Expert Systems With Applications 238 (2024) 121806

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A profitable trading algorithm for cryptocurrencies using a Neural Network model

Mimmo Parente [a], Luca Rizzuti [a,*], Mario Trerotola [a,b]

[a] Dipartimento di Scienze Aziendali - Management and Innovation Systems, Università degli Studi di Salerno, Salerno, Italy
[b] Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy

ARTICLE INFO

Keywords:
Cryptocurrencies
Machine learning
Neural network
Price prediction
Algorithmic trading
Explainable AI
Backtesting
Shapley values

ABSTRACT

Algorithmic trading enables the execution of orders using a set of rules determined by a computer program. Orders are submitted based on an asset's expected price in the future, an approach well suited for high-volatility markets, such as those trading in cryptocurrencies. The goal of this study is to find a reliable and profitable model to predict the future direction of a crypto asset's price based on publicly available historical data. We first develop a novel labeling scheme and map this problem into a Machine Learning classification problem. The model is then validated on three major cryptocurrencies through an extensive backtest over a bull, bear and flat market. Finally, the contribution of each feature to the classification output is analyzed.

1. Introduction

A financial trading system on public market exchanges comprises a set of rules and tools that help the trader agent make the best decisions during the investment phase. These rules and tools are trading algorithms applied to data relating to one or more financial assets, in order to identify and exploit profit opportunities. Multiple data sources can be used to make trading decisions, and one of the most widely publicized projects using such techniques in financial applications is *Standard & Poor's Neural Fair Value 25* portfolio. This uses an artificial neural network to select 25 stocks on a weekly basis from a total of 3000 stocks, with the aim of outperforming the market by calculating a stock's weekly fair value based on fundamental analysis. In the field of securities trading, the utility of complex models such as Neural Networks (NN), Support Vector Machines (SVM) and hybrid models has been extensively studied and promising results have been obtained (Kumbure et al., 2022). However, information regarding the incorporation of such methods into trading floor operations tends to remain hidden to the public, for commercial proprietary reasons (Gerlein et al., 2016). Hence, there is an ample of room to explore automated trading using modern machine learning (ML) approaches.

In this paper, we are interested in cryptocurrency trading. We aim to explore the possibility to learn generic price patterns (that is, not bound to specific assets) and evaluate them over a large period of time on off-label data. In particular, we propose a viable trading strategy that, once trained a NN on a large quantity of market historical data of hundreds of crypto assets, might automatically operate on the market to get positive profits. Our novel approach, differently from others, takes into account a single rich dataset of many different cryptocurrencies which turns out to be, as we show later, very effective compared to other approaches which take into account historical data of one or few assets at a time.

Cryptocurrency-related assets have seen a significant increase in market acceptance and have developed rapidly in recent years. As a result, many hedge funds and investors are beginning to include this type of asset in their financial portfolios, which has had a significant impact on the overall market and has generated considerable interest in trading algorithms for cryptocurrencies. According to Fang et al. (2022), the field of cryptocurrency trading research has experienced a significant surge in interest and activity in recent years. Specifically, the authors note that a staggering 85% of all the published scientific papers on algorithmic trading of cryptocurrency-related assets have been published just in the last five years. This suggests that the field of cryptocurrency trading research is rapidly evolving and that there is a growing interest in developing new strategies and approaches for trading cryptocurrencies.

While strategies applied to traditional financial markets can be adapted to the cryptocurrency market, they have certain unique characteristics that require new research efforts from the scientific community. Standard trading approaches are based on *fundamental* and *technical* analysis. The fundamental approach aims to determine whether an

* Corresponding author.
E-mail addresses: parente@unisa.it (M. Parente), lrizzuti@unisa.it (L. Rizzuti), mario.trerotola@polito.it (M. Trerotola).

https://doi.org/10.1016/j.eswa.2023.121806
Received 26 May 2023; Received in revised form 22 September 2023; Accepted 22 September 2023
Available online 28 September 2023

asset is currently traded at its fair market value using financial metrics and by conducting thorough research of the asset's business-related information. Technical analysis, on the other hand, analyzes price and volume series in order to predict an asset's future value, although this is founded on the assumption that future market prices can, in fact, be predicted.

According to the Efficient Market Hypothesis, Malkiel (2003), market agents are rational and new information is immediately reflected in the price, whereas the Adaptive Market Hypothesis (Chu et al., 2019) suggests that investors may behave irrationally in response to market volatility, thereby creating buying opportunities. This can be ascribed to human behavior, such as loss aversion, overconfidence and overreaction, which can increase market volatility in certain circumstances. The inefficiency of the major cryptocurrencies is investigated by Zhang et al. (2018) concluding that "results indicate that all these cryptocurrencies are inefficient markets". This makes the cryptocurrency market a fertile scenario for applying technical analysis using automated agents.

In fact, using algorithms for the trading might weaken these human behaviors. Another peculiarity of this scenario is that digital exchanges, and especially cryptocurrency exchanges, have lower fees than traditional brokers, which can make a significant difference in the high volume and frequencies of trades typical in this market. At present, a transaction fee on cryptocurrency exchanges can be as low as 0.1% per trade (at the time of writing, such fees are reduced to 0% on specific currency pairs). Additionally, most exchanges provide free trading API, thus further reducing barriers to entry for algorithmic trading.

1.1. Our contribution

We have developed a trading algorithm based on a Multi-Layer Perceptron (MLP) as a classifier with three classes, `Buy`, `Hold` and `Sell`. We designed a complete usual analysis pipeline: first we gathered price and volume time-series from a popular cryptocurrency exchange, then preprocessed them by feature extraction and labeling, and finally trained and tested the MLP. One of the peculiarities of our work is the dataset labeling algorithm, which is based on two thresholds to intercept significant market movements and two temporal windows, one in the past and one in the future for the forecasting of price trends.

A true trading strategy must be profitable in every phase of the market cycle: both in the bull market characterizing the past 10 years, in the bear/highly volatile market such as the one at the time of writing and in the flat one, like the period 2019–2021. Our findings show that a perfectly validated model with good performance standard metrics might perform poorly when used in a real simulation of a highly volatile market. To further validate our model, we set a simulation on a real scenario based on long term off-label historical data for different currencies (backtest phase).

We built a massive dataset of hundreds of cryptocurrencies spanning several years at a 4-h time resolution. The strategy was then evaluated over two different time intervals in order to assess its behavior on a long-term and a short-term real case scenario, which yielded very positive results in terms of Return On Investment (ROI). We then concluded the study with an analysis of the feature importance.

The Python source code developed for this research, datasets and output of the analysis can be found at Parente et al. (2023).

This paper is organized as follows: in Section 2 we review the relevant literature, analyzing it according to the different types of data sources used to predict cryptocurrency market trends. In Section 3 we show the preprocessing and feature extraction pipeline and the algorithm for data labeling. In Section 4 we describe the design of the MLP neural networks and rank these models according to the accuracy achieved in the testing phase. In Section 5 we report the results of the simulation. In Section 6 we describe the feature importance analysis conducted. Finally in Section 7 we give some conclusions and hints on future directions to be explored.

2. Related works

Forecasting the asset price or its direction calls for significant work on data integration and feature extraction from raw market data and/or other sources.

In Kraaijeveld and De Smedt (2020), Sattarov et al. (2020) and Valencia et al. (2019) sentiment analysis is employed on Twitter data feeds to predict the prices of major cryptocurrencies using standard ML models (Random Forest Regressor, SVM, MLP). Kim et al. (2016) uses messages from cryptocurrency-related web forums as primary data sources employing a probabilistic model based on Averaged One-Dependence Estimators (AODE).

In Guo et al. (2021) data are taken from blockchain, coin exchanges and Google Trends to predict Bitcoin price. The authors derived new features from the size of transactions and employed a proprietary model based on wavelet decomposition for preprocessing data and a Causal Multi-Head Attention Temporal Convolutional Network (WT-CATCN) for the modeling layer. Similarly, Li and Du (2023) encodes transactions found in the Bitcoin blockchain into a graph, searches for recurrent patterns, and correlates them with price changes using standard ML models (MLP, SVM). The authors in Saad et al. (2020) find that the Ethereum price is strongly positively correlated with various user activities on its blockchain, but they also identify an inverse correlation with the crude oil trend, probably due to the rise in energy costs.

Some research methods borrowed from traditional market methodologies seek correlations between cryptocurrencies and macroeconomic and/or financial indicators. Walther et al. (2019) evaluates macro and financial indexes to model the volatility of prices for five major cryptocurrencies by utilizing a GARCH-MIDAS framework (Engle et al., 2013). Similarly, Parvini et al. (2022) predicts the daily Bitcoin price by using data from other commodities and indexes, transforming them into the time–frequency domain using wavelet decomposition, and feeding the data into a Long Short-Term Memory (LSTM) network. Kim et al. (2021) also uses macroeconomic indexes and blockchain information to forecast the Ethereum price.

