Applied Soft Computing Journal 90 (2020) 106187

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Ensemble of machine learning algorithms for cryptocurrency investment with different data resampling methods

Tomé Almeida Borges[1], Rui Ferreira Neves *,[1]

Instituto de Telecomunicações, Instituto Superior Técnico, Torre Norte, Av. Rovisco Pais, 1, Lisboa 1049-001, Portugal

ARTICLE INFO

Article history:
Received 3 July 2019
Received in revised form 10 January 2020
Accepted 19 February 2020
Available online 26 February 2020

Keywords:
Financial markets
Cryptocurrencies
Technical analysis
Machine learning
Ensemble classification
Financial data resampling

ABSTRACT

This work proposes a system based on machine learning aimed at creating an investment strategy capable of trading on the cryptocurrency exchange markets. Additionally, with the goal of generating investments with higher returns and lower risk, rather than investing on predictions based on time sampled financial series, a novel method for resampling financial series was developed and employed in this work. For this purpose, the originally time sampled financial series are resampled according to a closing value threshold, thus creating a series prone to obtaining higher returns and lower risk than the original series. Out of these resampled series as well as the original, technical indicators are calculated and fed as inputs to four machine learning algorithms: Logistic Regression, Random Forest, Support Vector Classifier, and Gradient Tree Boosting. Each of these algorithms is responsible for generating a transaction signal. Afterwards, a fifth transaction signal is generated by simply calculating the unweighted average of the four trading signals outputted from the previous algorithms, to improve on their results. In the end, the investment results obtained with the resampled series are compared to the commonly utilized fixed time interval sampling. This work demonstrates that independently of using or not a resampling method, all learning algorithms outperform the Buy and Hold (B&H) strategy in the overwhelming majority of the 100 markets tested. Nevertheless, out of the learning algorithms, the unweighted average obtains the best overall results, namely accuracies up to 59.26% for time resampled series. But most importantly, it is concluded that both alternative resampling methods tested are capable of generating far greater returns and with lower risk relatively to time resampled data.

1. Introduction

As a result of the tremendous growth and remarkably high market capitalizations reached, cryptocurrencies are now emerging as a new financial instrument and their popularity has recently skyrocketed. Despite being relatively recent, with the rise of Bitcoin, the cryptocurrency exchange market has become a global phenomenon, particularly known for its volatility and diversity, attracting the attention of many new and old investors [1].

Financial time series forecasting is a challenging task as these series are characterized by non-stationarity, heteroscedasticity, discontinuities, outliers and high-frequency multi-polynomial components making the prediction of market movements quite complex [2]. The complex characteristics of financial time series and the immense volumes of data that must be analysed to successfully accomplish the task of forecasting financial time series have driven the adoption of more sophisticated methods, models and simulation techniques. Lately, machine learning or data mining techniques, widely applied in forecasting financial markets, have been offering improved results relatively to simple technical or fundamental analysis strategies. Machine learning methodologies are able of uncovering patterns and predict future trends in order to identify the best entry and exit points in a financial time series with the intention of achieving the highest returns with the lowest risk [3].

In this work a major objective is using technical indicators as input data to forecast, better than random guessing, the dichotomous event: will a specific currency pair be bullish or otherwise (bearish or sideways) in the next instant of a time series. To achieve this goal, several supervised machine learning classification approaches are suggested. This concept has been widely used in diverse financial markets, such as stocks, bonds or Foreign Exchange markets [4]. However, this work focuses on the cryptocurrency exchange market.

The main contributions of this paper are: development of a framework consisting of several supervised machine learning

* Corresponding author.
E-mail addresses: rui.neves@tecnico.ulisboa.pt, rui.neves@ist.utl.pt, rfmfn@lx.it.pt (R.F. Neves).

[1] All authors contributed equally to all sections.

https://doi.org/10.1016/j.asoc.2020.106187
1568-4946/

procedures to trade in a relatively new market, the Cryptocurrencies Market; compare the performance of 5 different methods for forecasting trading signals amongst themselves and with a B&H strategy as baseline; lastly, the main contribution of this paper is the development of an innovative procedure for resampling financial time series with the intention of obtaining improved results. In this paper, the financial series resampled according to a parameter derived from trading activity, in particular a variation percentage as well as a fixed variation, are compared in terms of profitability, risk and predictability to the commonly used time sampled financial series as baseline.

