENTAGE	N/A	Return on investment for the asset assuming a purchase 12 months prior.
roi30d	PERCENTAGE	N/A	Return on investment for the asset assuming a purchase 30 days prior.
vtydayret180d	RATIO	N/A	180D volatility, measured as the standard deviation of the natural log of daily returns over the past 180 days.
vtydayret30d	RATIO	N/A	30D volatility, measured as the standard deviation of the natural log of daily returns over the past 30 days.

Table 3.10: CoinMetrics' blockchain features belonging to the Market category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Feature Name	Aggregation	Unit	Description
nvtadj	RATIO	N/A	Ratio of the network value (or market capitalization, current supply) divided by the adjusted transfer value. Also referred to as NVT.
nvtadj90	RATIO	N/A	Ratio of the network value (or market capitalization, current supply) to the 90-day moving average of the adjusted transfer value. Also referred to as NVT.
nvtadjff	RATIO	N/A	Ratio of the free float network value (or market capitalization, current supply) divided by the adjusted transfer value. Also referred to as NVT.
nvtadjff90	RATIO	N/A	Ratio of the free float network value (or market capitalization, current supply) to the 90-day moving average of the adjusted transfer value. Also referred to as NVT.

Table 3.11: CoinMetrics' blockchain features belonging to the Economics category. CoinMetrics documentation <https://tools.coinmetrics.io>, accessed Jun. 2021

Chapter 4

Method

This section will cover the steps required for building, testing and analysing the models used for generating signals in the proposed trading systems and their benchmarks on the proposed trading agent.

The first step is defining the classification problem and target labels for input data. Once samples for each dataset are labeled, all of the available data for each currency is split in training and testing sets, with a 70:30 ratio.

The training set is used for feature selection and hyperparameter optimization of the models, while the testing set is used for making predictions and validating the model's performance by sequentially training a new model each day with a sliding window approach.

Finally predictions are used for both testing the resulting trading system by means of a custom back-testing agent, and explaining model behavior by means of SHAP value analysis.

4.1 Problem Statement

Generating a trading signal involves finding a function mapping each input vector to a discrete set of classes. As the aim of this work is using supervised machine learning models as the signal generation part of the trading strategies, we need to label our data prior to fitting models. The way to do so is identifying classes by the expected (ie. next day) price variation based on close price, defined as follows:

$$close_{pct} = \frac{close_{t+1} - close_t}{close_t}$$

- SELL for a price decrease above 1%

$$close_{pct} < -1\%$$

- HOLD for a stationary price action

$$-1\% < close_{pct} < 1\%$$

- BUY for a price increase above 1%

$$close_{pct} > 1\%$$

This models a predictive classification problem, whose input data is real-world timeseries data coming from both technical and fundamental analysis of cryptocurrency markets, with naturally imbalanced labels.

4.2 Feature Selection

Feature selection is the process of reducing the number of input variables when developing a predictive model: this is desirable to both reduce the computational cost of model training and improve model performance.

The process can be performed by various methods with advantages and disadvantages, that can be grouped in three main categories:

- Filter methods select variables regardless of the model, they are based on statistical methods such as features linear dependency reduction (unsupervised, meaning the method works on unlabeled data), or correlation with the target value (such as ANOVA f-score). These methods are particularly effective in computation and robust to overfitting, but tend to select redundant variables when relationships are not considered.

- Wrapper methods evaluate subsets of values by iteratively training a model. The main drawbacks are high computation time for large number of variables as the number of subsets is proportional to the number of input features, with an increasing risk of overfitting when the number of observation is insufficient.

- Embedded methods combine the advantages of the two previous methods, exploiting a trained model's embedded coefficients (such as a RandomForest's feature importance vector or SVM coefficients) to perform feature selection by picking features whose importance is above a certain threshold value. The main advantage of this method is that it allows explanation of the results by looking at the feature importances vector, while being computationally cheaper than a wrapper method and more accurate than a filter method.

When dealing with Cryptocurrency data, literature shows an heterogeneous search space: other than market data, other kinds of features are often included such as on-chain metrics and network transactions as well as cross-domain aspects which are not apparently related to cryptocurrencies such as macro economics, community, developer and social activity.

The aim of this section is to take a step back and analyse contributes from each feature domain to the final model, focusing on both market and fundamental aspects as expressed by blockchain information.

This was achieved by employing an embedded method leveraging feature importances as expressed by an XGBoost model [8] based on the whole training set. Features were selected by picking those whose importance was above the mean value for the specific dataset, and further analysed and compared across currencies seeking for relevant information.

4.2.1 Feature Hierarchy

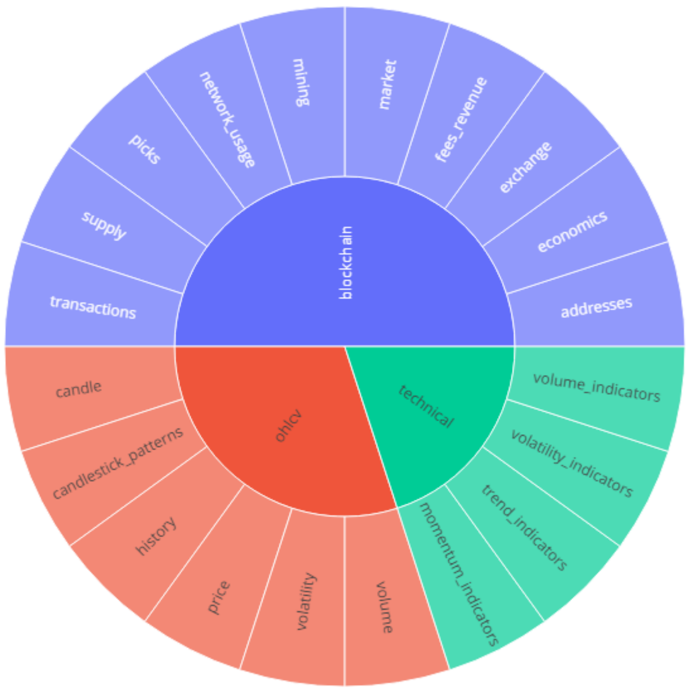

Figure 4.1: Sunburst graph representing the adopted feature hierarchy for feature selection

Due to the large number of input features and their non-uniformity across datasets, they were grouped in a three-level semantical hierarchy as represented in figure 4.1 to promote comparability across the analysed trading pairs, allowing a grasp to the common features influencing the cryptocurrency markets as a whole by analysing an overall mean of importances.

4.3 Hyperparameters Optimization

Machine learning models have various hyperparameters used to tune their performances. Finding optimal hyperparameters is a crucial step to achieve generalization and optimize model performance with regard to the chosen metric.

Choice of the target evaluation metric is crucial, as it depends from both the problem and the context where the resulting model will be used. For the sake of this work relevant classes are the ones triggering orders (ie. SELL and BUY): a false positive would involve opening a position - and therefore paying fees, while a false negative would imply a missed possibly profitable trade, but no costs. Hence, the precision metric was chosen as it fits our problem best, even though it does not take class imbalance in the training data into account: our chosen testing strategy - which will be discussed later in this chapter - introduces a certain bias to the models, deriving from price trends in the previous W days. Bias in a classification model would normally be undesirable, but in our case it helps moderating signals in direction of the actual market trend, which was considered an advantage.

4.3.1 Cross Validation

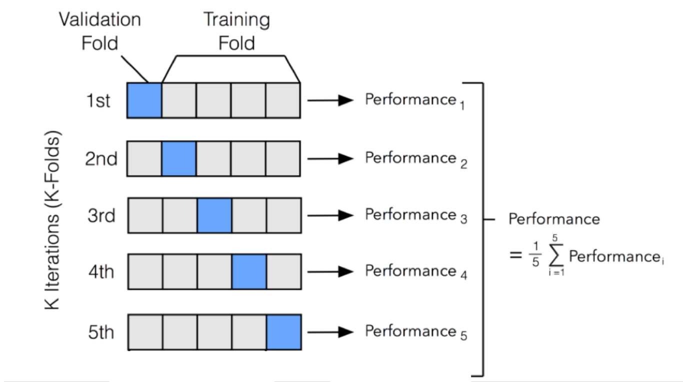

Figure 4.2: Yash Khandelwal, 2021, K-Fold Cross Validation, Kaggle <https://www.kaggle.com/discussion/204878>, accessed Jun. 2021

Cross validation is a statistical technique in which a given dataset is split into a certain number of "folds", as shown in figure 4.2. Iteratively, one fold is excluded and taken as test set while the rest of the data is used for training a model. Chosen results are evaluated for each iteration, then finally combined by a balanced mean, promoting model generalization and reducing the risk of overfitting.

Usually, folds contain random data. Since the data used in this work was "natural" and time-dependant (timeseries), each folds was representing a different class distribution than the original training data, resulting in inconsistent results. This has been addressed by implementing Cross Validation through a stratification process, shuffling the data in order to mantain the original class distribution ratios for each fold. Additionally, since algorithms such as k-NN require a certain minimum number of samples from each class to work correctly, and this condition was not mean in some data sets due to differences in number of samples, the number of folds k was chosen between 3 and 5 ensuring each fold holds at least 40 samples.

4.3.2 Grid Search

Grid search is a technique attempting to compute the optimal model hyperparameters by exhaustively testing all the possible combinations taken from a given parameter grid. As a model is trained and tested for each possible combination, when compared to other tuning methods such as Random Search or Bayesian Optimization grid search is costly both in terms of time and computing power, but it yields the best possible results when the right parameter grids are defined.

This technique was combined with cross validation using a stratifying splitter, with the drawback of training k multiple models but the advantage of achieving better generalization and parameter stability. Ranges for parameters used in grid for some of the best performing algorithms (in terms of trading results) are represented in tables 4.1, 4.2 and 4.3.

Parameter	Min	Max
n_estimators	100	500
colsample_bytree	0.8	1.0
colsample_bylevel	0.6	1.0
colsample_bynode	0.6	1.0
max_depth	2	6

Table 4.1: Ranges adopted in XGBoost Parameter grid for grid search

Parameter	Min	Max
hidden_layer_sizes	(2, 4)	(100,100)

Table 4.2: Ranges adopted in MLP Parameter grid for grid search

Parameter	Min	Max
n_estimators	100	500
max_depth	2	4
min_samples_leaf	0.05	1

Table 4.3: Ranges adopted in RandomForest Parameter grid for grid search

4.4 Model testing

This step involved using selected feature subsets for each datasets and hyperparameters for each pair of dataset and ML algorithm resulting from the previous steps to build the final models used for generating signals in the proposed trading systems.

This was achieved by building a sequence of models in a sliding window fashion, testing different window lengths $w \in [90, 180, 240]$. This means that for each exchange pair, at each simulation day t a new model is built and trained on the w samples preceding that day, with the first model being trained on the last w samples from the training set, then using for assigning the $t-th$ sample a predicted label as follows:

$$model_t = model.fit(dataset[t-w:t-1],\ labels[t-w:t-1])$$

$$pred_t = model_t.predict(data[t])$$

This process gave us a series of predictions *pred*, which was used in combination with *labels* to assess model performance by means of various metrics, including precision/recall and index of balanced accuracy [9] scores both for each class and in macro averages, calculated on the whole testing set.

4.5 Trading Simulation

The same predictions from 4.4 were used to assess the trading system's profitability by means of a custom developed trading agent, emulating trading on a real-world exchange by implementing its rules, fees and limitations and handling multi-day spot and margin positions (described in 2.2).

Orders were placed with both stop loss and take profit conditionals, meaning positions are automatically closed (with some caveats, which will be described in the following paragraph) when reaching certain price levels.

Each pair is initially allocated a 10000 USD equity and an equal value in liquidity pool allowance, split between fiat and collateral equivalent at the first available exchange price. For example, if we suppose a pair with an initial exchange price of USD 10000, USD 5000 are allocated in FIAT pool and 0.5 tokens are allocated in the collateral pool.