In addition to external data sources, some papers exploit the information in the order book. The order book contains the *limit orders* on both Buy and Sell sides, registered by the market participants. The orders placed by big market players, are often used as a source of information to predict the near-term direction of the asset's prices. Alec and Kercheval (2015), Tsantekidis et al. (2017) use Convolutional NN and SVM models respectively. Similarly, Guo et al. (2018) uses order book data to complement the price and volume information in order to extract features and predict prices.

Seasonality is another area of research borrowed from standard assets whose effects are evaluated on cryptocurrencies, see Baur et al. (2019) and Kaiser (2019).

Many papers rely on prices and volumes time series, basing the feature extraction on technical analysis or using techniques for time series forecasting/regression. In Lahmiri and Bekiros (2019) the daily prices of the 3 topmost cryptocurrencies are used to forecast the next day's prices with LSTM and Generalized Regression Neural Network. Similarly, Alonso-Monsalve et al. (2020) uses different neural network architectures to assess the predictability of trend direction in a minute time frame and finds that the hybrid LSTM with the Convolutional layer between the input and the Neural Network yields the best results.

The labeling of datasets for supervised learning is an open problem that is, broadly speaking, tackled using variations of two main ideas: price regression and price direction forecasting. In the former the differences lie in the size of the time frame used to assign a label (minutes, hours, days, etc.) and in the latter the difference is whether 2 or 3 labels are used, for `Buy/Sell` or `Buy/Hold/Sell`. In this paper we use three labels for the classification (as Kraaijeveld & De Smedt, 2020 and Tsantekidis et al., 2017).

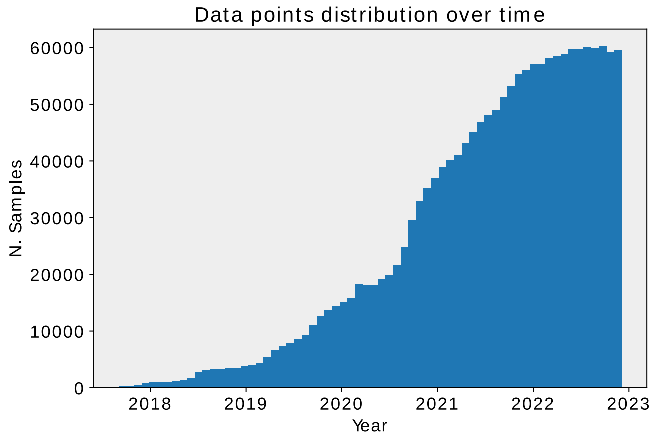


Fig. 1. Cryptocurrency samples distribution over time.

3. The dataset

In this section we describe the dataset used to apply our supervised ML algorithm. It consists of raw data gathered from the price and volume data time series of a popular cryptocurrency exchange. Then we show how the features were extracted from the raw data by computing candlestick patterns and financial indicators, and by moving average crossovers and temporal data. Finally, we design an algorithm to label the observations, parameterized in such a way that it can be used to maximize classification accuracy and the profitability of a given trading strategy.

3.1. Data collection

Data were downloaded from a popular crypto exchange platform exposing web based APIs and were collected in the usual *OHLC + volume* format, which encodes price variation in a given time frame in terms of *High, Open, Low, Close prices* and *Volume* exchanged during the given time frame. As usual *OHLC* is represented by "candlesticks" in the asset's price charts. As said in Section 1, we collected data in a single dataset consisting of 402 different crypto assets and set the time frame length at 4 h. The assets were all those available on the platform from August 17, 2017 to December 4, 2022. For those assets not available at the starting date, data were collected as soon as they were listed on the exchange. All the cryptocurrencies are those paired with the stable coin United States Dollar Tether (USDT). After the feature extraction phase, we obtained a dataset of 1.5 million samples with the distribution shown in Fig. 1.

3.2. Feature extraction

Extracting features from raw data is a delicate step in the standard ML pipeline process since it pertains to the field of making information of a phenomenon (social, physical, etc...) easily accessible to an ML model. An established method to summarize the trading market behaviors are the technical indicators. These, by using past prices, compute a quantitative estimation for the future price direction. The input of those indicators are past prices and volumes and they often use plain or exponential moving averages of the past prices and/or volumes (e.g MACD, CCI, ADX, …). The rationale behind our choice of technical indicators is to use simpler indicators (MA crossovers, Z-Scores) often used in computation of more complex indicators (MACD, CCI, ADX, …), and few other commonly used indicators, to let the NN discover useful patterns. We complemented these indicators with some temporal information and with the standard candlesticks patterns.

Candlestick patterns. Candlestick charts are typically used to visualize price fluctuations. A series of candlesticks may form a pattern and constitute a technical analysis tool that can suggest future price movements on the basis of past price behavior. By definition, the patterns are independent of the size of the time frame (Murphy, 1999). Our study considers 23 of the most popular candlestick patterns, both single and multiple candles, in their bullish and bearish version, e.g. Three Black Crows, Doji, Engulfing, Hammer. In this paper we use all the patterns mentioned in Pring (1991).

Technical indicators, known as oscillators, can assume values in a zero-centered interval and can be used as-is as features in ML applications. We use 6 of the most common technical indicators: Bollinger bands, ULTOSC, RSI, Close price percentage variation, Z-Score and volume Z-Score. Moreover, we also consider the Exponential Moving Average crossover so as to take into account the trends and trend reversals. See Kardile et al. (2021) and Murphy (1999) for more on technical indicators for financial markets.

- Bollinger bands. These are used to check whether prices are high or low on a relative basis. Given n periods, the price calculated using the n-period moving average is used as a reference price. Two lines are plotted above and below one standard deviation away from the reference price. The standard deviation of the price in the previous n-periods is used as a measure of volatility.
- RSI. The Relative Strength Index is a momentum indicator that measures the magnitude of recent price changes in order to evaluate overbought and oversold conditions. A lower RSI value indicates that the asset is oversold, while a higher value means that the asset is overbought.
- ULTOSC. The Ultimate Oscillator uses the values of three different moving averages with multiple time periods (or cycles), to identify overbought and oversold conditions in the market, thus improving the accuracy of the signals generated by the indicator.
- Close price percentage variation. This measures the percentage difference between the current price and the previous close price.
- Z-Score. This uses the z-score of the close price in a given number of time frames. In our implementation we used 30 close price past samples to compute the Z-Score of the actual time frame.
- Volume Z-score: In order to convert transaction volumes for comparing those of different cryptocurrencies, we used z-score normalization.
- EMA crossovers. The Exponential Moving Average crossovers are an established source of information for trend following and inversion. We considered 4 EMA crossovers based on 1 and 20, 20 and 50, 50 and 100, and 1 and 50 periods.

Temporal information. In Section 2 we noted that temporal information adds a statistical edge to price direction prediction. Thus, we have also added three other features based on time. Every single OHLC sample has an associated timestamp, used to extract the month of the year, the day of the week and the number of samples in the day (six a day). Summarizing, the overall feature vector has 36 entries: 23 candlestick patterns, 6 financial indicators, 4 EMA crossovers, and 3 temporal features.

3.3. Labeling algorithm

We propose the utilization of three distinct labels for classification purposes, analogously as in Kraaijeveld and De Smedt (2020) and Tsantekidis et al. (2017). The choice of using three labels aims to enhance the accuracy of the classification process, with respect to the use of only two, by addressing potential ambiguities that may arise when determining whether to open or close a position. By considering also the `Hold` label, we try to avoid confusion that may arise when deciding whether to refrain from opening a `Buy` position initially or to avoid triggering subsequent `Buy` orders after the same position has been opened. Furthermore, the use of the `Hold` label is a critical

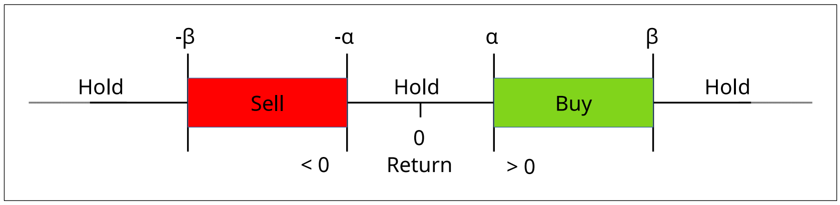


Fig. 2. Visual representation of label boundary based on the α and β parameters.

step towards aligning our classification algorithm to real-world trading practices.

The analysis we conducted uses a *supervised* ML model, thus each sample of the dataset has to be labeled for the model's training phase.

First let us define the temporal *windows* and the *return* for a single trade operation.

Definition 1. Given a time frame at time t and a cryptocurrency c, a **Forward Window** of size k, $\mathcal{FW}_{t,c}(k)$, is the sequence of time frames $t, t+1, \dots, t+k-1$ and a **Backward Window** of size k, $\mathcal{BW}_{t,c}(k)$, is the sequence of time frames $t-k+1, t-k+2, \dots, t$.

In the following we drop the subscripts t and c when the context is clear. Opening a position in the market at the open price of the time frame at time t, $Open_t$, and closing the position at the close price of the time frame $t+k$, $Close_{t+k}$, yields a revenue or a loss, formally defined as follows.