This paper is organized as follows: in Section 2 the fundamental concepts and related work are discussed; Section 3 documents the entire proposed system architecture; In Section 4 the case studies and results are presented and analysed; Section 5 provides the conclusions to the work developed.

2. Background & State-of-the-Art

In the year 2008, in a white paper called: "Bitcoin: A Peer-to-Peer Electronic Cash System" [5], self published by the pseudonymous Satoshi Nakamoto, first described Bitcoin and introduced the concept of decentralized cryptocurrency. Bitcoin and the remaining cryptocurrencies filled in an important niche by providing a virtual currency decentralized system without any trusted parties and without pre-assumed identities among the participants that supports user-to-user transactions.

Cryptocurrencies can be purchased, sold and exchanged for other currencies at specialized currency exchanges. On the cryptocurrency trading market, much like the underlying principle where the Foreign Exchange market was built on, traders are essentially exchanging a cryptocurrency for another cryptocurrency or fiat currency [6]. The cryptocurrency market is known for the large fluctuations in price, in other words, it is known for its volatility [7].

To study the change of rates in multiple cryptocurrency pairs, a time series can be built from sampling the market at a fixed time rate using historical data from a specific cryptocurrency exchange platform. Using a simple application programming interface (API) all historic data used in this thesis was retrieved from one single exchange: Binance.[2]

There are several tools to analyse different markets, but the two major categories are *Fundamental* and *Technical* analysis [8]. These two approaches to analyse and forecast the market are not exclusive, they may be applied together and attempt to determine the direction prices are likely to move. Since a typical fundamental analysis cannot be executed for the cryptocurrencies market and because technical analysis is more suited for short-term trading [9], only technical analysis was employed in this work. A more elaborate description of technical analysis and each technical indicator can be found in [8].

2.1. Time series forecasting

Analysing financial time series is extremely challenging due to the dynamic, non-linear, non-stationary, noisy and chaotic nature of any financial market [10]. This analysis is carried out as an attempt to find the best entry (buy) and exit (sell) points to gain advantage over the market, increasing gains and minimizing the risk.

Machine learning procedures are capable of analysing large amounts of seemingly noisy and uncorrelated data in order to detect patterns and predict future data. Moreover, this approach provides a reaction time much faster than any human investor could deliver [3].

In this thesis, four total multivariate learning methods were used. Two of them, the *Logistic Regression* and the *Support Vector Machine* methods are linear and the other two, the *Random Forest* and the *Decision Tree Gradient Boosting* methods are non-linear. In the end a combined solution, an ensemble of these 4 algorithms, is calculated.

A brief summary of each classification model utilized in this study follows:

2.1.1. Logistic Regression (LR)

A binomial logistic regression [11] is used to model a binary dependent variable. In this type of learning algorithm, a single outcome variable Y_i follows a Bernoulli probability function that takes the outcome of interest, usually represented with the class label $Y = 1$, with probability p while the unwanted outcome, usually represented with the class label $Y = 0$, has probability $1 - p$.

The odds in favour of a particular outcome can be represented as $\frac{p}{(1-p)}$, where p stands for the probability of the wanted outcome. The logit function is capable of transforming input values in the range of 0 to 1, to values over the entire real number range, which can be used to express a linear relationship between feature values and the logarithm of the odds, also known as log-odds, as such [12]:

$$logit(P(Y = 1|x)) = \beta_0 + \sum \beta_i x_i, \tag{1}$$

where $P(Y = 1|x)$ is the conditional probability of Y belonging to class label 1 given x_i, the feature values, β_0 is the intercept (point that intercepts the function in the Y axis) and β_i corresponds to the coefficient associated with each respective feature.

Joining the log-odds and Eq. (1), the conditional probability can be represented as:

$$P(Y = 1|x) = \frac{1}{1 + exp(-\beta_0 - \sum \beta_j x_{ij})}. \tag{2}$$

Eq. (2) is called the logistic or sigmoid function (due to its S-shape). From this function it can be seen that $P(Y = 1|x)$ varies between 0 (as x approaches $-\infty$) and 1 (as x approaches $+\infty$). Thus, it is clear that the logistic function is able of transforming any real input into the range of 0–1. As a matter of fact, the class probabilities are obtained as such.