Caveats

Since simulation of a real world exchange involves taking into account multiple factors, some precautions had to be taken in order to simplify the work:

- As markets are solely driven by demand and offer, a big order would impact the order book, thus price. We act under the hypothesis that orders placed by the system do not affect the market in such way.
- Due to the use of daily data, it is not possible to determine what comes first between *low* and *high* prices in each day's candle. In order to correctly simulate conditional orders we consider the worst case scenario. In other words *low* values are always supposed to happen before *high* values, thus stop loss conditionals are evaluated before take profit conditionals.
- In real-world orders it may happen that a stop loss conditional is triggered, but filled at a different price than the trigger level due to high volatility, in an event called "slippage". In our simulation, we suppose conditional closes are filled at trigger price.
- Exchanges providing margin trading services offer each user a share of their own liquidity pools to place orders against, usually depending on the user's funding amounts and traded volumes. This means there is a hard cap about the amount of fiat and currency that can be borrowed for margin orders on each pair. This issue was addressed by supposing each pair is traded on an account who is assigned the median tier, supposing a separate FIAT liquidity pool for each pair.
- Trading fees are applied on every spot order, and both at opening and closing time for margin orders. Such fees depend on various factors such as whether the order is maker[1] or taker[2], as well as traded volumes in past times. For

[1] Maker (or market maker) orders are those who cannot get matched with other existing open orders in the book, thus adding liquidity to the order book

[2] Taker orders are those who are immediately matched with existing open orders in the order book, thus removing liquidity

the sake of the simulation, we supposed our agent always pays the highest possible fees.

- Due to network lag, it may happen an order is executed at a different price than the limit imposed when it was placed. Since the difference is negligible, we supposed all orders placed were immediately filled at the specified price.

- Since we are day-trading, we suppose the time between collection and processing of a new sample and the generation of a trading signal is negligible with respect to the trading period, thus all orders are immediately placed at the current day's closing price.

- Fiat liquidity pool is not shared across currencies, meaning even though they are traded on the same account, orders for each pair are placed against a different pool with a different allowance.

4.5.1 Order Placement

Positions are opened by placing either a long-selling or short-selling margin order against the exchange's liquidity pools, given that some conditions are met. Position sizing is determined by a fixed fractional strategy, investing 10% of the equity value at the time of order placement. Orders are placed with initial 10% stop loss and 5% take profit conditionals, meaning the order is closed at trigger price if the next *low* price goes below the stop loss value, or the next *high* price goes above the take profit value.

Long-selling orders

- Margin wallet must contain at least the position's fiat value plus fixed fee for opening a long-selling position

- Total fiat value in open long-selling positions must not exceed the fiat pool allowance

Short-selling orders

- Margin wallet must contain the amount of fiat value needed to buy the invested collateral amount plus the fixed fee for opening a short-selling position on the spot market (including spot orders' fixed fee)

- Total collateral value in open short-selling positions must not exceed the collateral pool allowance

4.5.2 Order Handling

Before placing any order, existing open orders are handled according to specific rules, in order:

Long-selling Handling

- If low price goes below stop loss price, position is closed at stop loss price.
- If high price goes above take profit price, position is closed at take profit price.
- If predicted signal is SELL, position is closed at the current close price.
- If predicted signal is HOLD and position is older than 3 days, it is closed at close price.
- If predicted signal is BUY and close price has increased with regard to the previous day, stop loss is adjusted to 5% below current close price.

Short-selling Handling

- If high price goes above stop loss price, position is closed at stop loss price.
- If low price goes below take profit price, position is closed at take profit price.
- If predicted signal is SELL and price has decreased with regards to the previous day, stop loss is adjusted to 5% above current close price.
- If predicted signal is HOLD and position is older than 3 days, it is closed at close price.
- If predicted signal is BUY, position is closed at current close price.

4.6 SHAP Analysis

SHAP (SHapley Additive exPlanations) is an approach aiming to explain outputs of any machine learning model [10] based on Shapley values, a concept from cooperative game theory [11]. It uses the shapley-value based method to offer local explanations about the cause of individual predictions, as well as global explanations based on the addition of Shapley values from single predictions.

As shown in equation 4.1 the approach uses the coalitions concept to compute shapley values of features by employing the black-box model f for the prediction of input x. Shapley value ϕ_j^m is the average marginal contribution of feature j in all possible coalitions. Marginal contributions are calculated as in 4.2, where $\hat{f}(x_{+j}^m)$

and $\hat{f}(x_{-j}^m)$ are the black-box model outputs with and without the $j-th$ feature of instance x.

$$\phi_j(x) = \frac{1}{M}\sum_{m=1}^{M}\phi_j^m \tag{4.1}$$

$$\phi_j^m = \hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m) \tag{4.2}$$

The SHAP library offers different kinds of explainers for calculation or approximation of shap values both generic - which can be used on any model - such as the KernelExplainer, and specialized - such as DeepExplainer or TreeExplainer [12] - leveraging model's intrinsic characteristics (such as trees' splitting points) to approximate calculations, improving performance. In this work we chose to use the TreeExplainer on models based on the XGBoost algorithm, because it is nowadays one of the most popular and widely-used ensemble methods and, as we will see in the results section, one of the best performing algorithms in terms of trading results.

Chapter 5

Experimental Results

5.1 Feature Selection

Feature importances resulting from feature selection were studied to assess relevance of included data on the cryptocurrency markets, both as a whole and per-currency. In this section we will analyse both aggregated results and the most relevant per-currency results, in order to shine a light on the features characterising each with respect to its protocol and purpose. Graphs in this section represent feature importances from the feature selection step, grouped according to the hierarchy defined in 4.2.1. The main chart is a sunburst chart or radial tree map, a visualization typically used for representing hierarchical data: it is composed of concentrical rings each corresponding to a level in the hierarchy, with the outer rings corresponding to leaf nodes and summing up to the value of their parent node/section. Bar charts have been added to improve chart readability, offering an alternative representation of category/subcategory importances, easing comparisons.

Figure 5.1 represents mean feature importance across all studied currencies. These results show how blockchain-derived metrics and price action are overall more relevant than technical indicators. The most relevant subcategories of features among blockchain metrics are supply, addresses and market metrics. Surprisingly, mining and transaction-related features are not represented: this could be due to the fact that many of the altcoins are either pre-mined or have adopted different consensus schemes such as Proof-of-Stake, allowing to keep the network secure without the need for mining, reducing energy consumption and environmental impact. On a deeper level, we can see how supply distribution-related features represent an edge over traditional markets because they allow a grasp to asset's distribution among investors: as expected, the number of addresses holding an equity value of USD 1000 is the most relevant feature in the supply category,

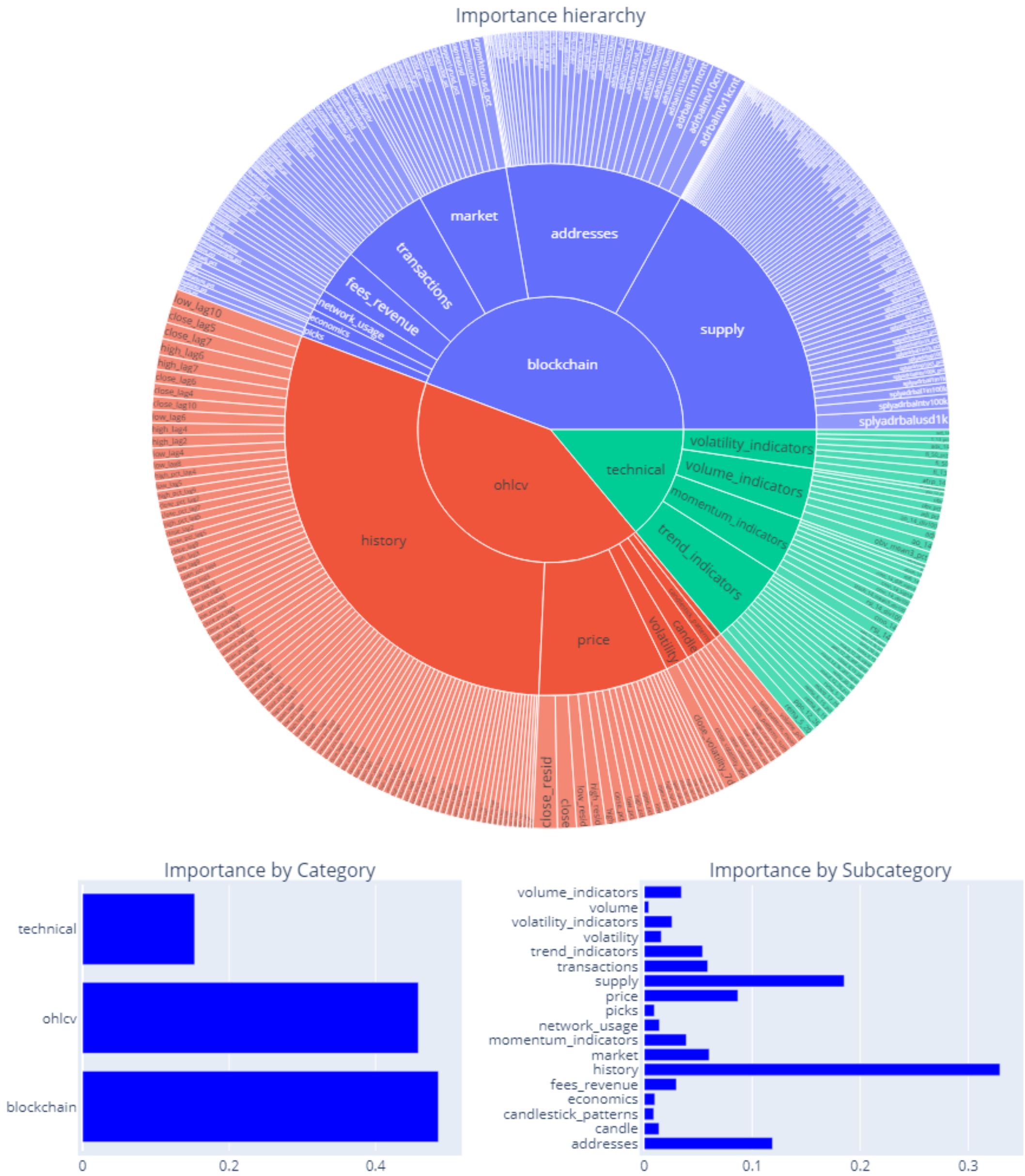

Figure 5.1: Graphs representing hierarchical mean feature importance across all analysed currencies

confirming how retail investors' trust is an important factor in a cryptocurrency's estabilishment.

Regarding OHLCV data, historical prices cover the most part of the category's importance, followed by price action and volatility features. In particular, features resulting from STL decomposition of price action are among the most important, proving how this decomposition method is more efficient in timeseries de-trending and de-seasonalization than other methods such as percent variation or splines derivatives. Technical indicators serve as a completion to market data: their importance is often higher than the corresponding subcategory of features in the other category. For example, the models seems to give more relevance to volume and volatility indicators than the corresponding features in the OHLCV category.

When considering currencies separately, we can see how importance distribution changes according to the characteristics of each different cryptocurrency, both in terms of protocol and offered service. In proof-of-work currency protocols (such as Bitcoin - Fig. 5.2, Bitcoin Cash - Fig. 5.3, Litecoin - Figure 5.4, or Ethereum - Figure 5.5) mining is a relevant feature category among the blockchain category - seemingly proportional to acknowledgment of the currency by miners themselves. Protocols offering smart contract platforms give transactions and network efficiency-related features more relevance, as they affect smart contract execution costs: we can see this in results for ETH, ADA (Figure 5.6) and BNB (Figure 5.7).

5.2 Model performance

Tables 5.1 and 5.2 report per-class and macro average precision (AVG) as well as index balanced accuracy (IBA) for the top-3 $algorithm : window$ pairs from each currency. While per-class precision is useful to assess performance of the classifier for that specific class, since precision does not take class balance into account, joint evaluation of AVG and IBA allows for a determination of the amount of bias introduced into the model: as our problem is a 3-class classification, the expected skill threshold for deeming results non-aleatory is 0.33. This means that:

- In cases where $AVG > 0.33$ and $IBA > 0.33$ the precision score is due to skill learned by the model.
- In cases where $AVG < 0.33$ and $IBA < 0.33$ the model did not acquire any skill.
- In cases where $AVG > 0.33$ and $IBA < 0.33$ the precision score is probably due to bias induced by class imbalance in the training window, and not skill learned by the model.
- Cases where $AVG < 0.33$ and $IBA > 0.33$ are to be considered anomalies, as the low precision would mean the classifier did not learn any skill, therefore the high IBA value would be unjustified.