Definition 2. The **return** of the trade of a cryptocurrency c, opened at time t and closed at the time frame $t+k$, is given by:

$$\mathcal{R}_{t,c}(k) = \frac{(1-f)\cdot Close_{t+k} - (1+f)\cdot Open_t}{Open_t}$$

where f is the fee applied by the exchange for each trade operation.[1]

Here too we drop the subscripts t and c and retain only the duration k of the opened position, when clear from the context.

The labeling algorithm, depicted in Fig. 3, has two parameters, α and β, which are used to set the thresholds of the return values $\mathcal{R}(k)$, where k is the size of the *Forward* window. The former is used to establish a low value below which it is not convenient to place a trading order and the latter is a high value above which we consider a price variation not influenced by the technical framework but by exogenous forces. In this extreme scenario we likewise do not place orders. In summary, in order to trigger a `Buy`, `Sell` or `Hold` signal, given $Open_t$ we want to predict the price variation at the end of the Forward window $Close_{t+k}$. Fig. 2 depicts the behavior of the labeling algorithm as a function of the α and β parameters.

The algorithm in Fig. 3 initiates with five input parameters: the array of closing prices $closePs$, the Backward and Forward window sizes ($backW$ and $forW$ respectively), and the two threshold values α and β.

The first step of the algorithm is to update the close prices with their Exponential Moving Average (EMA), utilizing the $backW$ window size.

The algorithm then enters a loop on each such close price and computes its return. Next, the algorithm determines if the absolute value of the return falls within the range $]\alpha, \beta[$. If this condition is met, the algorithm then distinguishes between positive and negative returns. A positive return labels the time frame as `Buy`, while a negative return labels the time frame `Sell`.

On the other hand, if the condition is not met, the algorithm proceeds to the "Set label to `Hold`" block. This situation arises when the return value is either too small (less than α) or too large (greater than β), indicating that it is advisable to avoid making any transactions at that time.

[1] The fee is given here as a percentage of the investment and, for simplicity's sake, is assumed to be equal for both a buy and a sell operation.

Fig. 4 depicts a price chart. The rectangles represent the windows around a price sample, marked with an arrow. The *Backward* window is of size 5 and the *Forward* window is of size 2. The 5EMA of the close prices in the backward window is the starting point of a trend, shown as a dotted line, giving the price direction to compute the label of the current price sample.

4. MLP models, training and testing

In this section we introduce the classification model and the methodology applied for the training and the testing phase. We adopted the MLP model with four layers as a classifier, trained on the dataset labeled with the labeling algorithm reported in Fig. 3.

4.1. Multi-layer perceptron

The MLP consists of an input layer, two hidden densely connected layers and a categorical output layer with three nodes. When defining the MLP architecture in terms of the number of layers and neurons, we followed two guidelines: the universal approximation theorem (Hornik et al., 1989) and the principle that an MLP should have the least number of neurons to generalize well (Hunter et al., 2012). The universal approximation theorem ensures that a feedforward neural network with one input layer, one hidden layer and one output layer can approximate any function of the input with any desired degree of accuracy, provided that a sufficient number of hidden units are available.

Using a bottom-up approach, see e.g. Setiono (2001), we began with a small number of neurons on each hidden layer and compared the accuracy on the train and test set. On undersized networks, these two numbers are comparable up to the second digit after the decimal point. As we added neurons to each hidden layer, the network began to overfit on the training set. We selected the minimum number of neurons that exhibited slight overfitting and provided the highest accuracy on the test set. This approach allowed us to avoid managing overfitting with dropout layers or l1/l2 penalty terms on the weights of the network.

Specifically, the input layer has 128 nodes, the two hidden layers have 64 and 32 nodes, respectively, all with the LeakyReLU activation function. Finally, the output layer has three nodes with the Softmax activation function, which ensures a probabilistic output for the three classes.

4.2. Window sizes, α and β parameters

Forward and *Backward* windows, and α and β, described in Section 3.3, are the parameters and the thresholds used to label the dataset. Here we describe how these values are computed. The thresholds α and β emerge from a statistical analysis of the open-close percentage change in prices for the whole dataset. The value of α is set to the 85-th percentile and the value of β is set to the 99.7-th percentile. The absolute values of the thresholds are thus $\alpha = 0.038$ and $\beta = 0.24$. The β marking the outlier boundary, see Fig. 2, is incremented by 10% each time the size of the *Forward* window is increased. Thus, for windows of size 2, $\beta = 0.24 + 0.024$, for size 3, $\beta = 0.24 + 2 * 0.024$ and so on.

This choice of α leads to an unbalanced dataset towards the `Hold` class, which represented approximately the 70%, resulting in poor classification results. To overcome this, we used a random undersampling on the majority class to balance the dataset, see e.g. Buda et al. (2018).

Name	Description
$closePs$	Array of close prices
$backW$	Backward window size
$forW$	Forward window size
α	Low threshold
β	High threshold

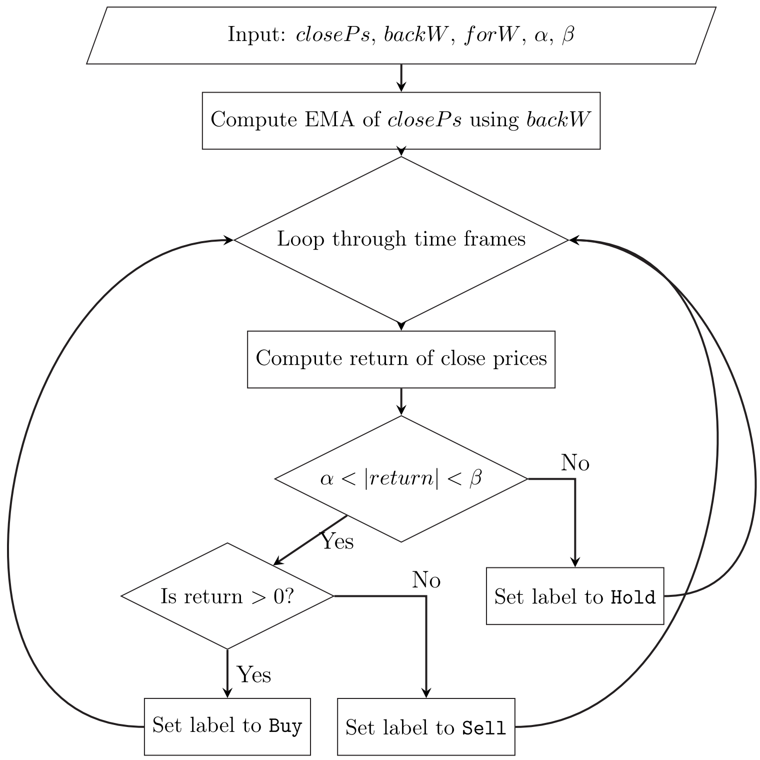


Fig. 3. Labeling algorithm flowchart.

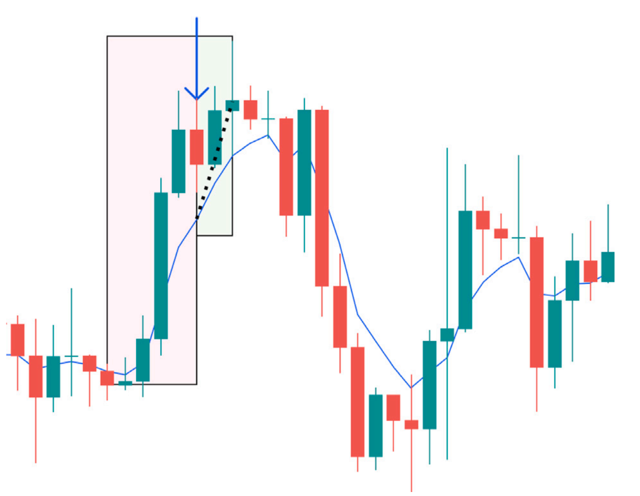


Fig. 4. A price chart with backward and forward windows of sizes 5 and 2, respectively, with a positive price variation.

Finally, to obtain the best combination of *Forward* and *Backward* window sizes, a grid search was conducted using the sizes from 1 to 5, hence considering 25 labeling schemes. The training and testing dataset was built with data on currencies from August 17, 2017 to December 31, 2021, and from April 1, 2022 to December 4, 2022 with the 25 different labeling schemes (the first three months of the year 2022 were later used to select the best model by profitability as described in Section 5).

Table 1
Top 5 models ranked by accuracy (third column) and the corresponding combination of Backward and Forward window sizes (first and second column).