In this work the optimization problem utilized to obtain the coefficients and intercept, minimizes the following cost function [11]:

$$\min_{\boldsymbol{\beta},\beta_0} \frac{\boldsymbol{\beta}^2}{2} + C \sum_{i=1}^{n} log(exp(-Y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + \beta_0)) + 1), \tag{3}$$

where $\frac{\boldsymbol{\beta}^2}{2}$ is the L2 regularization penalty, C is a parameter inverse to the regularization strength and $\sum_{i=1}^{n} log(exp(-Y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + \beta_0)) + 1)$ corresponds to the negative log-likelihood equation [13]. In the negative log-likelihood equation, $\boldsymbol{\beta}$ represents the coefficients associated with each respective feature value, $\boldsymbol{x}_i$, both are in a vector structure. A method of finding a good bias–variance trade-off for a model is by tuning its complexity via the regularization strength parameter, C.

2.1.2. Random Forest (RF)

A Random Forest [14] is a method of ensemble learning where multiple classifiers are generated and their results are aggregated. RF is an enhancement of the method bootstrap aggregating (bagging) of classification trees.

[2] The python package "Binance official API" (https://github.com/binance-exchange/binance-official-api-docs/) was utilized to retrieve the used historical data from Binance's API (https://api.binance.com).

Before all else, a classification or decision tree is a simple model that takes into account the whole dataset and all available features. These trees tend to have high variance and overfit on training data, leading to a poor generalization ability on unseen data [13].

In the bagging method however, multiple decision trees are independently constructed using random samples drawn with replacement (known as a bootstrap sample) of the dataset in order to reduce the variance. Bagging is capable of reducing overfitting while increasing the accuracy of unstable models [15].

Random Forests improve the variance reduction on bagging by reducing the correlation between trees [16]. In order to do so, an additional layer of randomness is added to the bagging procedure. Instead of using all the available features, only a random subset of features is used to grow each tree of the forest. This strategy turns out to be robust against overfitting [17]. Reducing the amount of features will reduce the correlation between any pair of trees in the ensemble, hence, the variance of the model is reduced [14].

Let N be the number of data points in the original dataset, briefly, each tree in the random forest algorithm is created as follows [16]:

1. Draw a random sample of size N with replacement (hence) from the original data (bootstrap sample);
2. Grow a random forest tree to the bootstrapped data. Until the minimum node size is reached, recursively repeat the following steps for each terminal node of the tree:
   (a) Select a fixed amount of variables at random from the whole set;
   (b) Split the node using the feature that provides the best split according to the objective function;
   (c) Split the node into two daughter nodes.

Note that the size of the bootstrap sample is typically chosen to be equal to the number of samples in the original dataset as it provides a good bias–variance trade-off [12]. Each time the previous steps are repeated a new tree is added to the ensemble. Repeating this processes multiple times outputs an ensemble of trees with as many trees as this process was repeated. The predicted class probabilities of an input sample are computed as the mean predicted class probabilities of all trees in the forest.

2.1.3. Gradient Decision Tree Boosting (GTB)

Gradient Decision Tree Boosting in this work is utilized through the XGBoost framework, a scalable machine learning system for tree boosting [18].

Boosting is a general method for improving the accuracy of any given learning algorithm [19]. Boosting is the process of combining many *weak classifiers*[3] with limited predictive ability into a single more robust classifier capable of producing better predictions of a target [20]. Boosting is an ensemble method very resistant to overfitting that creates each individual members sequentially [13].

Gradient boosting replaces the potentially difficult function or optimization problem existent in boosting. It represents the learning problem as a gradient descent on some arbitrary differentiable loss function in order to measure the performance of the model on the training set [21].

In this paper the objective function for this model is applied as follows:

$$Obj = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k). \tag{4}$$

[3] Classifier whose error rate performs only slightly better than random guessing.

In Eq. (4), the first term, $L(y_i, \hat{y}_i)$, can be any convex differentiable loss function that measures the difference between the predicted label $\hat{y}_i$ and its respective true label y_i for a given instance. In this work's proposed system the log-likelihood loss will be used as loss function. Through the usage of this loss function, the calculation of probability estimates is enabled. Combining the principles of decision trees and logistic regression, the conditional probability of Y given x can be obtained [22].