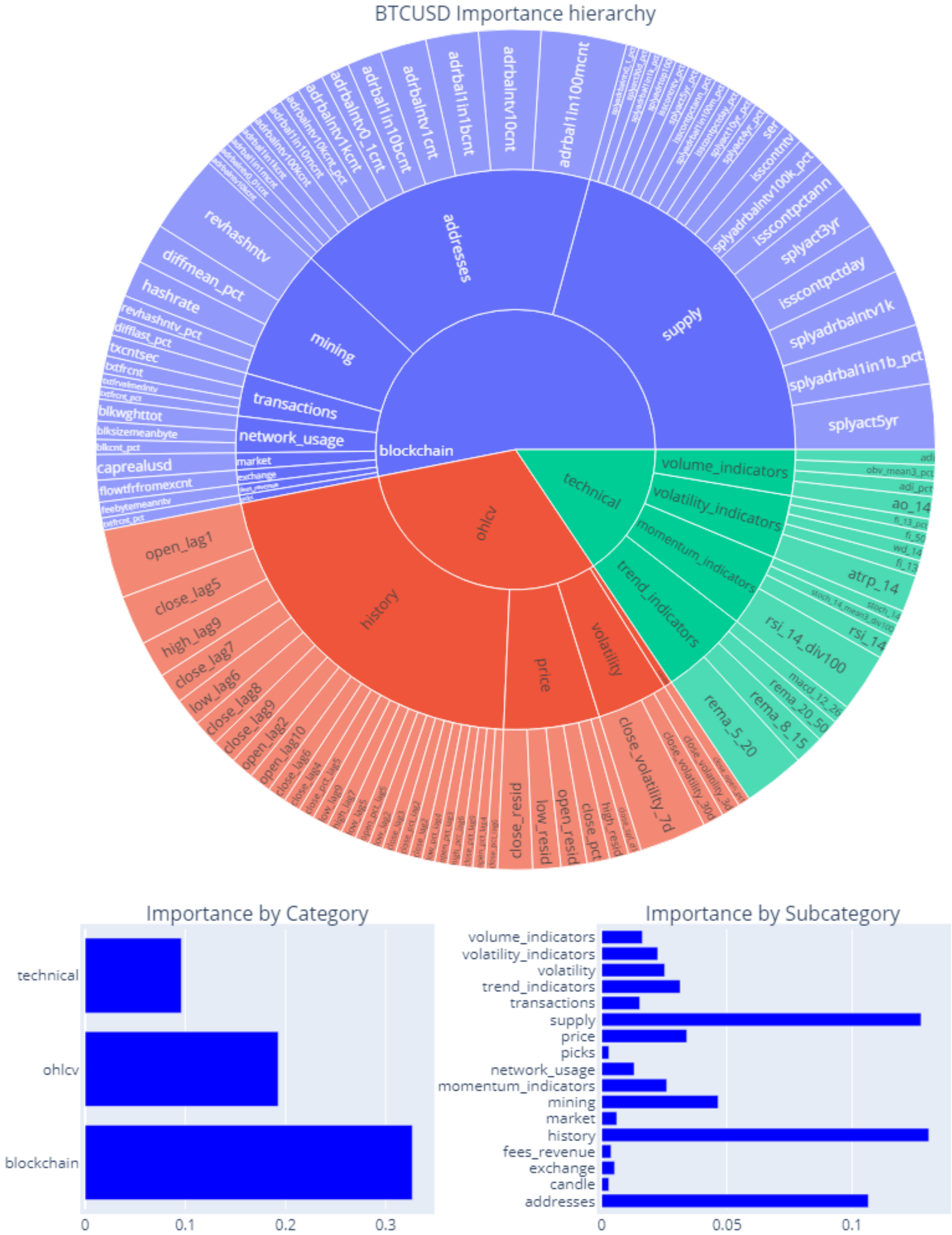

Figure 5.2: Graphs representing selected feature importance for the BTCUSD pair

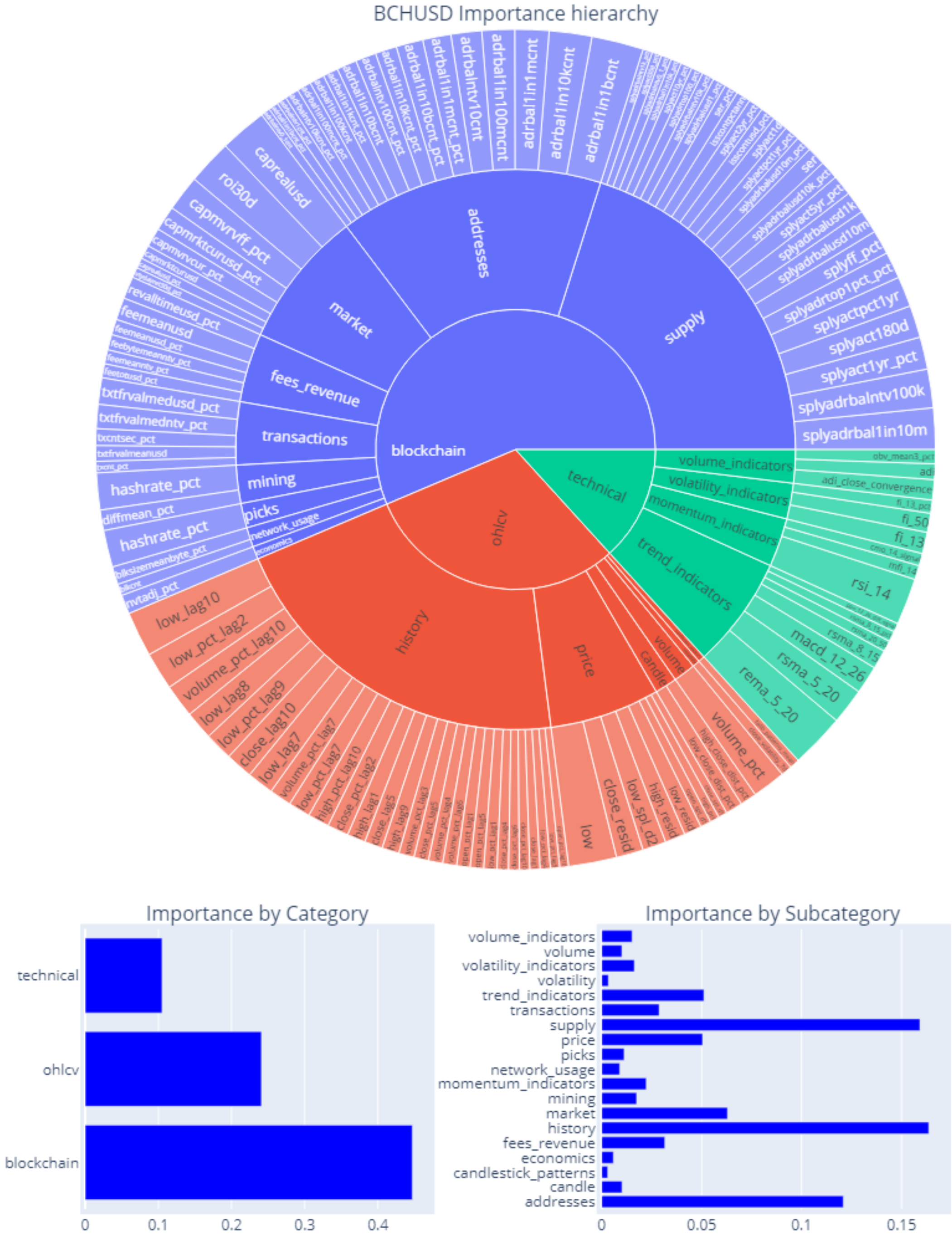

Figure 5.3: Graphs representing selected feature importance for the BCHUSD pair

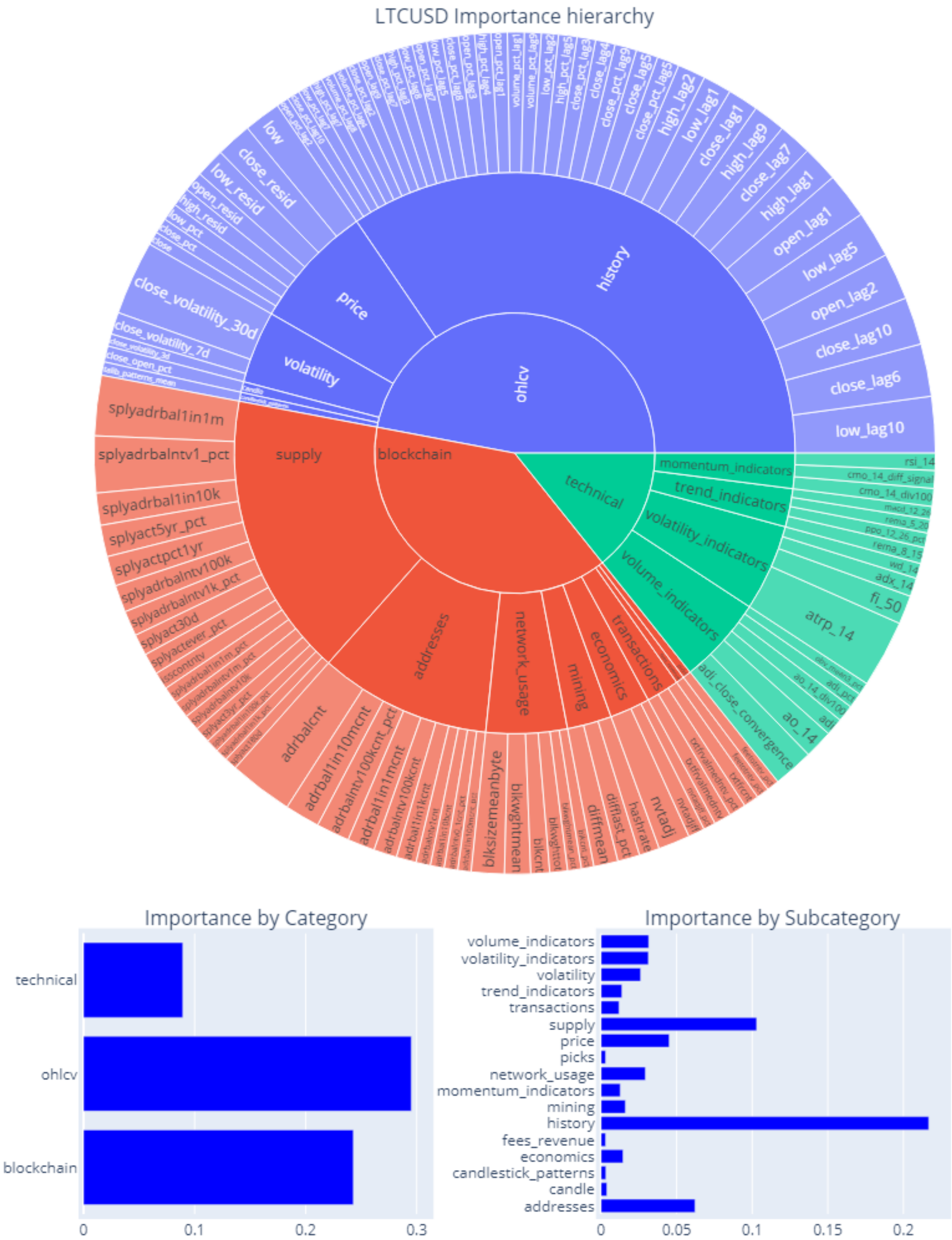

Figure 5.4: Graphs representing selected feature importance for the LTCUSD pair

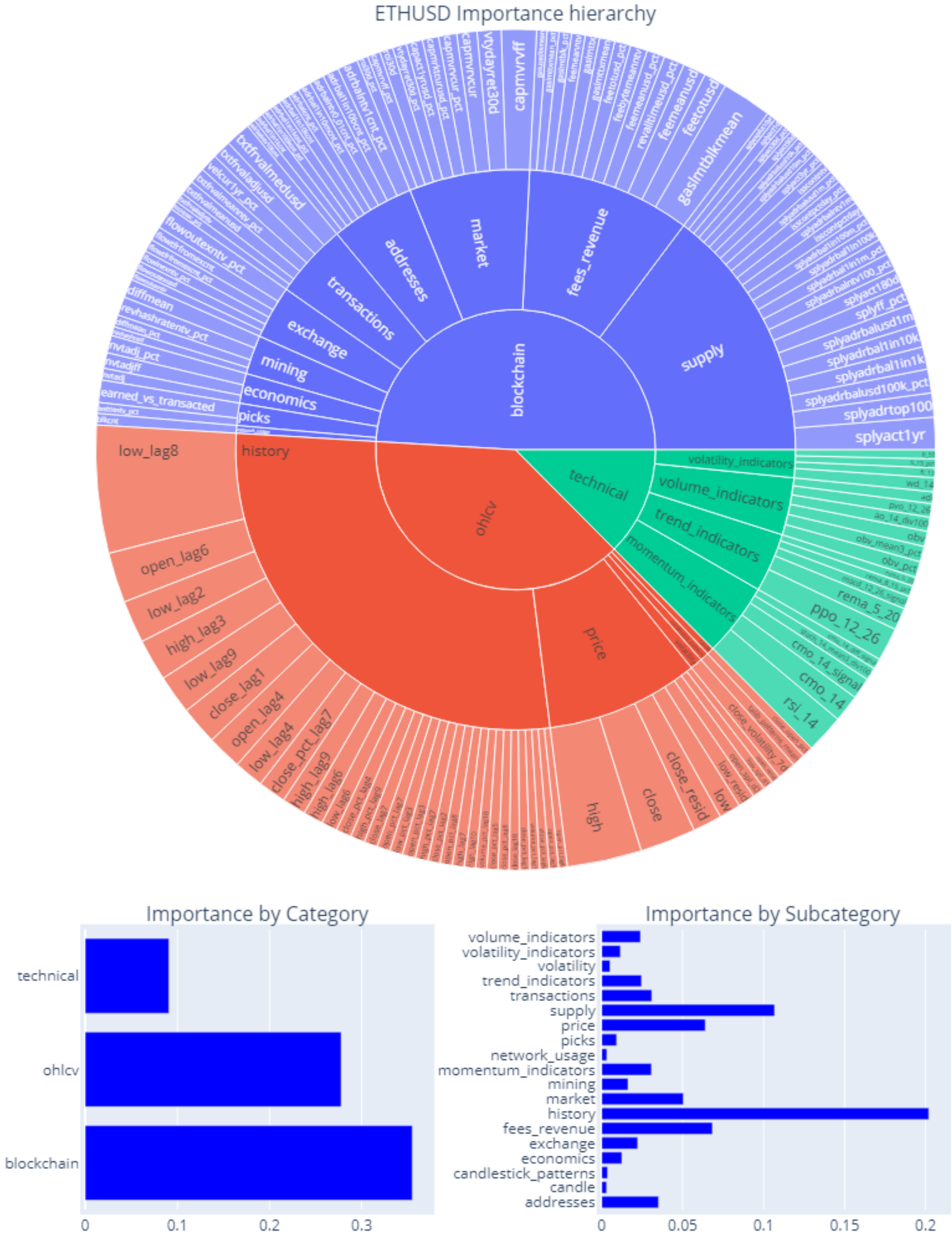

Figure 5.5: Graphs representing selected feature importance for the ETHUSD pair

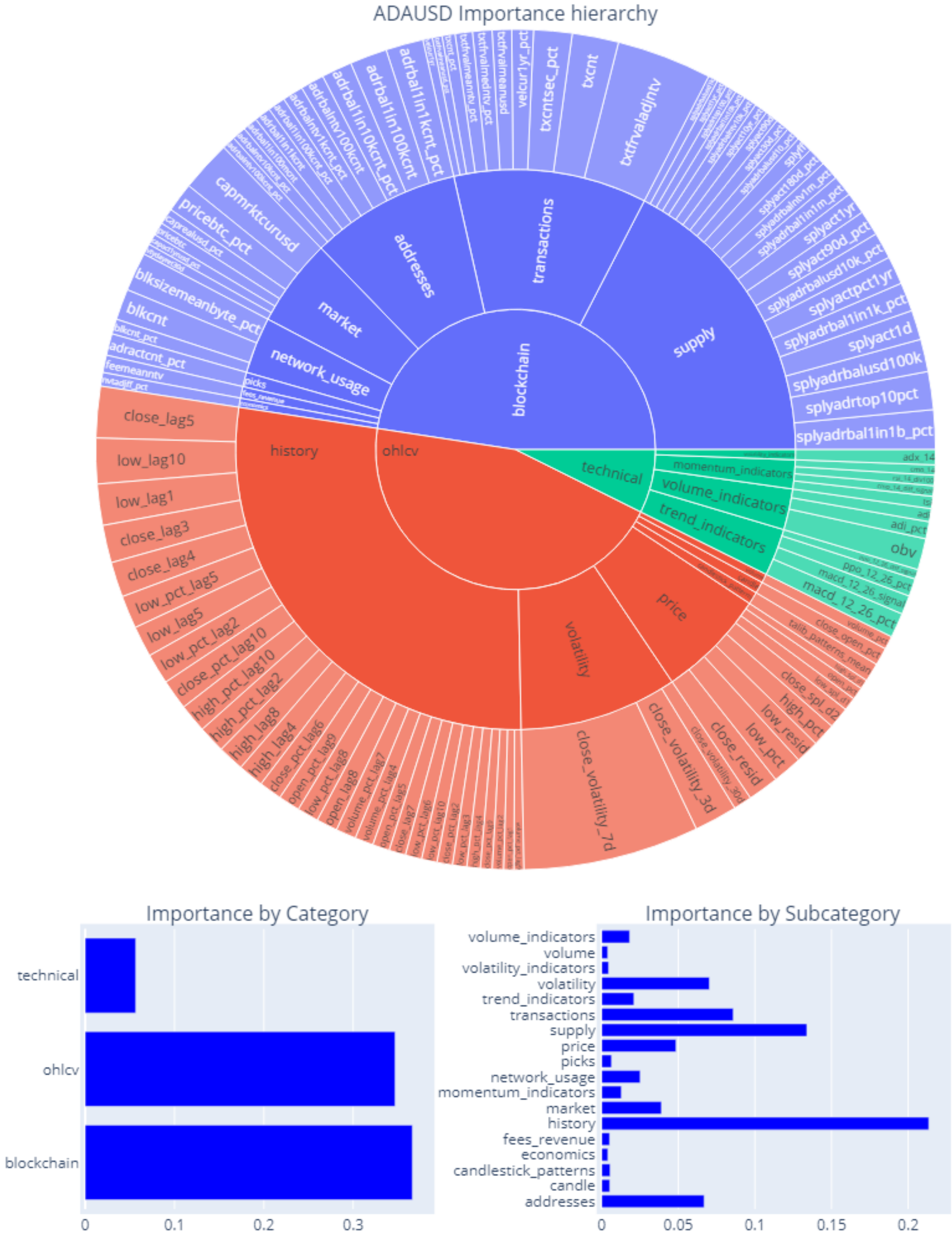

Figure 5.6: Graphs representing selected feature importance for the ADAUSD pair

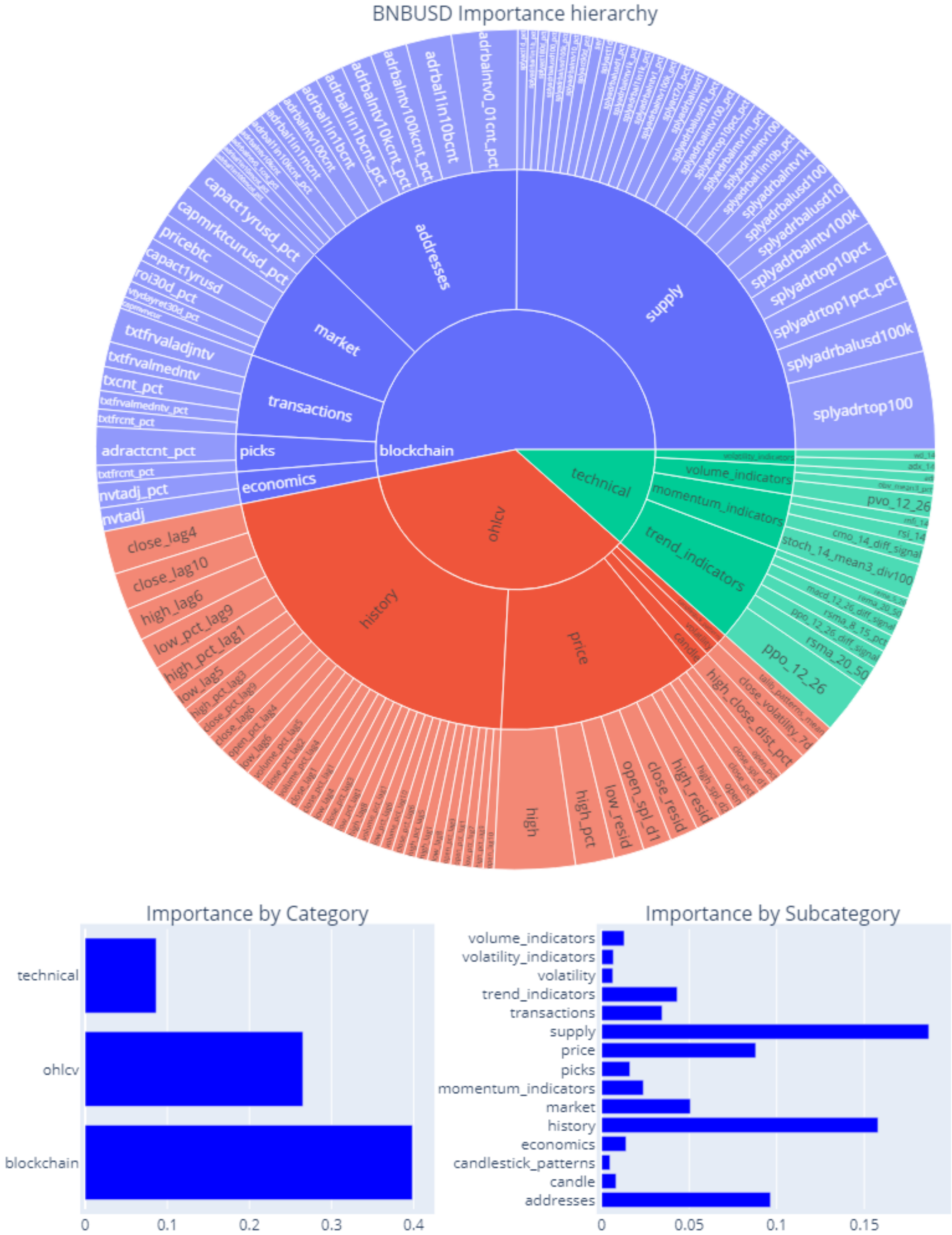

Figure 5.7: Graphs representing selected feature importance for the BNBUSD pair

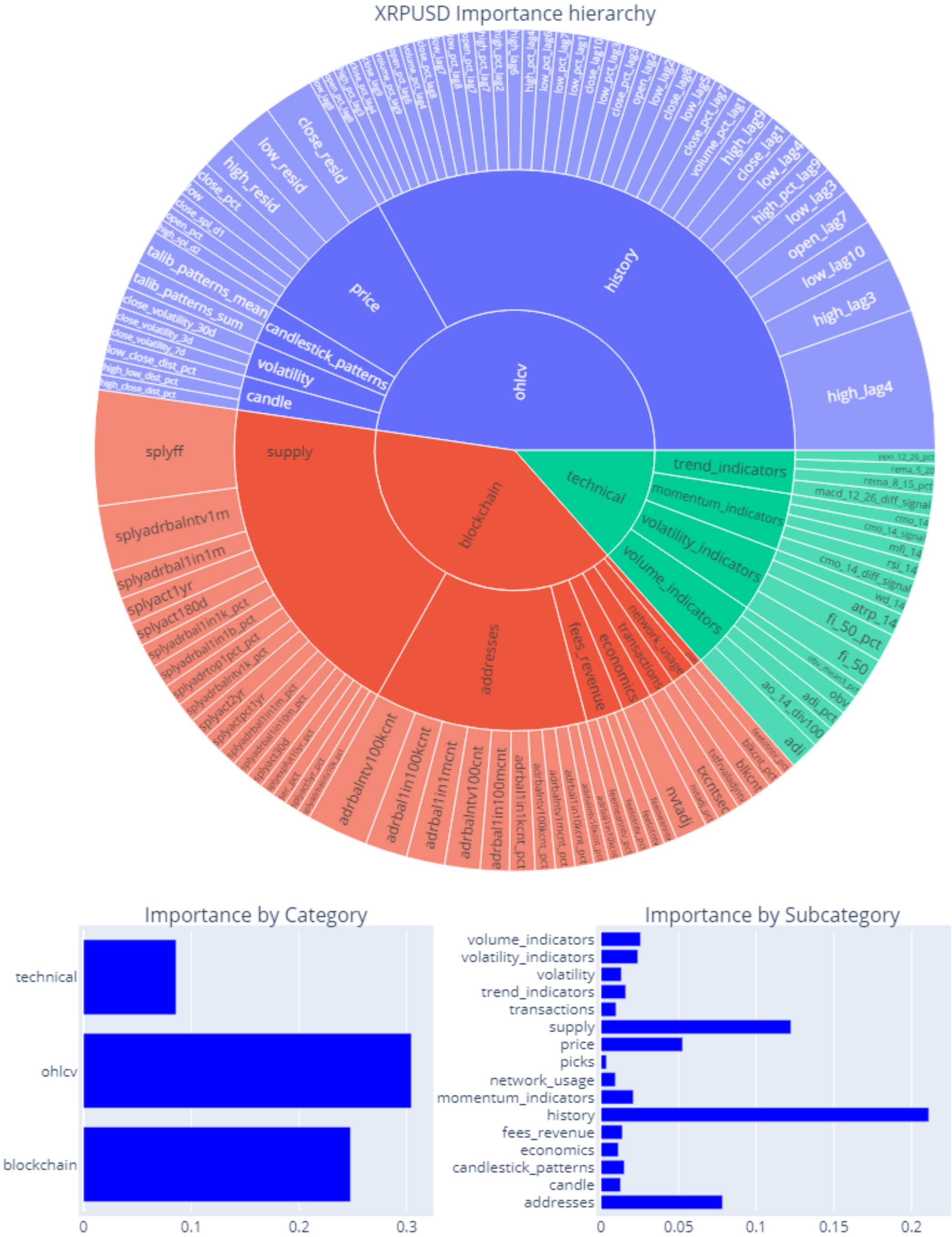

Figure 5.8: Graphs representing selected feature importance for the XRPUSD pair

By looking at the results it is evident how many of the models do not reach the skill threshold, yet present relevant performance in terms of precision (AVG): the k-NN model applied on ZEC in particular has the lowest IBA value among the analysed ones, yet it is second in rank by AVG, meaning the model is highly biased. This can be confirmed by the per-class scores, indicating a 100% success rate in identification of the HOLD class, which is clearly an anomaly. These results also show comparable performance - in terms of precision - between complex ensemble models and simpler algorithms paired with data augmentation techniques.

	Algo	Crypto	W	SELL	HOLD	BUY	AVG	IBA
1	SMOTE+k-NN	WAVES	180	0.500	0.366	0.800	0.585	0.398
2	k-NN	ZEC	240	0.342	1.000	0.571	0.582	0.133
3	Bagging + DT	TRX	240	0.482	1.000	0.379	0.582	0.229
4	SMOTE + SVC (Poly)	EOS	180	0.727	0.389	0.538	0.573	0.368
5	MNB	ADA	240	0.667	0.087	0.524	0.572	0.209
6	SMOTE+k-NN	WAVES	90	0.545	0.424	0.679	0.567	0.413
7	k-NN	WAVES	180	0.439	0.471	0.750	0.566	0.334
8	SVC (Poly)	DOGE	180	0.626	0.295	0.587	0.565	0.370
9	SMOTE+k-NN	BNB	90	0.619	0.619	0.350	0.554	0.405
10	SMOTE+k-NN	TRX	240	0.714	0.235	0.474	0.544	0.294
11	XGBoost	BCH	240	0.593	0.438	0.533	0.543	0.365
12	SMOTE + SVC (Poly)	ZRX	90	0.686	0.242	0.412	0.541	0.324
13	RandomForest	BTC	180	0.532	0.598	0.470	0.530	0.319
14	k-NN	DOGE	90	0.487	0.289	0.634	0.525	0.344
15	SVC (Poly)	EOS	180	0.579	0.333	0.600	0.525	0.392
16	RandomForest	BTC	240	0.569	0.551	0.461	0.521	0.302
17	XGBoost	LINK	180	0.553	0.000	0.632	0.520	0.371
18	RandomForest	NEO	180	0.537	0.000	0.682	0.520	0.255
19	RandomForest	BCH	240	0.511	0.500	0.538	0.518	0.174
20	RandomForest	DASH	240	0.576	0.220	0.583	0.518	0.323
21	MNB	DOGE	180	0.596	0.232	0.521	0.518	0.317
22	SMOTE + SVC (Poly)	XMR	90	0.548	0.231	0.582	0.517	0.312
23	SVC (Poly)	ZRX	90	0.618	0.188	0.484	0.517	0.321
24	RandomForest	LINK	240	0.528	0.000	0.647	0.517	0.355
25	SMOTE+k-NN	QTUM	240	0.702	0.211	0.429	0.516	0.318
26	XGBoost	BCH	180	0.579	0.429	0.471	0.514	0.338
27	RandomForest	DASH	180	0.595	0.182	0.557	0.510	0.318
28	SVC (Poly)	ETC	240	0.548	0.562	0.431	0.509	0.258
29	Bagging + DT	LINK	240	0.511	0.000	0.643	0.509	0.363
30	RandomForest	ETH	240	0.457	0.667	0.460	0.508	0.227
31	SVC (Poly)	TRX	90	0.846	0.059	0.375	0.506	0.226
32	XGBoost	XRP	240	0.571	0.243	0.551	0.505	0.349