$BackW$	$ForW$	Acc	Buy		Hold		Sell		Samples
			Pre	Rec	Pre	Rec	Pre	Rec	
5	1	0.72	0.71	0.75	0.67	0.59	0.76	0.81	463k
4	1	0.70	0.69	0.72	0.65	0.60	0.74	0.77	422k
3	1	0.67	0.65	0.67	0.65	0.58	0.69	0.74	379k
5	2	0.66	0.66	0.66	0.61	0.57	0.70	0.75	610k
4	2	0.64	0.64	0.64	0.61	0.55	0.68	0.73	582k

For each combination of *Forward* and *Backward* window sizes, we trained and tested the model three times, with a proportion of 70%–30%, using different random seeds and averaging the results. We then selected the top five accuracy value combinations of *Forward* and *Backward* windows as reported in Table 1. The different window lengths may lead to considerable variations in the size of the samples reported in the last column (due to the rebalancing of the majority class).

4.3. Comparisons with other models

Our choice to use MLP model has derived from a thoroughly experimentation of our dataset on others classifiers: XGBoost, Logit and SGDLinear. Here follows the analyses we have done on all these models.

Table 2
Model comparison for the combination of forW and backW whose accuracy is maximized: forW = 5 and backW = 1.

Model	Accuracy	Buy		Hold		Sell		Samples
		Prec	Rec	Prec	Rec	Prec	Rec	
MLP	0.72	0.71	0.75	0.67	0.59	0.76	0.81	463k
XGB	0.70	0.71	0.73	0.65	0.58	0.74	0.78	463k
Logit	0.64	0.67	0.72	0.52	0.41	0.69	0.77	463k
SGDLinear	0.59	0.58	0.84	0.60	0.03	0.60	0.87	463k

XGBoost. XGBoost is a supervised classification algorithm that employs an ensemble technique to construct a model. By combining predictions from multiple models, it aims to attain superior predictive performance. This model supports multi-class problems and exhibits high ability in solving non-linear problems. In our experimental setup, we utilize XGBoost, a variant of the Random Forest algorithm that incorporates a gradient boosting mechanism. During the learning process, XGBoost trains a random forest, but instead of aggregating trees, it utilizes gradient boosted trees that learn from errors at each boosting round. In the construction of each tree, a loss function with a regularization term, is optimized to maximize classification accuracy.

Logit. Logit is a binary classifier that learns a vector of weights $[w_1, \dots, w_n]$ and a bias term b and uses a logistic binary function to map the linear combination of weight vector and the new observations into a 0,1 classes.

SGDLinear. The SGDLinear binary classifier is a ML algorithm that learns a vector of weights by using the Stochastic Gradient Descent (SGD) technique, updating model parameters incrementally with individual data samples. It can handle large datasets efficiently and is suitable for online learning, making it a popular choice for various classification tasks. (Let us note that in our experiments, it outperforms the SVM classifier by many order of magnitude in terms of computational time.)

To compare the binary classifiers Logit and SGDLinear with our model which adopts three classes, we used One-vs-Rest strategy. A number N of binary classifiers were trained, one for each class vs the other two remaining classes. To predict a new observation, the N classifiers are queried and the values constitute the membership of the observation to the specific class, then $argmax(class1, class2, \dots, classN)$ is chosen as the predicted class.

We implemented the above models using a hyperparameters grid searching and have reported their performance in Table 2. The MLP and XGBoost performance overcome all linear models, with the former slightly better. MLP and XGBoost clearly excel at modeling highly non linear phenomena, differences in the performance metrics are small but constant during many runs with different seeds. MLP is clearly the best of the group, this has led to our choice. The linear models exhibit behaviors in line with other papers (Akyildirim et al., 2021; Jaquart et al., 2021; Ozer & Okan Sakar, 2022) in terms of accuracy, but the performances are still below those of MLP and XGB. It is worth noting that the recall of the `Hold` class for the SGDLinear model is surprisingly quite low.

5. Backtesting

In Section 4 we described the design and the training of an MLP and selected the top five combinations, in order of accuracy, of *Forward* and *Backward* window sizes used to label the dataset (see Table 1). Then we computed the profit on a simple trading strategy using the same data on these 5 models and we obtained a different ranking of the combination of window sizes. Hence, profitability is an evident and irrefutable metric when assessing algorithmic trading strategies (see e.g. Olorunnimbe & Viktor, 2023) and we were, therefore, induced to further investigate this situation. *Backtesting* is a standard method for evaluating ex-post the performance of an algorithmic trading strategy which uses profitability as a measure of the goodness of the model adopted. To simulate a real trading scenario, it uses historical data and does not open multiple long or short positions on the same asset simultaneously. At the end of the simulation period, all positions are closed and the profit is computed.

Table 3
Performance and metrics for the model selected for the backtest.

	Acc	Buy			Hold			Sell			Samples
		Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	
Test	0.66	0.65	0.68	0.67	0.65	0.52	0.57	0.70	0.78	0.73	201k
Train	0.68	0.66	0.70	0.68	0.66	0.53	0.58	0.71	0.80	0.75	469k
Dummy	0.33	0.32	0.32	0.32	0.32	0.32	0.32	0.36	0.36	0.36	201k

Table 4
MLP and dummy model backtesting generated ROI by different stop-losses. Models trained on labels generated with $backW = 5$ and $forW = 2$.

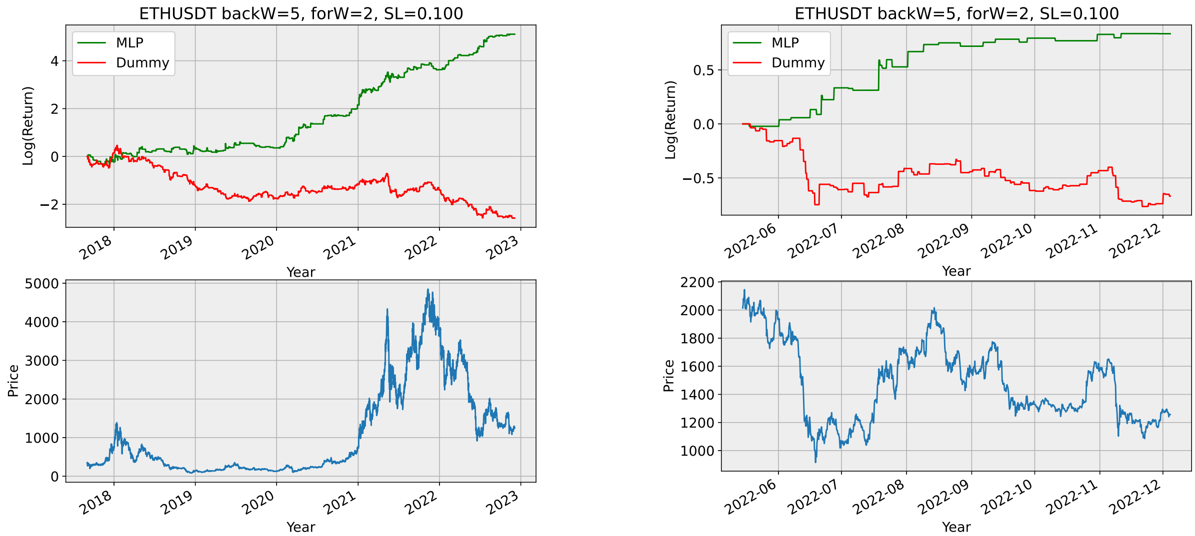
coin	ROIs				Stop loss
	MLP Long	MLP Short	Dum Long	Dum Short	
ALGOUSDT	0,22	0,89	0,27	0,80	0,00
BTCUSDT	0,23	0,84	0,06	0,69	0,00
ETHUSDT	0,08	0,76	0,08	0,69	0,00
ALGOUSDT	0,29	1,41	0,02	0,67	0,01
BTCUSDT	2,33	0,94	0,09	0,67	0,01
ETHUSDT	2,40	1,68	0,12	0,78	0,01
ALGOUSDT	0,65	1,44	0,03	0,57	0,025
BTCUSDT	17,76	0,82	0,15	0,67	0,025
ETHUSDT	24,18	1,76	0,21	0,69	0,025
ALGOUSDT	11,89	1,85	0,01	0,58	0,05
BTCUSDT	53,25	1,05	0,23	0,56	0,05
ETHUSDT	82,58	1,75	0,09	0,59	0,05
ALGOUSDT	36,18	2,08	0,01	0,47	0,10
BTCUSDT	61,22	1,18	0,14	0,50	0,10
ETHUSDT	165,91	2,30	0,08	0,51	0,10

The model to be used in backtesting was selected by means of a specially designed simulation algorithm implementing a standard strategy (see e.g. Pring, 1991): the algorithm scans the list of OHLC prices searching for the first `Buy`, which is the signal to enter the market, whereupon it places an order. The algorithm then looks for the next `Sell` label and the position is closed. The earning/loss is capitalized and the strategy restarts, thus obtaining compound interest. In our accounts we included the commission fees of 0.1% applied by some market brokers at the time of writing. The strategy used the data from the first three months of 2022, (not used in the training/testing phases, see Section 4.2). The period was chosen since the Bitcoin and Ethereum assets were following a downward trend (final price < initial price). The model with the windows sizes $(5, 2)$, the fourth in Table 1, obtained the greatest return, averaged over all the assets, and it was thus selected for the final backtesting. This model was then trained on the data from August 17, 2017 to December 4, 2022 for all the currencies apart from the data relative to *Bitcoin*, *Ethereum* and *Algorand* which were later used for backtesting.