The second term of Eq. (4), $\Omega(f_k)$, is used to measure the complexity of a tree f_k and is defined as:

$$\Omega(f_k) = \gamma T + \frac{\lambda \|w\|^2}{2}, \tag{5}$$

where T is the number of leaves of tree f_k and w is the leaf weights (i.e. predicted values stored at the leaf nodes). Including Eq. (5) in the objective function Eq. (4) forces the optimization of a less complex tree, which assists in reducing overfitting. The second term of this equation, $\frac{\lambda\|w\|^2}{2}$, corresponds to the L2 regularization utilized previously in LR. λ is the L2 regularization strength and γT provides a constant penalty for each additional tree leaf [23].

2.1.4. Support Vector Machine (SVM)

A Support Vector Machine [24] is a classifier algorithm whose primary objective is maximizing the margin, of a separating hyperplane (decision boundary) in an n-dimensional space, where 'n' coincides with the number of features used. The margin corresponds to the distance between the separating hyperplane and the training samples that are closest to this hyperplane, the support vectors.

The hyperplane is supposed to separate the different classes, that is, in a binary classification problem, the samples of the first class should stay on one side of the surface and the samples of the second class should stay on the other side.

Decision boundaries with large margins tend to have a lower generalization error of the classifier, whereas models with small margins are more prone to overfitting, hence, it is important to maximize the margins [12]. With the purpose of achieving a better generalization ability, a slack variable, ξ, indicating the proportional amount by which a prediction is misclassified on the wrong size of its margin is introduced. This formulation, called soft-margin SVM [24], enables controlling the width of the margin and consequently can be used to tune the bias–variance trade-off. With this soft-margin formulation, data points on the incorrect side of the decision boundary have a penalty that increases with the distance from the margin.

In order to maximize the margin, the hyperplane has to be oriented as far from the support vectors as possible. Through simple vector geometry this margin is equal to $\frac{1}{\|\mathbf{w}\|}$ [16], hence maximizing this margin is equivalent to finding the minimum $\|\mathbf{w}\|$. In turn, minimizing $\|\mathbf{w}\|$ is equivalent to minimizing $\frac{1}{2}\|\mathbf{w}\|^2$, the L2 regularization penalty term previously utilized in LR [16]. Therefore, in order to reduce the number of misclassifications, the objective function may be written as follows:

$$\min_{\mathbf{w},b,\xi} \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{n} \xi_i \text{ s.t. } Y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i \geq 0 \,\forall_i, \tag{6}$$

where $\|\mathbf{w}\|^2$ is the squared norm of the normal vector $\mathbf{w}$. C is the regularization parameter, responsible for controlling the trade-off between the slack variable penalty, ξ, and the size of the margin, consequently tuning the bias–variance trade-off. In order to calculate probability estimates, Platt scaling is employed [25].

It is likely that a non-linear margined SVM could provide better results, nonetheless, in this work a linear SVM is utilized for the simple reason that throughout the development of this work the available computational power was limited, and Platt scaling by itself is already a complex procedure that results in a significantly slower execution. Hence, in order to avoid increasing computational complexity even further, only a linear SVM was employed. Kernel methods, though, can deal with such linearly inseparable data by creating non-linear combinations of the original features to project them onto a higher dimensional space via a mapping function [12].

2.1.5. Ensemble Voting (EV)

The goal behind ensemble voting is to combine different classifiers into a meta-classifier with better generalization performance than each individual classifier alone. The weaknesses of one method can be balanced by the strengths of others by achieving a systematic effect [26].

In this work, a heterogeneous ensemble[4] was combined in a linear manner: the prediction's probability estimates from each different individual classifier were combined according to a simple unweighted average, giving equal probability to each individual output. This process, also known as soft majority voting [26], is the reason why all previous learning algorithms ought to yield a probability estimate for each class label, often with the cost of additional computer complexity.

2.2. State-of-the-art

Having described the essential background for this thesis, a literature review on works dedicated to forecasting financial markets using technical analysis now follows. Table 1 summarizes some of the most relevant studies applied to financial market analysis that were considered throughout the development of this paper's approach.