Table 5.1: Table reporting precision for each class, average precision and average index balanced accuracy for the top-3 combinations of algorithms and window size across all currencies, sorted by average precision score.

	Algo	Crypto	W	SELL	HOLD	BUY	AVG	IBA
33	XGBoost	XMR	240	0.559	0.357	0.503	0.504	0.331
34	XGBoost	BTC	240	0.429	0.548	0.520	0.504	0.364
35	MNB	XMR	240	0.598	0.371	0.432	0.503	0.294
36	SMOTE + MLP	BTG	90	0.606	0.190	0.500	0.500	0.320
37	SMOTE + SVC (Poly)	BNB	180	0.571	0.532	0.333	0.498	0.335
38	SMOTE+k-NN	BNB	180	0.529	0.569	0.333	0.498	0.333
39	MNB	ZRX	90	0.621	0.250	0.417	0.498	0.309
40	XGBoost	XRP	180	0.571	0.237	0.530	0.496	0.336
41	RandomForest	LTC	180	0.590	0.225	0.503	0.494	0.321
42	XGBoost	ZEC	180	0.548	0.273	0.548	0.488	0.334
43	SMOTE+k-NN	BTG	180	0.680	0.200	0.333	0.488	0.262
44	SVC (Poly)	LTC	90	0.429	0.000	0.684	0.488	0.191
45	XGBoost	LTC	240	0.556	0.263	0.510	0.488	0.324
46	Bagging + DT	ETH	180	0.481	0.647	0.364	0.485	0.236
47	SMOTE+k-NN	ETH	180	0.552	0.257	0.516	0.484	0.265
48	RandomForest	EOS	90	0.509	0.444	0.477	0.483	0.326
49	SMOTE+k-NN	NEO	90	0.567	0.171	0.487	0.483	0.280
50	XGBoost	XEM	240	0.562	0.175	0.545	0.483	0.338
51	k-NN	BTG	180	0.566	0.333	0.429	0.482	0.265
52	MNB	ETC	90	0.528	0.542	0.333	0.479	0.306
53	Bagging + SVC (Poly)	ADA	90	0.556	0.000	0.450	0.477	0.262
54	k-NN	ETC	180	0.417	0.524	0.483	0.476	0.342
55	SMOTE + MLP	QTUM	180	0.585	0.147	0.508	0.473	0.315
56	SMOTE + MLP	QTUM	240	0.542	0.206	0.528	0.473	0.316
57	k-NN	XEM	90	0.327	0.800	0.382	0.469	0.205
58	RandomForest	DASH	90	0.556	0.238	0.467	0.468	0.301
59	k-NN	ADA	90	0.556	0.000	0.421	0.466	0.249
60	XGBoost	NEO	90	0.565	0.167	0.442	0.464	0.285
61	XGBoost	XEM	180	0.508	0.245	0.519	0.463	0.302
62	XGBoost	XRP	90	0.532	0.102	0.530	0.455	0.299
63	XGBoost	ZEC	240	0.528	0.190	0.492	0.442	0.293

Table 5.2: Table reporting precision for each class, average precision and average index balanced accuracy for the top-3 combinations of algorithms and window size across all currencies, sorted by average precision score. (Second Part)

5.3 Trading systems back-testing

In this section we will discuss trading activity of the trading actor in response to signals generated by some of the most skilled machine learning models in comparison with a buy-and-hold baseline strategy, in which the whole equity amount is bought at the start of the simulation period and sold at the last trading day. The top-3 trading results by equity at the last trading day have been collected and are presented in tables 5.3 and 5.4. By comparing these results with the ones presented in 5.2 we can evince how the best performing models in terms of precision do not necessarily correspond with the best ones in terms of trading performance, with more complex algorithms such as MLP, XGBoost and Random Forests being more reliable in terms of signal generation. Nevertheless, in most cases the proposed systems beat the baseline strategy,

	Algo	Crypto	W	Baseline	Equity
1	MLP	BTC	240	61636.73	21918.67
2	SMOTE + MLP	BTC	240	61636.73	21273.61
3	XGBoost	BTC	180	61636.73	19673.31
4	RandomForest	DOGE	90	11095.23	18373.29
5	XGBoost	DOGE	90	11095.23	18027.81
6	XGBoost	LTC	240	6981.00	17578.25