The result of the test is reported in Table 3. The performance metrics are comparable to those in Table 1, even though the data for *Bitcoin*, *Ethereum* and *Algorand* are missing. We interpret this as an indicator of *good model generalization* on generic technical patterns not bound to specific coins.

The size of the dataset is different because of the differences in the composition of the datasets: those in Table 1 had all the coins except for 3 months of data for everyone, in Table 3 there are all coins except those 3 selected for the backtest.

Once the model had been selected, trained and tested, we extracted the features to finalize the backtesting for the above three currencies over 5 years and 4 months for *Bitcoin* and *Ethereum* and 3 years and 6 months for *Algorand*, and then queried the model. Trading strategies



(a) Long period backtest (b) Short period backtest

Fig. 5. ETHUSDT backtest.

can be described unambiguously using regular expressions on an alphabet composed of three symbols: `Hold`, `Buy`, `Sell`. In this way the trading strategy described above can be written as:

$$(\texttt{Buy[Buy|Hold]}^{*}\texttt{Sell})^{+} \tag{1}$$

where $*$ and $+$ are the transitive closures, see Hopcroft et al. (2006) for more on regular expressions.

To assess the quality of the equity curves, we compared our model with a baseline dummy classifier that randomly responds with the same distribution of labels in the unbalanced original dataset, where the `Hold`, `Buy`, `Sell` labels are approximately in the 70, 15, 15 proportion, respectively. The overall return R for n trades on the coin c, is then computed as follows:

$$R = \prod_{i=1}^{n}(1 + \mathcal{R}_c^i(2)) \tag{2}$$

where: n is the number of trades executed and $\mathcal{R}_c^i(2)$ is the return of a single trade computed according to Definition 2 on a *Forward* window of size 2.

Table 4 reports the ROIs of all the periods for every single backtest, both for the MLP and the Dummy models, for 5 different stop-loss thresholds, computed as:

$$ROI = \frac{FinalValueofInvestment - InitalValueofInvestment}{InitalValueofInvestment} \tag{3}$$

To better simulate a real scenario, a safeguard stop-loss has been implemented. Precisely, we have experimented five values of stop-loss, ranging from 0% to 10%, see Table 4. Wide stop-losses show a better performance than narrow ones, thus in the following we consider the 10% stop-loss threshold.

The ROIs are largely positive: the peak is reached for *Ethereum* in the long period with a ROI of 165.91. We have highlighted the whole period (called *long*) and a period ranging from May 15 to December 4, 2022, called *short*. We chose the latter interval to better assess the model under heavily volatile market conditions impacted by two exogenous events. The first of these occurred in June 2022: the TerraUSD-Classic (USTC) lost its Dollar peg, wiping out about $500 billion of market capitalization (see Briola et al., 2023 for a detailed description of the event). The second crash occurred in November 2022 and involved a popular cryptocurrency exchange: FTX. This constitutes a good example of an exogenous-force-driven crash; Jalan and Matkovskyy (2023) concludes that the major fault lay with the management and not in the crypto environment.

Table 5
MLP and dummy model backtesting number of transactions and max return.

Period	Coin	MLP Max Ret	Dummy Max Ret	MLP trs	Dummy trs
long	ALGO	0.54	−0.99	376	643
short	ALGO	0.18	−0.42	36	110
long	BTC	0.29	−0.77	246	893
short	BTC	0.16	−0.44	28	106
long	ETH	0.76	−0.91	379	930
short	ETH	0.26	−0.41	37	111

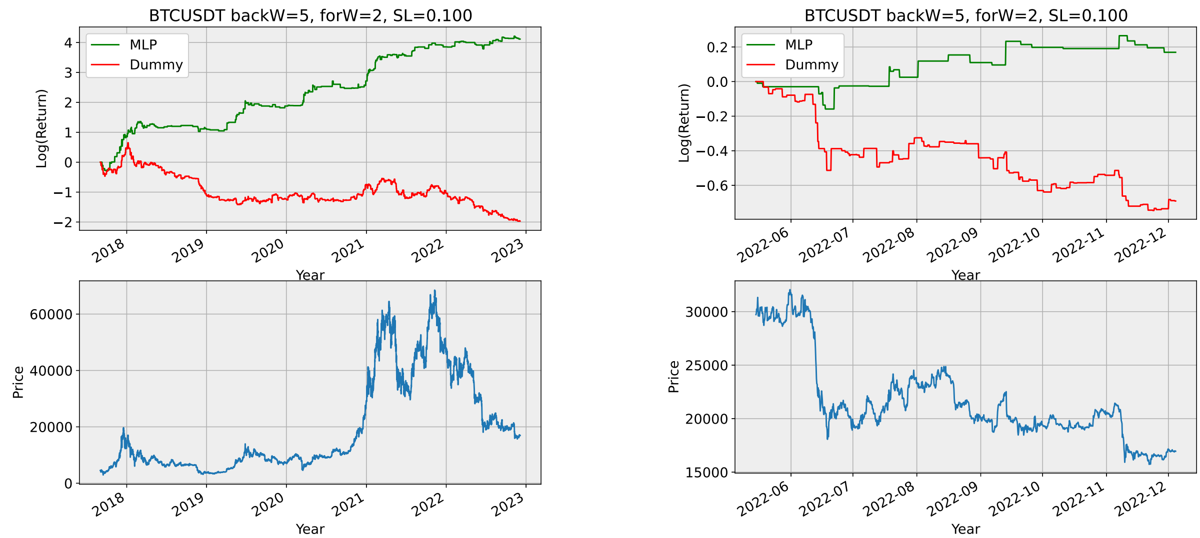
In order to acquire a complete overview of the trading, we also reported the number of trades and the maximum return per trade obtained in the period in Table 5.

As expected, the Dummy model returns slowly decrease to zero, yielding negative ROIs, reported in the figures in logarithmic scale. The long backtest shown in Figs. 5(a), 6(a) and 7(a) depicts a curve that is typical of a trading strategy with a statistical edge over a random one: a slow but stable increase in the return.

The short backtests in Figs. 5(b), 6(b) and 7(b) during the crashes expose the behavior expected from a strategy, mostly based on lagged indicators. In the next section will illustrate how lagged indicators determine most of the output of the MLP classifier. The initial Terra-Luna crash is smoothed by the MLP model in comparison to the Dummy model. The same behavior can be inferred from the FTX crash, near the end of the charts.

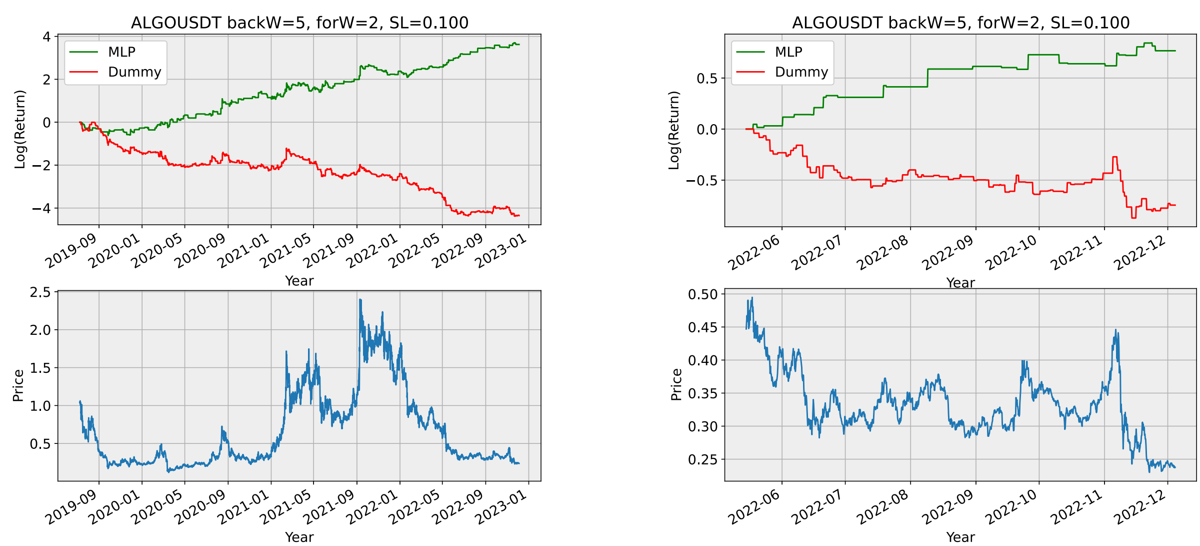
5.1. Comparisons with recent papers

Comparing methodologies on price forecasting and trading strategies is a complex task. A fair comparison of different results requires to have some invariants in the experiment setup. To compare models from different methodologies a not empty intersection of the datasets is required. This way researchers can use shared data to evaluate models using the same metrics. This is on the data science side. On the financial side the same problem arises because of the added step of backtesting, since this requires a shared period of the data in common, at the same (or similar) time sampling interval, to be used as evaluation set. Just to mention, the model is a component of a trading strategy: it acts as an oracle that guesses the future price, but the final decisions is up to



(a) Long period backtest   (b) Short period backtest

Fig. 6. BTCUSDT backtest.