De Prado in [4], rather than using, as is customary, fixed time interval bars (minute, hour, day, week, etc.), proposes forming bars as a subordinated process of trading activity. Fixed time interval bars often exhibit oversampling, which leads to penalizing low-activity periods and undersampling penalizing high-activity periods, as well as poor statistical properties, like serial correlation, heteroscedasticity and non-normality of returns. According to the author, alternative bars, relatively to the commonly used time bars, achieve better statistical properties, are more intuitive particularly when the analysis involves significant price fluctuations and the conclusions tend to be more robust. The author mentions, the concept of resampled interval bars is not common yet in literature. As a matter of fact, throughout literature, no actual experimentation was found that could validate De Prado's claims.

Cardoso and Neves [27] proposed a system based on genetic algorithms to create an investment strategy intended to be applied on the Credit Default Swaps market. This market, similarly to the cryptocurrency market is still growing, is subject to speculation and is quite volatile. The employed strategy utilized several instances of genetic algorithms with the objective of increasing profitability. The obtained results suggest that it is possible to create a profitable strategy using only technical analysis as input data, reaching commonly a return on investment (ROI) over 50% in the CDS market. Jiang and Liang [28] present a model-less convolutional neural network trained using a deterministic deep reinforcement method in order to manage a portfolio of cryptocurrency pairs. Historic data sampled every 30 min from the 12 most-volumed cryptocurrencies of a given exchange is the only input of their system. The authors obtained an increase of 16.3% in their portfolio value, a sharpe ratio of 0.036 and a maximum drawdown of 29.6%, while the B&H method, used as benchmark, ended with a final portfolio value of 0.87, −1.54 sharpe ratio and 38.2% maximum drawdown.

Nakano et al. [29] utilize a seven layered artificial neural network (ANN) to create trading signals in the Bitcoin exchange market. Only technical indicators derived from Bitcoin/USD 15-minute return data were used as input. The authors defined three strategies of generating a trading signal. Two of them entered long and short and long positions but one only used long positions (similarly to this work's strategy). The strategy using only long positions without considering the bid–ask spread, yielded a final ROI of 12.14% while the simple B&H strategy obtained only 2.28%. The two remaining strategies, as expected, performed substantially better, exceeding a 50% ROI on both cases. Finally, the authors noted that increasing the amount of technical indicators (from 2 to 5) originated better results.

McNally et al. [30] try to predict the price direction of Bitcoin. For this, a Bayesian optimized recurrent neural network (RNN), a Long Short Term Memory (LSTM) network and ARIMA were explored and compared. The daily historical prices combined with two daily blockchain related features were the only inputs. The LSTM achieved the highest classification accuracy of 52.7% followed by RNN with 50.2% and ARIMA with 50.05%. The author concluded that Deep Learning models require significant amounts of data and 1-minute precision data would have been used if available while developing the work.

Greaves and Au [31] tried to predict whether Bitcoin's price increased or decreased in the next hour using accuracy as the classification metric. In this paper LR, SVM, Neural Network (NN) and a Baseline (percentage of the average price increase) were used. However, the distinctive aspect of this paper are the used inputs. Apart from using "current Bitcoin price", the remaining features were all related to blockchain network features (e.g."mined bitcoin in the last hour" or "number of transactions made by new addresses in a given hour"). They obtained accuracies of 53.4% for baseline, SVM followed with 53.7%, LR with 54.3% and finally NN were the best with 55.1%. The authors concluded that using input data from the blockchain alone offers limited predictability as price is dictated by exchanges and these fall outside the range of blockchains. Finally, it is presumed that price related features obtained from cryptocurrency exchanges are the most informative in regards to future price prediction. Similarly Tan and Yao [32] concluded that a series with technical indicators yields better results in terms of returns relatively to time series with weekly data in forecasting foreign exchange rates on a weekly basis with a NN model.