7	MNB	DOGE	240	11095.23	17377.14
8	XGBoost	XEM	240	416.08	16259.38
9	RandomForest	LTC	240	6981.00	16196.13
10	XGBoost	XRP	240	4831.23	15772.18
11	RandomForest	LTC	180	6981.00	15495.45
12	RandomForest	DASH	240	2295.38	15226.74
13	RandomForest	XRP	240	4831.23	15007.59
14	RandomForest	DASH	180	2295.38	14689.75
15	RandomForest	XEM	240	416.08	14641.43
16	XGBoost	DASH	240	2295.38	14568.04
17	XGBoost	XMR	240	1249.09	14103.73
18	XGBoost	XRP	180	4831.23	13971.98
19	XGBoost	XEM	180	416.08	13795.71
20	k-NN	ETH	180	1587.31	13571.55
21	RandomForest	ETH	240	1587.31	13555.45
22	MLP	XMR	240	1249.09	13382.90
23	XGBoost	ZEC	180	3338.68	13025.06
24	SMOTE + MLP	XMR	240	1249.09	12962.99
25	XGBoost	ETH	240	1587.31	12836.76
26	SMOTE + MLP	ZEC	180	3338.68	12224.74
27	MLP	QTUM	240	4171.42	12104.75
28	Bagging + DT	WAVES	240	13800.90	12057.29
29	RandomForest	ZEC	180	3338.68	12025.55
30	XGBoost	QTUM	240	4171.42	11836.22

Table 5.3: Top-3 trading results by final equity value for each currency, sorted by equity value.

	Algo	Crypto	W	Baseline	Equity
31	RandomForest	WAVES	240	13800.90	11810.76
32	RandomForest	WAVES	180	13800.90	11806.79
33	XGBoost	BCH	240	3448.88	11698.89
34	MLP	EOS	180	5527.42	11690.35
35	SMOTE + k-NN	QTUM	240	4171.42	11593.43
36	MLP	LINK	240	8503.54	11573.73
37	XGBoost	BCH	180	3448.88	11520.95
38	RandomForest	LINK	180	8503.54	11516.37
39	MLP	EOS	240	5527.42	11457.38
40	RandomForest	LINK	240	8503.54	11398.24
41	SMOTE + SVC (Poly)	EOS	180	5527.42	11361.54
42	XGBoost	TRX	180	7129.59	11346.05
43	MLP	TRX	240	7129.59	11238.91
44	SVC (Poly)	TRX	180	7129.59	11212.79
45	MNB	ZRX	90	4620.10	11171.49
46	RandomForest	BNB	240	4902.53	10967.19
47	RandomForest	BCH	240	3448.88	10964.19
48	RandomForest	NEO	180	4480.13	10956.35
49	SMOTE + MLP	ZRX	90	4620.10	10934.07
50	Bagging + DT	BNB	240	4902.53	10834.45
51	RandomForest	BTG	240	5065.39	10820.52
52	RandomForest	BTG	180	5065.39	10789.44
53	RandomForest	ZRX	240	4620.10	10761.93
54	RandomForest	BNB	180	4902.53	10740.11
55	k-NN	NEO	180	4480.13	10668.82
56	SMOTE + k-NN	NEO	180	4480.13	10653.59
57	XGBoost	ETC	180	3309.49	10638.10
58	SMOTE + MLP	BTG	240	5065.39	10621.64
59	MNB	ADA	240	6502.07	10542.48
60	XGBoost	ADA	90	6502.07	10024.33
61	k-NN	ADA	90	6502.07	9947.57
62	SVC (Poly)	ETC	240	3309.49	9651.03
63	MNB	ETC	90	3309.49	9648.92
64	MLP	VEN	180	252997.47	7802.80
65	RandomForest	VEN	180	252997.47	7742.75
66	MLP	VEN	240	252997.47	7147.65

Table 5.4: Top-3 trading results by final equity value for each currency, sorted by equity value. (Second Part)

The proposed graphs allow analysis of trades placed in the backtesting phase, expressing insights about how the automated backtesting actor reacts to signals and results of the trades on equity values. While trading activity is continuous in the analysed periods, plots have been split into quarters to promote interpretability. All subplots share the same temporal axis, with the first three plots representing candlestick data of the corresponding quarter with different overlays:

- The first subplot reports volume information as a bar chart on the secondary axis, as well as generated signals represented as scatter plot overlays according to the legends.