(a) Long period backtest   (b) Short period backtest

Fig. 7. ALGOUSDT backtest.

the trading strategy. It uses information like commission fees, analysis of previous model prediction, (e.g. buy only after two buy signals in a row), money management. Finally often there is a difficulty to acquire information to reproduce the tests, since the source code is missing.

In what follows we compare our work with five recent papers (Akyildirim et al., 2021; Alonso-Monsalve et al., 2020; Cavalli & Amoretti, 2021; Jaquart et al., 2021; Ozer & Okan Sakar, 2022) and in Tables 6 and 7 we report a summary. Each paper uses a 2 classes labeling and a timeframe of 1 min, or 1, 4, and 24 h, spanning over different periods of time. The data sources are OHLC+V for all of them, plus some other sources, like Twitter (Tw), Blockchain (Bch) and others financial indices. We have chosen their best accuracy to compare with ours. For papers that backtest their models, to compare ROIs, we wrote specific code that apply our own strategy to the common coins on the same period. We reported the results o backtest comparisons in Table 7.

In the paper (Alonso-Monsalve et al., 2020) the price direction for six popular coins (Bitcoin, Dash, Ether, Litecoin, Monero, and Ripple) is forecasted, spanning one year (third quarter of 2018 to second quarter of 2019) at a time frame of 1 min. They use technical indicators and moving averages as features to feed different neural networks: CLSTM, MLP, CNN, RBFNN. To compare their metrics with ours, since they report for each model the accuracy of the prediction on each coin, we have computed the mean accuracy of the best model, the CLSTM, for the six coins.

The paper (Ozer & Okan Sakar, 2022) forecasts price direction for Bitcoin, Ethereum and Litecoin. Data belongs to the period between July 1, 2017 and April 30, 2021. The features have been determined from technical indicators. The authors use 5 different classical models. We report the accuracy of the logistic classifier, which is their best average accuracy computed. A backtest was conducted on all of the three coins at 4H and 1D time frame. The timespan and the coin set has a non empty intersection with ours, and this allowed us to compare the results of BTC and ETH. The period spans from the beginning of May 2020 to the end of April 2021, a period with a sideway trend that

Table 6
Comparing post 2020 similar works by accuracy.

Ref	Coins	Model	Acc	TF	Period	Source
Alonso-Monsalve et al. (2020)	6 Topmost	CLSTM	68.17	1M	1 Year	OHLC+V
Ozer and Okan Sakar (2022)	BTC, ETH, LTC	Various	55.9	4H	4 Years	OHLC+V
Jaquart et al. (2021)	BTC	LSTM	56	1H	9 Month	OHLC+V, others
Akyildirim et al. (2021)	12 Topmost	SVM	53	1H	5 Year	OHLC+V
Cavalli and Amoretti (2021)	BTC	CNN	74.2	1D	7 Years	Price, Tw, Bch
Our paper	402 coins	MLP	72	4H	5 Year	OHLC+V

Table 7
Comparison of ROIs.

Coin	Our paper	Ozer and Okan Sakar (2022)	Period	Fee
Btc	39	63	May 2020–Apr 2021	0.1%
Eth	66.5	106	May 2020–Apr 2021	0.1%
Coin	Our paper	Jaquart et al. (2021)	Period	Fee
Btc	−0.6	Negative	Set–Dic 2019	30 Bps – 0.3%
Coin	Our paper	Cavalli and Amoretti (2021)	Period	Fee
Btc	99.6	96.1	Feb–Oct 2020	no fee
Btc	−4.4	−28.3	Feb–Mar 2020	no fee

ends with a strong bullish move. The backtest was done with buy-only strategy, like ours. The main difference in the experiment is the number of trades, our system trades at a lower frequency. Nothing can be said to the edge of the strategy in a bearish market, as the trend in the period is sideway with a short bullish in the last months.

The paper (Jaquart et al., 2021) predicts Bitcoin price direction. Data span from March 4, 2019 to December 10, 2019 and use Twitter sentiments, blockchain transactions and a variety of minute level prices for commodities and indices: Bitcoin, gold, oil, the indices MSCI World, S&P 500, and VIX. The models used are neural networks and decision trees, plain or boosted ensemble. The accuracy is reported for each model for 1, 5, 15 and 60 min time frame. In Table 6 we reported the accuracy of the best model in the 60 min time frame. The authors did a backtest from September to December 2019 and exhibit a ROI on the best model (LSTM) of 115% but without commission fees applied. The Bitcoin trend in this interval is bearish and it is stated that by applying fee at 0.3% turns the ROI negative, without quantifying the loss. The peculiarity of the work is the very high frequency trading strategy, which executes 2852 trades during the backtesting, (ours just 4). We can try to estimate the loss by supposing that no compounding is applied and 0.1% fees, the transactions reported sum to a 258% loss due to the fees plus the earning or loss for the trades.

In Akyildirim et al. (2021) the authors predict price direction of the 12 most popular cryptocurrencies (BTC, BCH, DSH EOS, ETC, ETH, IOT, LTC, OMG, XMR, XRP, ZEC), whose data span from August 10, 2017 to June 23, 2018. The dataset is composed by OHLC+V data at different time frames along with trading data taken from a popular broker. The features are computed by OHLC+V and pure traded data and then by transforming them using technical indicators, moving averages and log returns for different time intervals. The labels are computed by $Signum(Close-Open)$. The accuracy reported in the paper is the average for the 1H time frame, which is also close to the best accuracy (low variance) on the test set.

The paper (Cavalli & Amoretti, 2021) forecasts price direction for Bitcoin. Data span from April 28, 2013 to February 15, 2020. The feature set uses data from market prices, Twitter sentiment and blockchain. The model adopted is a 1D Convolutional Neural Network. The backtest is arranged in two different runs, the first one from February to October 2020 (bullish) and the second one from February to March 2020 (a short period downtrend). No commission fees are applied to the transactions. In the first run the performance values are similar to ours, though in a slight advantage. In the second run the system registered a loss of 28% (without commission fees), whereas our system exhibits a loss of 4.4% (with fees).

6. Feature importance

The field of interpretability/explainability in ML pertains to the comprehension of a model's behavior in predicting outcomes by mapping input to output. It is a continuously evolving area of study that aims to enhance the transparency of ML models (Doshi-Velez & Kim, 2017). Despite some exceptions (e.g. linear models, decision trees, random forests), where explanation of the model's behavior is accessible in terms of how much a feature determines the *global* model behavior, Molnar (2022), MLP models have no direct way for their interpretation and are often treated as a black box oracle. Global model behavior provides a general understanding of how the model works: which features are most important, and how they interact with each other. Global explanations are usually computed based on the analysis of multiple instances and are more suitable for generalizing the behavior of the model.

SHAP (SHapley Additive exPlanations), Lundberg and Lee (2017), is a game theoretic approach to explain the output of any black-box ML model. SHAP computes a value for the feature importance attribution, which is an implementation of the popular Shapley Values (Shapley, 1953). It is used to solve an attribution problem, by distributing the prediction of a model for a specific input to its base set of features and showing how influential a feature is in making a decision. Clearly, this approach provides a local explanation for each specific input of the model. In the implementation, we encoded the label `Buy` with −1, `Hold` with 0 and `Sell` with 1. SHAP values are in the range of the classifier output, and thus the values are in the $]-1,1[$ range although they may sometimes assume values slightly outside the range boundaries.

Overall feature importance can be inferred by computing the mean of SHAP absolute values for each feature, as shown in Fig. 8.

The chart in Fig. 9 shows the top-10 contributions of each feature to each prediction. More precisely this chart can be used to spot correlations between feature values and SHAP values using colors. For example, the Bollinger feature has a clear correlation with the output of the classifier, increasing the Bollinger value (on the vertical axis) and the output steers from `Sell` to `Buy` (on the horizontal axis). It is counterintuitive since in the trading literature a higher value of the Bollinger indicator means trend inversion. However, the model learned Bollinger values as a trend continuation feature. EmaCross1_21 and RSI, display a similar behavior. EmaCross21_50, instead, shows an inverse correlation with its SHAP value. The remaining features do not exhibit a behavior that can be correlated with SHAP values.