Żbikowski in [33] used a set of 10 technical indicators calculated from Bitcoin's historical price (with a 15-min precision) as input to investigate the application of SVM with Box Theory and Volume Weighted in forecasting price direction in the Bitcoin Market with the purpose of creating trading strategies. A simple B&H strategy used as baseline, which obtained a ROI of 4.86%, was outperformed by the BOX-SVM, with 10.6% ROI, and the VW-SVM, with 33.5% ROI. Mallqui and Fernandes [34] similarly attempt to predict the price direction of Bitcoin, but on a daily basis. The authors, besides the OHLC values and volume, experimented adding several blockchain indicators, as well as a few "external" indicators (such as crude oil and gold future prices, S&P500 future, etc.). Several attribute selection techniques were employed and always considered the OHLC values and volume as the most relevant attributes. Several ensemble and individual learning methods were experimented in this work. Nonetheless, the best performers were the SVM by itself and an ensemble of a Recurrent Neural Network and a Tree Classifier. The SVM obtained a final accuracy of 59.4% (utilizing 80% of the original dataset dedicated to training) and the ensemble an accuracy of 62.9% (utilizing 75% of the original dataset dedicated to training).

[4] Ensemble containing different learning techniques.

Akyildirim et al. [35] predict in the most liquid twelve cryptocurrencies utilizing data with different sampling frequencies ranging from daily to minute level. The authors utilized the methodologies SVM, LR, RF and ANN and historical price and technical indicators as inputs. The objective is predicting in a binary form the price direction in the next time step. ANN performed the worse with an accuracy slightly under 55%, it was concluded that no significant gain was acquired from using ANN, however, a larger sample size should be experimented with. LR obtained accuracies averaging 55%, SVM averaged slightly above 56% and finally, RF obtained the best accuracies at around 59%.

3. Implementation

In order to validate that utilizing resampled data rather than time sampled data is in fact advantageous, a single trading system capable of forecasting financial movements on these two types of data was developed. The forecasting results obtained from the various resampling datasets are represented and compared against each other in Section 4.

3.1. System's architecture

This trading system contains 5 different methodologies for forecasting used to detect the best entry and exit points in the cryptocurrency market. In order to predict these best entry and exit points in a financial market, the direction of price, rather than price levels, is forecast in this work. This method has proven to be effective and profitable by many authors in literature as can be seen in the state-of-the-art results mentioned in Section 2. Simply put, this work attempts to solve a binary classification problem.

It is expected beforehand that the predictions made from the ensemble will live up to expectations by exceeding the performance made from each individual learner's predictions. In order to create a good ensemble, it is generally believed that the base learners should be as accurate and diverse as possible [26]. Thus, in this proposed system a set of individual learners with these characteristics was chosen.

LR is one of the most widely used linear statistical model in binary classification situations [12]. It is a simple and easily implementable model that offers a reasonable performance, as was seen in Table 1. The linear SVM and non-linear RF methods, are used due to their well above average performances. Throughout supervised learning literature [36] and as was seen in Table 1, when compared to other learning methods, these two methods generally achieve top performances and are recommended by the authors. GTB is a non-linear method employed through the extreme gradient boosting (XGBoost) framework. This method is an efficient and scalable implementation of gradient boosting known for winning many machine learning competitions. These algorithms were considered preferable over Artificial Neural Network related models where due to its multi-layered process, an idea of the relationship between inputs and outputs is not provided [16].

In this system, as shown in Fig. 1 the common systematic procedure to predict time series using machine learning algorithms was put into practice by following these modules in this specific order: Data module; Technical Rules module; Machine Learning module and Investment Simulator module. In this same figure, the input and output of each module is also represented.

3.2. Data preparation module

The starting point for this work's proposed system is a collection of homogeneously sampled time series that will be processed in this module in order to acquire informative features. In this work the several alternative bar representations attained through data resampling are to be tested and compared to the common time sampling, thus, the original time series dataset containing columns for Open time, Volume and Open, High, Low and Close prices is now to be resampled. The original datasets are as detailed as one can get from Binance: 1 sample per minute. Contrarily to the customary time representation, the resampled data, is intended to place more importance in high-frequency intervals by overrepresenting the constituting individual samples, when compared to low-frequency intervals.

The reasoning behind the resampling procedures is that from a machine learning perspective, if both high and low-frequency intervals are equally represented, a mistaken prediction would yield an equivalent penalization for both cases. Now, assuming high-frequency intervals are overrepresented relatively to low-frequency intervals, if a high-frequency interval with many samples (all consisting of small time intervals) were to be erroneously predicted, a penalization for each single misclassification would be appointed resulting in a collectively large penalization. On the other hand, an erroneous prediction on a low-frequency interval would also be penalized, but not as heavily since it contains less samples to classify when compared to a high-frequency interval. Additionally, from a financial perspective, it is more important to successfully forecast high-frequency periods as larger returns or losses can be obtained when compared to more stable low-frequency periods.