- The second and third subplots report respectively long-selling and short-selling orders, represented as color-coded scatter plot overlays: an opened order is represented by a circle of a certain color, and its closure is represented as a filled dot of the same color.

- The fourth subplot represents the asset's equity value and baseline strategy as line plots.

Figures 5.9 - 5.17 represent activity of the best trading system based on an XGBoost model, trained on 180 day windows of the BTCUSD dataset - which is the largest one in terms of samples. As we can see, at the beginning of Q4 2016 the system starts missing trades, giving the baseline strategy an advantage which becomes more relevant across 2017. However the system is still profitable, recovering some of the distance with the baseline strategy during bearish cycles.

Figure 5.18 represents trades placed the trading system based on the best model by precision based on a k-NN classifier mitigating class imbalance by means of the SMOTE technique, trained on 180 days wide windows of the WAVESUSD dataset. As it is evident from the equity plot, the system is able to compensate the bearish trend by consistently taking short opportunities. Despite the high confidence in the model, trade profits are limited by take profit conditionals, with the equity being surpassed by the baseline strategy in the last part of the quarter. The situation is similar to figure 5.19, representing the same setup trained on 90 days wide windows - constituting the best model by IBA.

5.4 Trading signals explainability

This section proposes an analysis of the impact of features in the decision making processes behind the relevant BUY/SELL outputs of models based on the XGBoost classifier - as it is nowadays one of the most popular and widely used machine learning algorithms. The proposed charts are divided in three main sections:

Figure 5.9: Results from backtesting on the BTCUSD pair using XGBoost in Q4 2016

- The first subplot is a line plot representing test precision for the analysed class, calculated on outputs from the whole testing period.
- The second subplot is a line plot representing the absolute mean SHAP value taken across each day's training window for the top 10 values by influence from the three beeswarm charts in the last section, sharing the X (temporal) axis with the first subplot.
- The third subplot is in turn split into three subplots, representing beeswarm summary plots from SHAP values belonging respectively to the first, median and last training windows. SHAP beeswarm plots indicate how a feature assuming a certain color-coded value influences the model output in a positive

Figure 5.10: Results from backtesting on the BTCUSD pair using XGBoost in Q1 2017

$(x > 0)$ or negative $(x < 0)$ way.

These visualizations may be useful to a quantitative trader in assessing what variables influence precision for relevant outputs providing insights not only on the factors influencing signal generation, but in how these change across the testing period as well.

A preliminary analysis of the results confirms the influence of residuals from STL decomposition of OHLC data, as hypothesized in 5.1: these features figure in almost every plot, even outperforming the other features in models where the testing period is particularly extended such as in the BTCUSD pair. Blockchain features' relevance is consistent as well, albeit the specific relevant features for each currency shift in value, probably in response to protocol changes and evolution.

Figure 5.11: Results from backtesting on the BTCUSD pair using XGBoost in Q2 2017

Analysis on models trained on 180 days window for the BTCUSD pair shows how predictions for the BUY class (Figure 5.20) are heavily influenced by residuals from STL decomposition. In particular, closing price residual seems to be the overall most important feature across the tested period. Blockchain-related features are overall less relevant, with the attention shifting from hashrate and issuance in the starting period to outgoing flows from exchanges during the 2017 bull cycle, and finally settling back on issuance and supply metrics across the second half of 2018. Looking at the SELL class (Figure 5.21) residuals are still an important subset of features, however in this case they are approached by candlestick analysis-related features: this probably due to their intrinsic sentiment information, as bearish

Figure 5.12: Results from backtesting on the BTCUSD pair using XGBoost in Q3 2017

patterns induce fear in the market, leading to massive sales. Again, blockchain-related features seem to be marginal in the decision process but this time the addresses subcategory is constantly present across the testing set.

BUY class for the LTCUSD pair of models trained on 240 days wide windows (Figure 5.22) see a similar situation to BTC, with STL residuals being the most important subset of OHLC features and supply/transaction subcategories being among the most relevant among the blockchain class. Technical indicators are also present - more than in the BTC pair - with relevance shifting from MACD to volatility and momentum. The situation for the SELL class (Figure 5.23) sees, apart from price residuals, historical price and volume information being of relevance while technical indicators' presence is inconstintent. Blockchain-derived

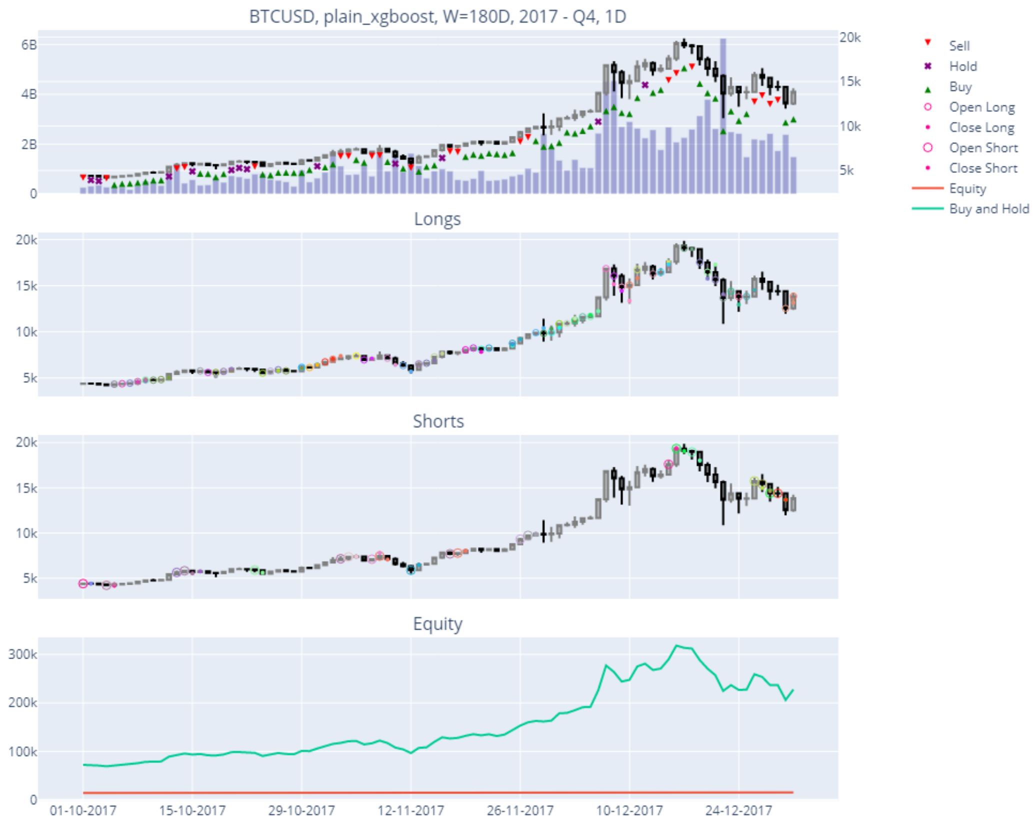

Figure 5.13: Results from backtesting on the BTCUSD pair using XGBoost in Q4 2017

features' presence is inconsistent as well, with interest shifting between supply and transactions subcategories.

Analysing the DOGEUSD pair, models trained on 90 days wide windows show different results than the other currencies: outcomes for the BUY class (Figure 5.24) are mainly driven by historical data and other different factors varying across the analysed period; they are initially affected by transaced value and market cap, while in the middle periods the most important features are related to candlestick analysis, free-float NVT and others belonging to the supply subcategory. The last part of the testing period sees increase in relevance of transactions, issuance and addresses related features. Analysis on the SELL class (Figure 5.25) sees an initial influence of historical data, shifting to features deriving from candlestick patterns,

Figure 5.14: Results from backtesting on the BTCUSD pair using XGBoost in Q1 2018

transactions and addresses-related features across the testing period.

Figure 5.15: Results from backtesting on the BTCUSD pair using XGBoost in Q2 2018

Figure 5.16: Results from backtesting on the BTCUSD pair using XGBoost in Q3 2018

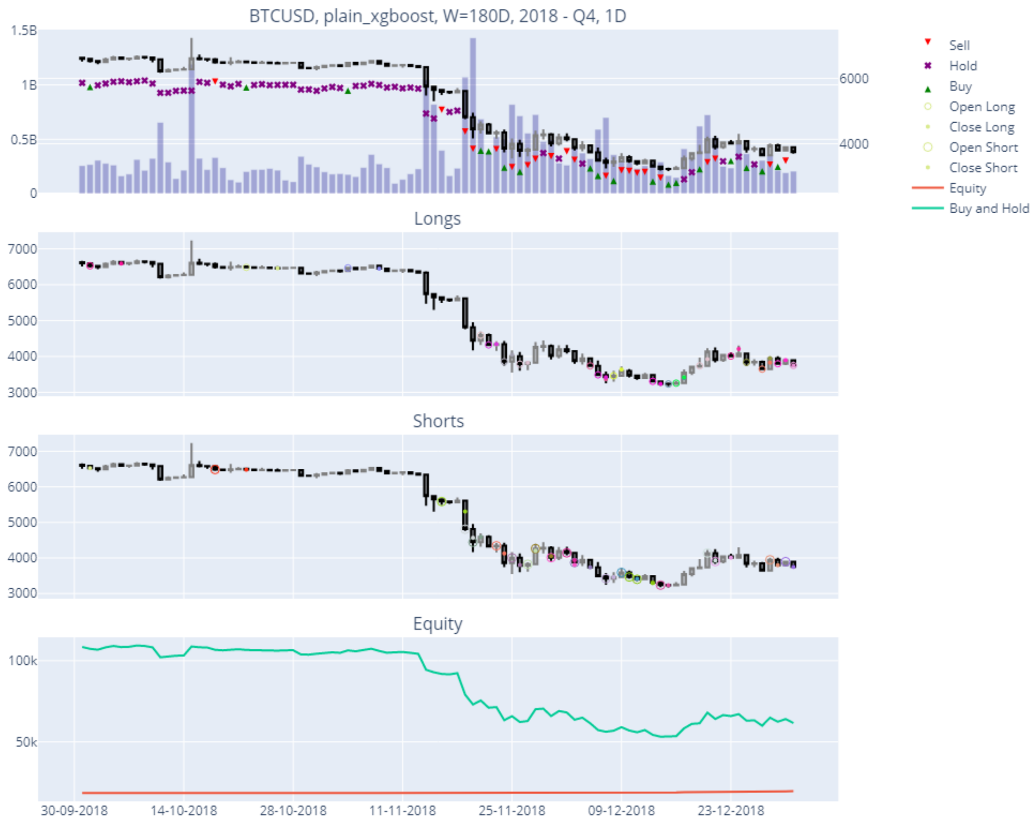

Figure 5.17: Results from backtesting on the BTCUSD pair using XGBoost in Q4 2018

Figure 5.18: Results from backtesting on the WAVESUSD pair using SMOTE + k-NN in Q4 2018 using 180D windows