As a local explanation, the SHAP values may be used to ask how a specific input can be explained in terms of feature values, or in other

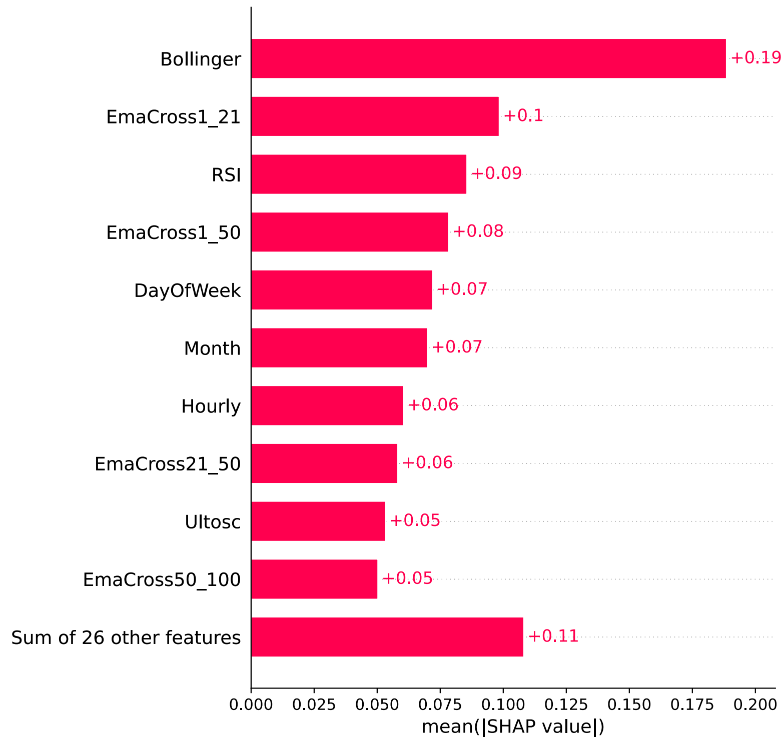


Fig. 8. Top-10 feature mean contributions.

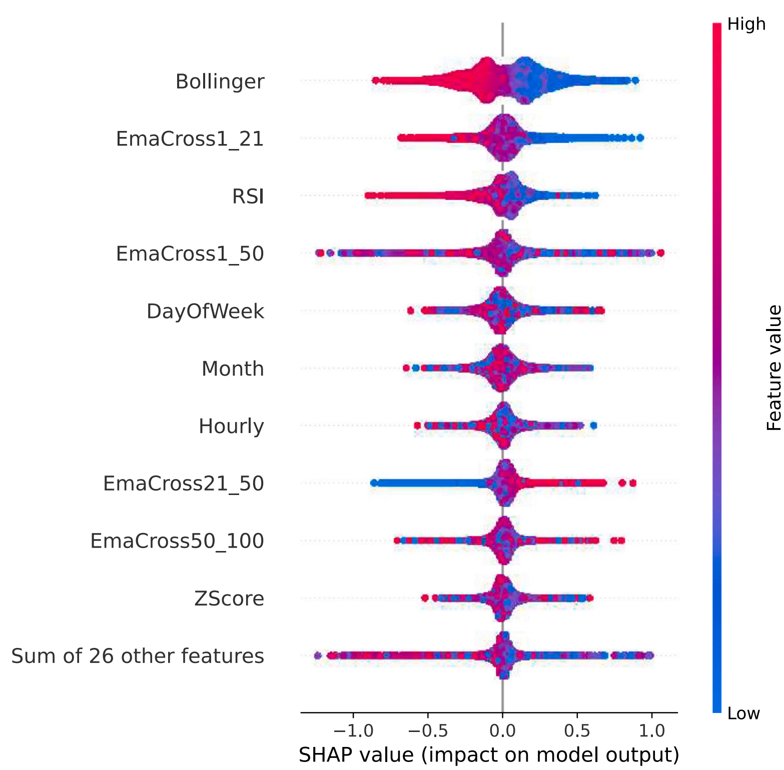


Fig. 9. Feature impact on model output.

words, how a specific feature value steers the output of the classifier into the prediction. Waterfall plots achieve this by charting the contribution of each feature to the prediction. The chart in Fig. 10 shows the SHAP values for the ten most important features. The vertical axis lists the features and their standardized values, while the horizontal axis reports the classifier output. The arrows show the extent to which every feature contributes to the prediction.

The impact of the remaining 26 features on the output is almost 0. In Fig. 11, the output strays a few hundredths from the expected value 0 for the `Hold` label.

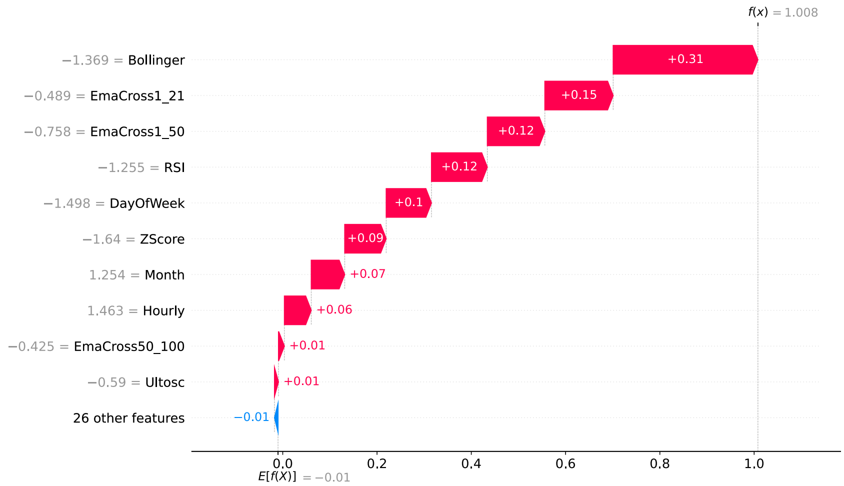


Fig. 10. Prediction of an input vector in terms of top-10 features whose standardized values are on the left.

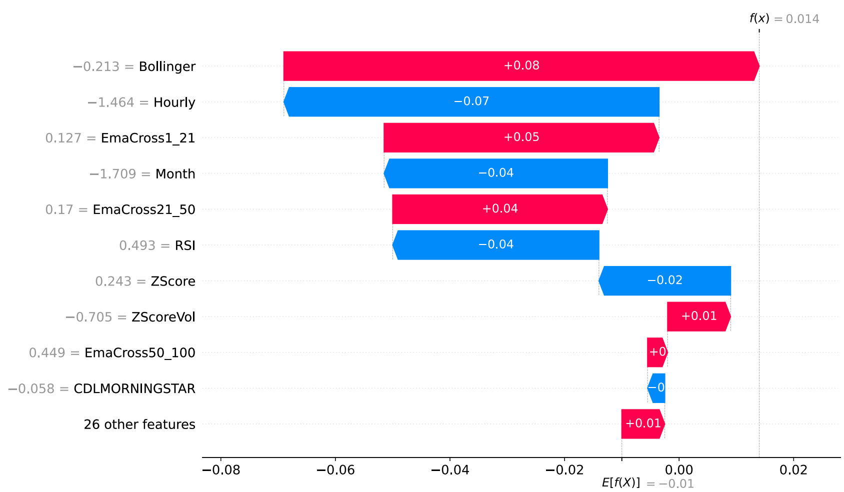


Fig. 11. Prediction of an input vector in terms of top-10 features whose standardized values are on the left.

In conclusion, the investigation of the feature importance indicates that the top-10 most valuable features are comprised solely of technical indicators, moving average crossovers and temporal information. Candlestick patterns are seen to be relatively ineffective due to their infrequency or absence in some cases, and their ability to provide relevant information for determining the output of the classifier is limited.

7. Conclusion

The trading automation poses various challenges related to the feature extraction and labeling process, as well as the practical implementation of predictive models in real or simulated market settings. Prediction of the trend direction of a blockchain-based asset price presents multiple technical complexities due to poor regulation by authorities and market manipulations resulting in the pumping and dumping of selected assets (Eigelshoven & Ullrich, 2021; Fratrič P. Sileno et al., 2022; Gandal et al., 2018) which leads to high volatility. Higher price fluctuations imply higher profit opportunities, but these profit opportunities are not easy to spot. The proposed approach reformulates the traditional forecasting model as a classification process. In particular, we have shown that a simpler approach supported by a massive dataset and smart labeling can achieve a comparable or better accuracy than more complex models. For example, a recent work (Ozer & Okan Sakar, 2022) achieves a 56% accuracy on a binary labeled scheme. Under the assumption of a balanced dataset, this is only 6% over a random baseline. Whereas we have achieved a 66% accuracy, which is an excellent performance compared to a 33% baseline with a 3-classes problem.

We have developed a comprehensive pipeline for predicting short-term price trends and leveraging them to generate profitable trading strategies. Our findings reveal that the accuracy of price trend predictions is highly dependent on the use of a large dataset that enables the identification of patterns via technical analysis tools. We have shown that our trading system is profitable in every market conditions: bull, flat and in the more difficult bear/high volatile ones.

Moreover, we have also implemented a backtesting process proving that a technical analysis approach can provide a clear statistical edge over random trading operations or a simple “buy and hold” approach. In fact the backtest phase has shown a desirable characteristic in a trading system: the low volatility of the equity curves. Furthermore, by employing various stop-loss thresholds, it is possible to fine-tune the risk profile showing thus that high thresholds corroborate the well-established principle that higher risks are typically associated with higher rewards. It is important to note that our results do not guarantee that the same level of performance can be replicated in different time periods or with different coins. Moreover real market setups have to cope with liquidity issues that minor coins often experience, which may unpredictably and adversely impact the system’s performance.