This work's resampling is intended to sample data according to a fixed variation threshold rather than to a fixed time period. To exemplify the resampling procedure, a detailed description of the resampling mechanism for a fixed threshold consisting on a fixed percentual variation now follows.

Firstly, the percentual first order differencing of the closing values is calculated. Secondly, the sets of consecutive samples whose individual variations aggregated together reach or exceed a pre-defined threshold of absolute percentual variation must be identified. For this purpose, a customized cumulative sum of the percentual first order differencing is responsible for defining the boundaries of each group through assigning different numerical identifiers to each sample. To do so, starting on a specific sample, the total cumulative absolute variation of the specific sample as well as the consecutively posterior samples are added up until the variation sum reaches or exceeds the fixed threshold. This occurrence dictates the starting and ending boundary for each group: whenever a threshold is crossed, a group ends on the actual sample and a new one begins on the next sample with the next numerical identifier and cumulative sum reset back to zero.

In the end, all samples of the original dataset with the same identifiers are grouped into a single sample of the final dataset. To accomplish this, the Open and Close values of the new resampled data point are respectively the open value of the first entry and the close value of the last entry in the set of samples that make part of the respective group. The new High value is the highest of high values out of all the entries in the group and the new Low value is the lowest. Lastly, the new Volume corresponds to the sum of all volumes for each data point in the group. This process is done in a orderly manner that iterates through all samples (in a descendant order, where the oldest entry is at the top and the most recent is at the bottom), therefore, only consecutive samples can be grouped together. This process is illustrated in Fig. 2. As can be observed, whenever the value of the seventh column (Cumulative sum w/restart when threshold is reached) is

Table 1
Summary of the most relevant works covered in the state-of-the-art.

Ref.	Year	Financial market	Dataset time period (Data frequency)	Used methodologies	Evaluation function	System performance	B&H performance
[27]	2017	177 credit default swap markets	1/12/2007–1/12/2016 (Daily data)	Genetic algorithm	ROI	87.84% (ROI)	NA
[28]	2016	12 most-volumed cryptocurrencies exchange data	27/08/2015–27/08/2016 (30-min data)	Model-less convolutional Neural Network	Portfolio Value Maximization	16.3% (ROI portfolio value)	0.876% (ROI portfolio value)
[29]	2018	Bitcoin/USD exchange data	31/07/2016–24/01/2018 (15-min data)	ANN (only w/long positions implemented)	ROI	6.68% (ROI)	2.28% (ROI)
[30]	2016	Bitcoin/USD exchange data	19/08/2013–19/07/2016 (Daily data)	Long short term Memory network	Accuracy, root mean square error	52.7% (Accuracy)	NA
[31]	2015	Bitcoin/USD exchange data	01/02/2012–01/04/2013 (Hourly data)	SVM, LR, NN	Accuracy	53.7% for SVM, 54.3% for LR and 55.1% for NN (Accuracy)	NA
[33]	2015	Bitcoin/USD exchange data	09/01/2015–02/02/2015 (15-min data)	BOX-SVM, VW-SVM	Accuracy	10.58% for BOX-SVM, 33.52% for VW-SVM (ROI)	4.86% (ROI)
[34]	2019	Bitcoin/USD exchange data	01/04/2013–01/04/2017 (daily data)	SVM and Ensemble (RNN and tree classifier)	Accuracy and mean square error	59.4% and 62.9% (Accuracy)	NA
[35]	2018	12 most liquid cryptocurrencies exchange data	10/8/2017–23/6/2018 (15-min data)	RF (best performing model)	Accuracy	53% (Accuracy)	NA