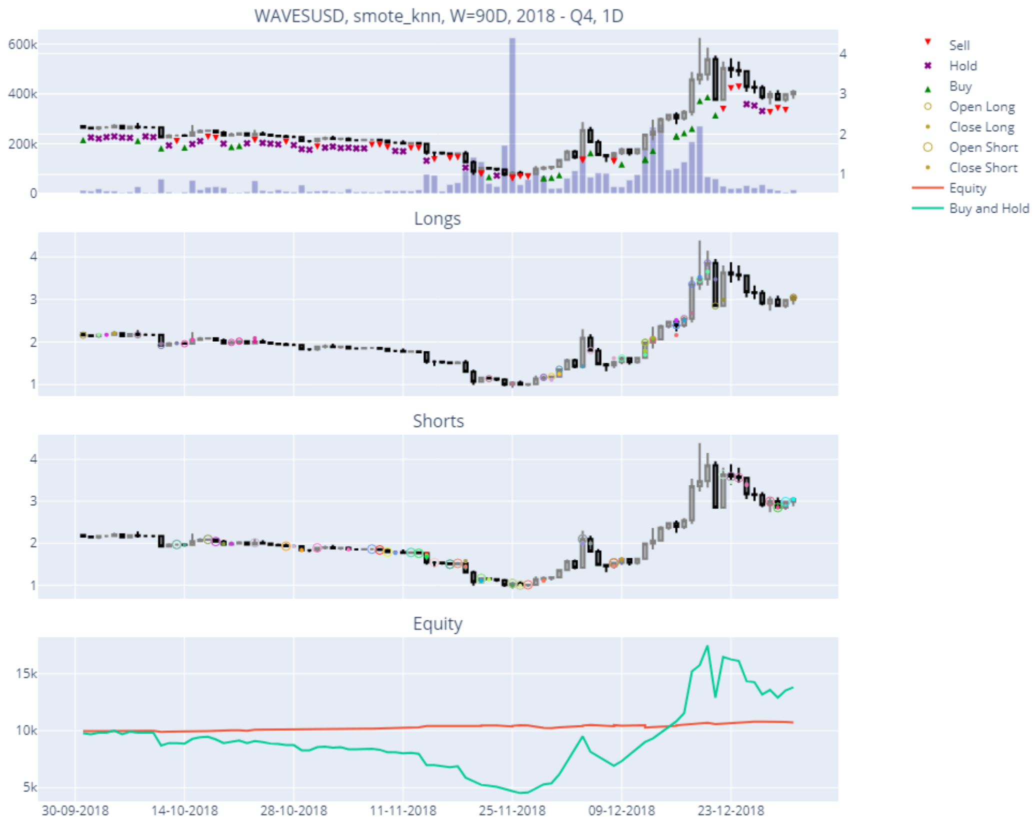

Figure 5.19: Results from backtesting on the WAVESUSD pair using SMOTE + k-NN in Q4 2018 using 90D windows