Based on the analysis of feature importance, it has been determined that the top ten valuable features exclusively consist of technical indicators, moving average crossovers, and temporal information, with candlestick patterns showing a relatively low level of effectiveness.

These results lay the groundwork for future research endeavors that aim to augment the existing pipeline with additional information gleaned from multi-timeframe price action analysis. This approach can be accomplished by combining the pipeline's key components, such as the large dataset, labeling algorithm, best features and neural network, with novel information derived from a detailed analysis of price behavior across multiple timeframes.

Furthermore, technical analysis has ample room to explore with the support of ML techniques. Technical indicators are just one tool, other sources of information used by traders to spot price directions comprises technical patterns, supports and resistances (fixed and dynamic), Fibonacci levels, cross assets analysis.

As a prospective avenue for future research, it is worthwhile to subject our trading system to empirical testing across diverse financial markets, including FOREX, individual stocks, stock indexes, commodities, and CFDs (Contracts for Difference). This endeavor is based on the premise that these markets exhibit congruence in terms of adhering to similar technical patterns.

Nonetheless, prudent consideration must be given to CFDs, as they represent leveraged derivatives characterized by elevated risk exposure. While CFDs generally exhibit substantial correlation with the prices of underlying securities, this association is not always exact. The notable feature of CFDs lies in their requirement for a comparatively small amount of capital (via margin trading) to facilitate trading activities. This aspect significantly reduces the entry threshold for implementing the trading system in actual financial markets, making the exploration of real-world experimentation a feasible pursuit.

CRediT authorship contribution statement

Mimmo Parente: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing. **Luca Rizzuti:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Mario Trerotola:** Conceptualization, Data curation, Investigation, Initial writing of software, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data are publicly available at the url indicated in the bibliography.

References

Akyildirim, E., Goncu, A., & Sensoy, A. (2021). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*, *297*, 3–36.

Alec, N., & Kercheval, Y. Z. (2015). Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, *15*, 1315–1329.

Alonso-Monsalve, S., Suárez-Cetrulo, A. L., Cervantes, A., & Quintana, D. (2020). Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. *Expert Systems with Applications*, *149*, Article 113250.

Baur, D. G., Cahill, D., Godfrey, K., & (Frank) Liu, Z. (2019). Bitcoin time-of-day, day-of-week and month-of-year effects in returns and trading volume. *Finance Research Letters*, *31*, 78–92.

Briola, A., Vidal-Tomás, D., Wang, Y., & Aste, T. (2023). Anatomy of a stablecoin's failure: The Terra-Luna case. *Finance Research Letters*, *51*, Article 103358.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249–259.

Cavalli, S., & Amoretti, M. (2021). Cnn-based multivariate data analysis for bitcoin trend prediction. *Applied Soft Computing*, *101*, Article 107065.

Chu, J., Zhang, Y., & Chan, S. (2019). The adaptive market hypothesis in the high frequency cryptocurrency market. *International Review of Financial Analysis*, *64*, 221–231.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv:1702.08608.

Eigelshoven, F., & Ullrich, A. (2021). Cryptocurrency market manipulation: A systematic literature review. In *International conference on information systems* (p. 1225).

Engle, R. F., Ghysels, E., & Sohn, B. (2013). Stock market volatility and macroeconomic fundamentals. *The Review of Economics and Statistics*, *95*, 776–797.

Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., & Li, L. (2022). Cryptocurrency trading: A comprehensive survey. *Financial Innovation*, *8*, 1–59.

Fratrič P. Sileno, G., Klous, S., & van Engers, T. (2022). Manipulation of the bitcoin market: an agent-based study. *Financial Innovation*, *8*, 60.

Gandal, N., Hamrick, J., Moore, T., & Oberman, T. (2018). Price manipulation in the bitcoin ecosystem. *Journal of Monetary Economics*, *95*, 86–96.

Gerlein, E. A., McGinnity, M., Belatreche, A., & Coleman, S. (2016). Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, *54*, 193–207.

Guo, T., Bifet, A., & Antulov-Fantulin, N. (2018). Bitcoin volatility forecasting with a glimpse into buy and sell orders. In *2018 IEEE international conference on data mining* (pp. 989–994).

Guo, H., Zhang, D., Liu, S., Wang, L., & Ding, Y. (2021). Bitcoin price forecasting: A perspective of underlying blockchain transactions. *Decision Support Systems*, *151*, Article 113650.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). *Introduction to automata theory, languages, and computation* (3rd ed.). USA: Addison-Wesley Longman Publishing Co. Inc..

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359–366.

Hunter, D., Yu, H., Pukish, . M. S., III, Kolbusz, J., & Wilamowski, B. M. (2012). Selection of proper neural network sizes and architectures–A comparative study. *IEEE Transactions on Industrial Informatics*, *8*, 228–240.

Jalan, A., & Matkovskyy, R. (2023). Systemic risks in the cryptocurrency market: Evidence from the ftx collapse. *Finance Research Letters*, *53*, Article 103670.

Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. *The Journal of Finance and Data Science*, *7*, 45–66.

Kaiser, L. (2019). Seasonality in cryptocurrencies. *Finance Research Letters*, *31*.

Kardile, R., Ugale, T., & Mohanty, S. N. (2021). Stock price predictions using crossover sma. In *2021 9th international conference on reliability, Infocom technologies and optimization (trends and future directions)* (pp. 1–5).

Kim, H. M., Bock, G. W., & Lee, G. (2021). Predicting ethereum prices with machine learning based on blockchain information. *Expert Systems with Applications*, *184*, Article 115480.

Kim, Y. B., Kim, J. G., Kim, W., Im, J. H., Kim, T. H., Kang, S. J., & Kim, C. H. (2016). Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PLoS One*, *11*, 1–17.

Kraaijeveld, O., & De Smedt, J. (2020). The predictive power of public twitter sentiment for forecasting cryptocurrency prices. *Journal of International Financial Markets, Institutions and Money*, *65*, Article 101188.

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, *197*, Article 116659.

Lahmiri, S., & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals*, *118*, 35–40.

Li, X., & Du, L. (2023). Bitcoin daily price prediction through understanding blockchain transaction pattern with machine learning methods. *Journal of Combinatorial Optimization*, *45*, 1–24.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *30*.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, *17*, 59–82.

Molnar, C. (2022). *Interpretable machine learning* (2nd ed.). CM.

Murphy, J. J. (1999). Technical analysis of the financial markets: A comprehensive guide to trading methods and applications. *Penguin*.

Olorunnimbe, K., & Viktor, H. (2023). Deep learning in the stock market—A systematic survey of practice, backtesting, and applications. *Artificial Intelligence Review*, *56*, 2057–2109.

Ozer, F., & Okan Sakar, C. (2022). An automated cryptocurrency trading system based on the detection of unusual price movements with a time-series clustering-based approach. *Expert Systems with Applications*, *200*, Article 117017.

Parente, M., Rizzuti, L., & Trerotola, M. (2023). Cryptotrading.zip. http://dx.doi.org/10.6084/m9.figshare.22953377.v1, Figshare URL: https://figshare.com/articles/software/CryptoTrading_zip/22953377.

Parvini, N., Abdollahi, M., Seifollahi, S., & Ahmadian, D. (2022). Forecasting bitcoin returns with long short-term memory networks and wavelet decomposition: A comparison of several market determinants. *Applied Soft Computing*, *121*, Article 108707.

Pring, M. (1991). *Technical analysis explained: the successful investor's guide to spotting investment trends and turning points*. McGraw-Hill.

Saad, M., Choi, J., Nyang, D., Kim, J., & Mohaisen, A. (2020). Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions. *IEEE Systems Journal*, *14*, 321–332.

Sattarov, O., Jeon, H. S., Oh, R., & Lee, J. D. (2020). Forecasting bitcoin price fluctuation by twitter sentiment analysis. In *2020 international conference on information science and communications technologies* (pp. 1–4).

Setiono, R. (2001). Feedforward neural network construction using cross validation. *Neural Computation*, *13*, 2865–2877.

Shapley, L. S. (1953). A value for n-person games. In H. W. Kuhn, & A. W. Tucker (Eds.), *Contributions to the theory of games II* (pp. 307–317). Princeton: Princeton University Press.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2017). Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th conference on business informatics* (pp. 7–12).

Valencia, F., Gómez-Espinosa, A., & Valdés-Aguirre, B. (2019). Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. *Entropy*, *21*.

Walther, T., Klein, T., & Bouri, E. (2019). Exogenous drivers of bitcoin and cryptocurrency volatility –A mixed data sampling approach to forecasting. *Journal of International Financial Markets, Institutions and Money*, *63*, Article 101133.

Zhang, W., Wang, P., Li, X., & Shen, D. (2018). The inefficiency of cryptocurrency and its cross-correlation with dow jones industrial average. *Physica A. Statistical Mechanics and its Applications*, *510*, 658–670.