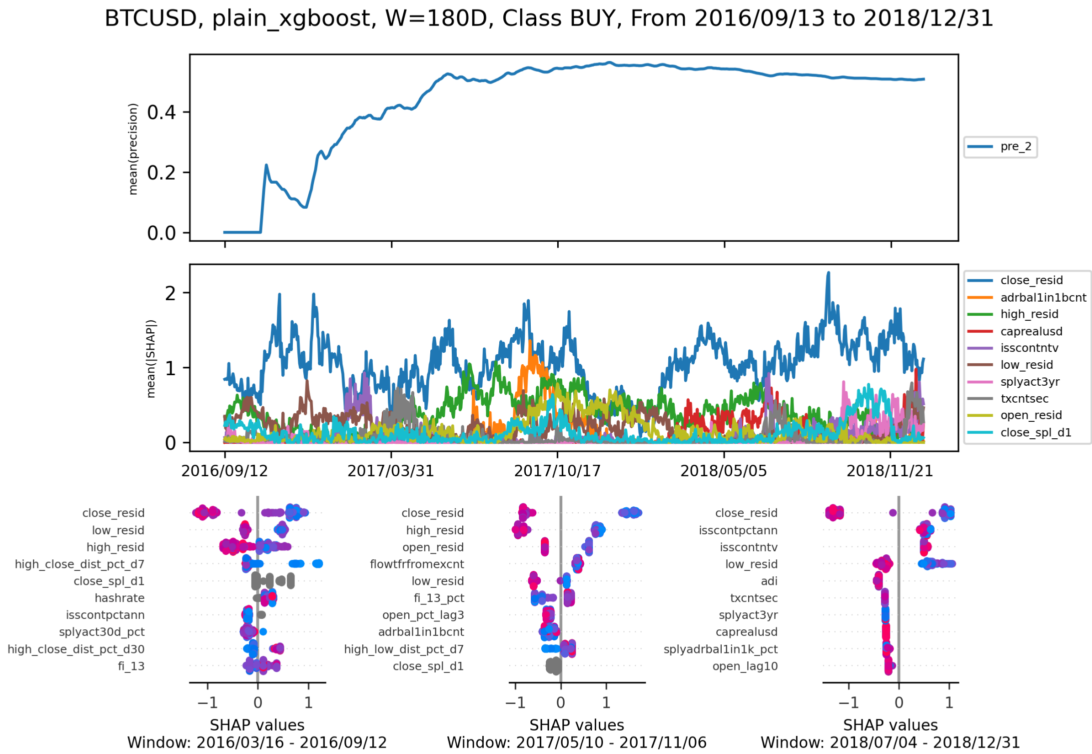

Figure 5.20: SHAP analysis for BUY class of the BTCUSD pair using XGBoost

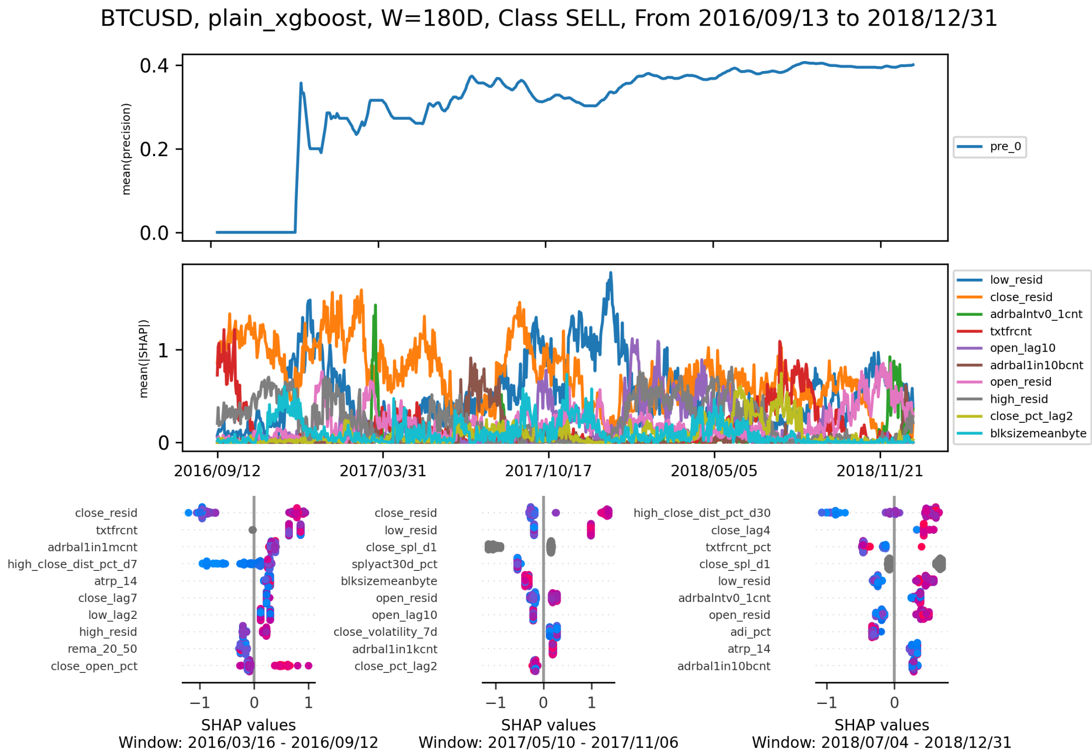

Figure 5.21: SHAP analysis for SELL class of the BTCUSD pair using XGBoost

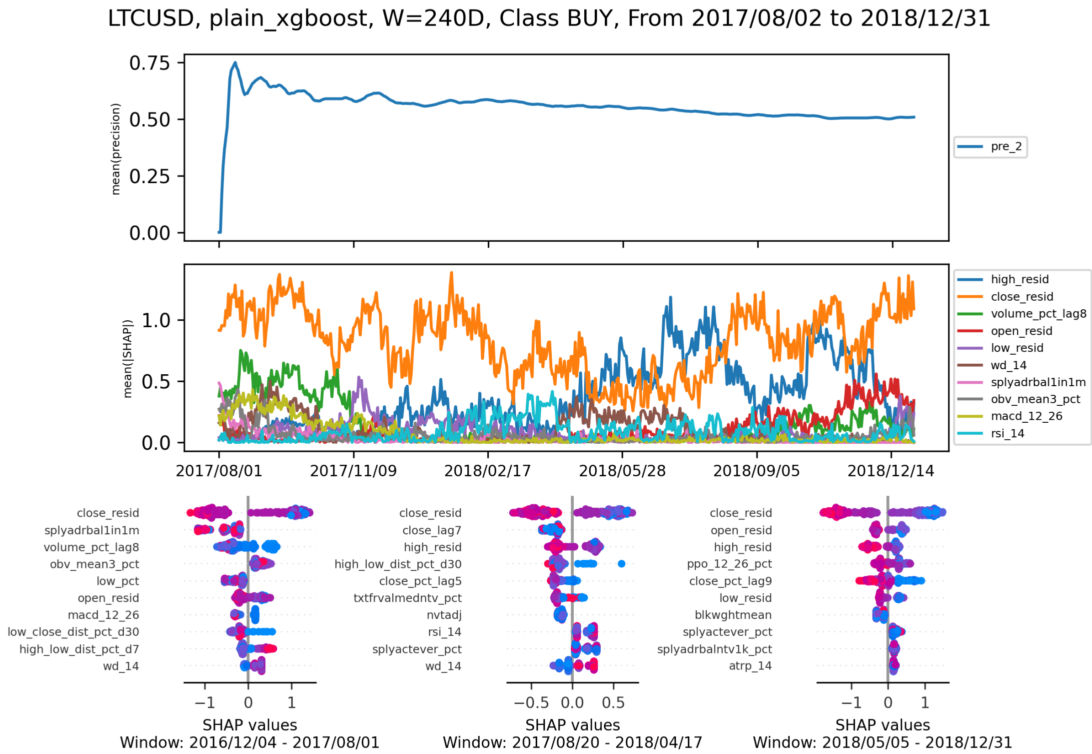

Figure 5.22: SHAP analysis for BUY class of the LTCUSD pair using XGBoost

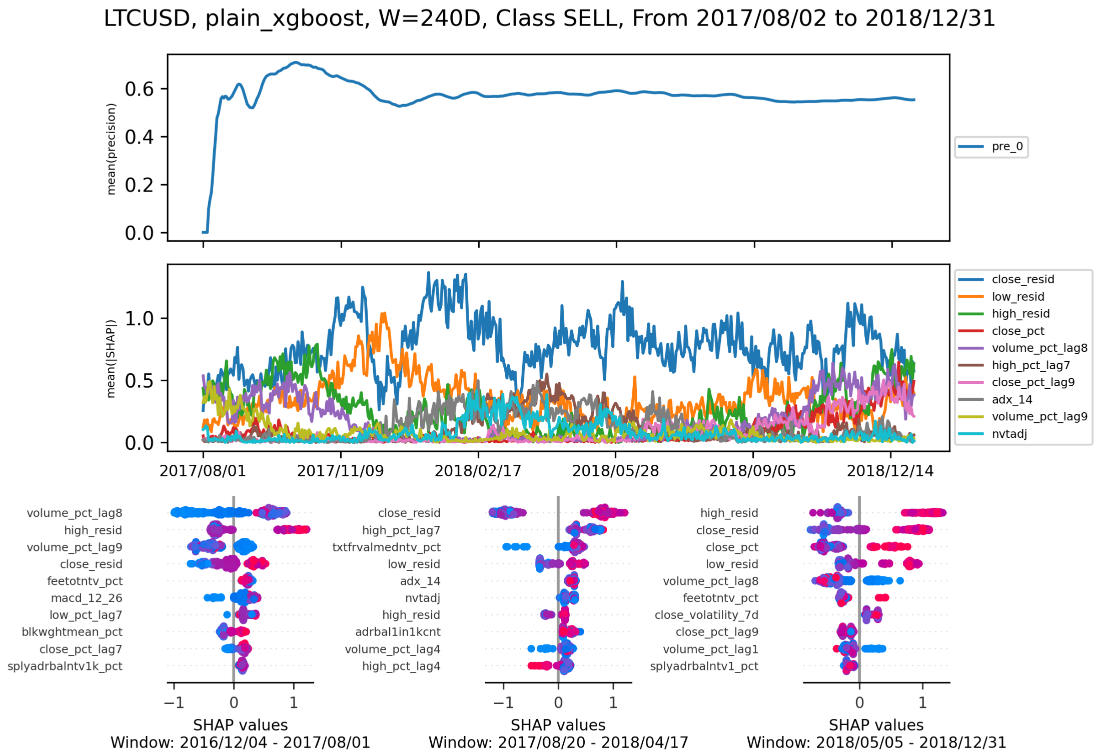

Figure 5.23: SHAP analysis for SELL class of the LTCUSD pair using XGBoost

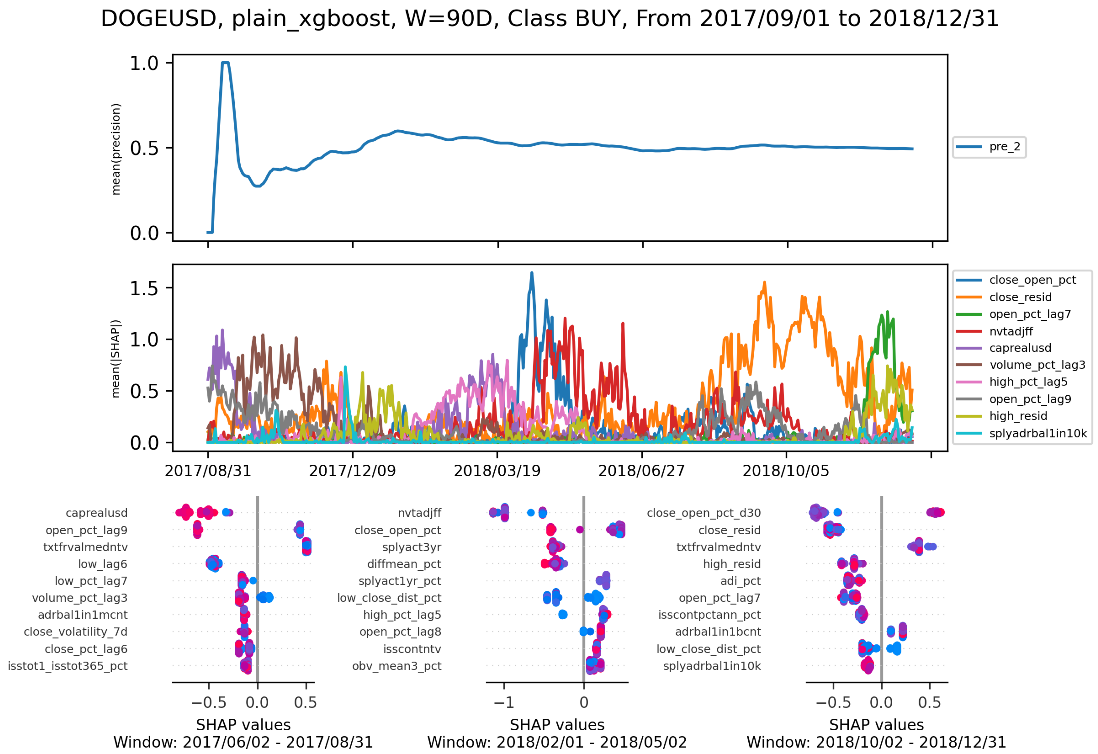

Figure 5.24: SHAP analysis for BUY class of the DOGEUSD pair using XGBoost

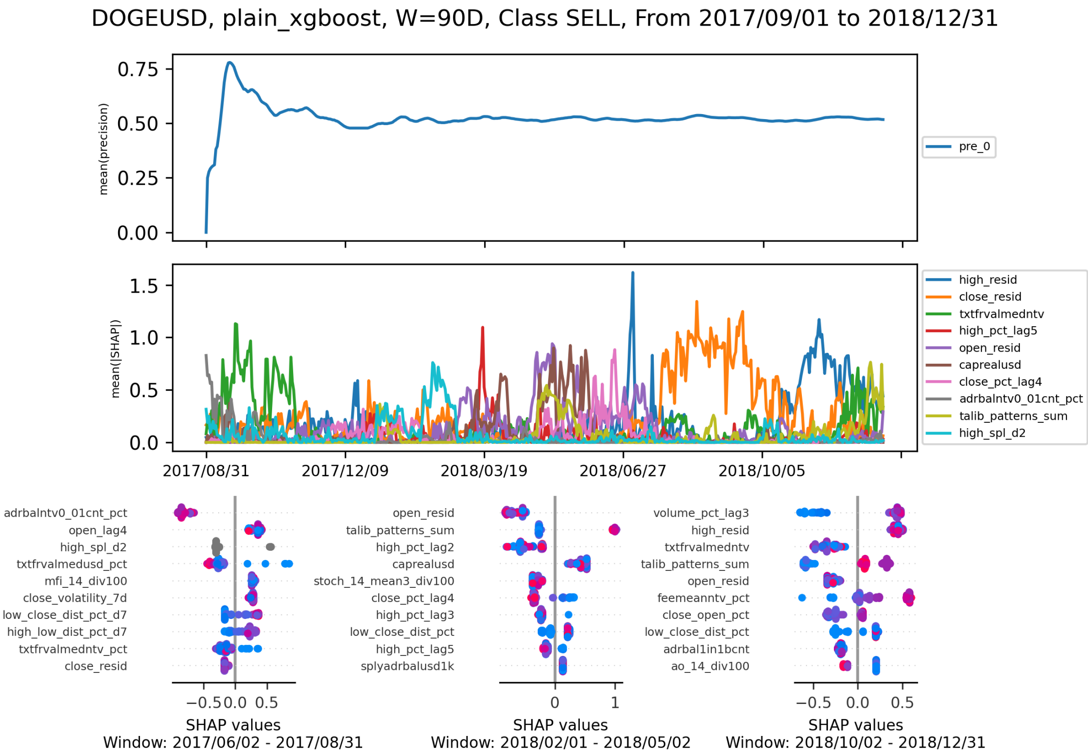

Figure 5.25: SHAP analysis for SELL class of the DOGEUSD pair using XGBoost

Chapter 6

Conclusions and future work

This work aims to be a study for exploration of possibilities offered by the cryptocurrency markets: while relatively young, their capitalization is nowadays (June 2021) above 1 Trillion USD, making them an important financial reality whose popularity is bound to increase in the coming years.

Challenges in price forecasting on these markets lie in the data, both in terms of sourcing and uniformity: while market data is freely available on most exchanges, complete blockchain-related information is harder to find due to speculation and computational complexity in its extraction. Differences in protocol, implementations and age imply each cryptocurrency has a different set of features and number of data points, sometimes limiting model applicability. Moreover, using timeseries data as input for a classifier poses another different kind of challenge: classifiers are usually thought for data sets where class distributions are uniform, ignoring the presence of trends and seasonality thus requiring specific precautions both in data processing and handling. After analysis of the available data sources, freely available community blockchain data from Coinmetrics was used in conjunction with market data from the Kraken exchange. After cleansing, input datasets were uploaded on centralized databases, pre-processed and joined to obtain one dataset for each cryptocurrency pair with daily granularity, which was further split in training and a validation sets.

Due to the huge number of input features in each dataset, a feature selection step was performed for both reducing dimensionality and gaining valuable insights about the characteristics of the market as a whole and each of the studied currencies, showing how the influence of protocol and "offered service" affects the relative importance distribution. Once relevant features were fixed, we performed hyperparameter optimization with cross validation to improve generalization of the outputs and ensure model performance.

Resulting models were tested for making predictions on the validation sets in a sliding window fashion, exploring performance of different algorithms ranging from the simpler ones such as k-NN and SVC to more complex ensemble methods such as Random Forests and XGBoost. Furthermore, generated signals were used for back-testing trades by simulating operations on a real-world exchange by means of a custom developed trading agent based on the rules and limitations of the Kraken exchange. Analysis of backtesting results have shown how the best models in terms of performance metrics do not always correspond to the best trading systems in terms of final equity value.

The main key point in favor of the proposed work lies in the exploration of new features and explanation of their impacts on the classification models: the proposed charts express valuable information which could be useful to quantitative traders in choosing what metrics to monitor in their decision making processes. Moreover, backtesting results show how although trend-dependant, trading results from the proposed systems can be profitable by beating the baseline buy and hold strategy most of the times, with the best results providing good performance in signal generation and thus order handling. In most cases the proposed trading systems beat or are on par with the baseline buy and hold strategy, proving how the proposed models are successful in identifying patterns that lead to successful trades. However, inability to predict high volatility events, in conjunction with the adoption of a fixed fractional strategy for automated trading expose the systems to potential losses: for this reason, this work could be used for building Decision Support Systems (DSS) to be used by human quantitative traders, rather than fully automated trading systems.

Future work could involve the use of lower granularities for input data: as cryptocurrency markets are always active and present high volatility events happening on lower time frames, the models could be able to find more relevant patterns, further increasing trades performance. Other possible improvements could involve the integration of portfolio management models in the back-testing agent, determining position size (instead of adopting a fixed fractional strategy) on the base of volatility and risk:reward estimates.

Bibliography

[1] Giancarlo Giudici, Alistair Milne, and Dmitri Vinogradov. «Cryptocurrencies: market analysis and perspectives». In: *Journal of Industrial and Business Economics* 47.1 (Sept. 2019), pp. 1–18. DOI: 10.1007/s40812-019-00138-6. URL: https://doi.org/10.1007/s40812-019-00138-6 (cit. on p. 4).

[2] Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. «STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion)». In: *Journal of Official Statistics* 6 (1990), pp. 3–73 (cit. on p. 31).

[3] John J Murphy. *Technical analysis of the financial markets : a comprehensive guide to trading methods and applications*. eng. New York: New York Institute of Finance, 1999. ISBN: 0735200661 (cit. on p. 34).

[4] Giuseppe Attanasio, Luca Cagliero, Paolo Garza, and Elena Baralis. «Quantitative Cryptocurrency Trading: Exploring the Use of Machine Learning Techniques». In: DSMM'19 (2019). DOI: 10.1145/3336499.3338003. URL: https://doi.org/10.1145/3336499.3338003 (cit. on pp. 35, 36).

[5] J.W. Wilder. *New Concepts in Technical Trading Systems*. Trend Research, 1978. ISBN: 9780894590276. URL: https://books.google.it/books?id=WesJAQAAMAAJ (cit. on p. 38).

[6] A. Elder. *Trading for a Living: Psychology, Trading Tactics, Money Management*. Wiley Finance. Wiley, 1993. ISBN: 9780471592242. URL: https://books.google.it/books?id=rB2mIvrHLxMC (cit. on p. 38).

[7] J.E. Granville. *New Key to Stock Market Profits*. Prentice-Hall, 1963. URL: https://books.google.li/books?id=0XV0AQAAMAAJ (cit. on p. 38).

[8] Tianqi Chen and Carlos Guestrin. «XGBoost». In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2016). DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785 (cit. on p. 54).

[9] V. García, R. A. Mollineda, and J. S. Sánchez. «Index of Balanced Accuracy: A Performance Measure for Skewed Class Distributions». In: *Pattern Recognition and Image Analysis*. Ed. by Helder Araujo, Ana Maria Mendonça, Armando J. Pinho, and María Inés Torres. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 441–448. ISBN: 978-3-642-02172-5 (cit. on p. 57).

[10] Scott M Lundberg and Su-In Lee. «A Unified Approach to Interpreting Model Predictions». In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 4765–4774. URL: `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf` (cit. on p. 60).

[11] Lloyd S. Shapley. *A Value for N-Person Games*. Santa Monica, CA: RAND Corporation, 1952. DOI: `10.7249/P0295` (cit. on p. 60).

[12] Scott M. Lundberg et al. «From local explanations to global understanding with explainable AI for trees». In: *Nature Machine Intelligence* 2.1 (2020), pp. 2522–5839 (cit. on p. 61).