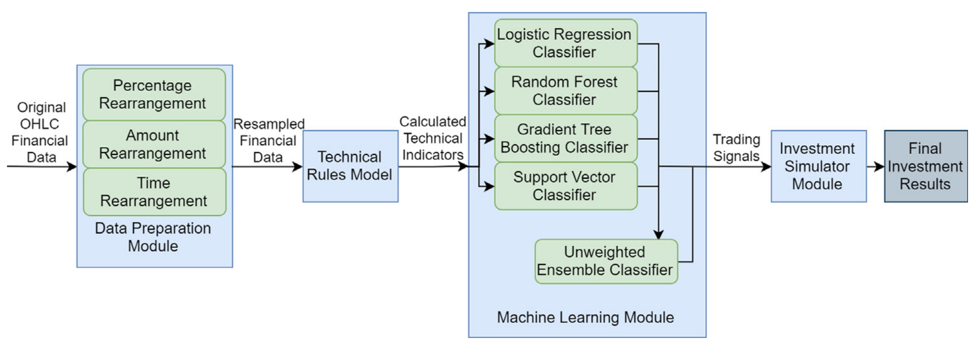


Fig. 1. Simplified overview of the proposed system's modular architecture.

equal or larger than a fixed threshold, in this case 2%, the group being numbered with the identifier 'i' is closed, the value for the cumulative sum is restarted and a new group with identifier '$i+1$' is initiated in the next sample.

It is worth adding that financial markets generate a discrete time series, making it impossible to achieve a perfectly even-sized resampled dataset with data from any cryptocurrency market. Nevertheless, utilizing a finer-grained periodic (smallest possible time period between data points) original data, is more likely to build a more regularly and consistently sized resampled dataset. Similarly to utilizing finer-grained data, if the threshold is increased, uneven resampled data becomes less noticeable as differences between resampled data points become percentually more insignificant resulting in a seemingly more consistent dataset. The values considered for the threshold should be larger than the payed fees, so each profitable transaction covers the fees completely, but not too large, otherwise too many data points would be grouped resulting in loss of information and consequently profit loss.

To resample the data according to different parameters, simply the first order difference series and a fixed threshold must be defined according to the chosen resampling parameter. In this work, besides time resampling, two alternative resampling processes were tested. Nevertheless, the process of resampling is analogous for any of two methods, thus, only the percentage resampling procedure is described in full detail.

In spite of the original dataset being already sampled according to time, due to the lack of rearrangement, it contains plenty more data points than each resulting resampled dataset of the three previous rearrangements. To make comparisons between the different resampling methods fair, a simple time grouping was implemented. This type of resampling consists on simply grouping consecutive data points. In the end both percentage and time resampled datasets ought to have a similar amount of samples in order to generate valid comparisons.

3.3. Technical indicators module

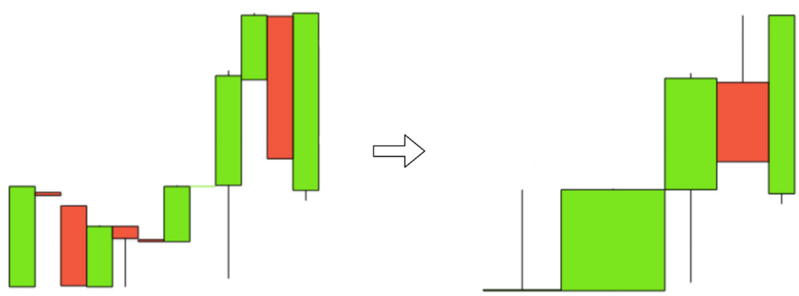
This module is responsible for computing and outputting the respective technical indicators for each data sample of the resampled time series outputted from the Data Module. Using technical analysis simplifies the problem of forecasting future market movements, to a pattern recognition problem (also referred to as a classification problem), where inputs are technical indicators derived from historical prices and outputs are an estimate of the price or its trend [37].

The remainder of this section will be dedicated to explaining in further detail the technical indicators applied on the historical data of cryptocurrency markets as well as the motivations

	Open	High	Low	Close	Close absolute percentual variation	Cumulative sum w/restart when 2% threshold is exceeded	Newly assigned group number
0	4.883	4.883	4.883	4.883	NA	NA	NA
1	4.856	4.954	4.856	4.954	1.45%	1.45%	0
2	4.948	4.948	4.945	4.945	0.18%	1.64%	0
3	4.935	4.935	4.857	4.857	1.78%	3.42%	0
4	4.856	4.916	4.856	4.915	1.19%	1.19%	1
5	4.915	4.915	4.856	4.903	0.24%	1.44%	1
6	4.902	4.902	4.9	4.9	0.06%	1.50%	1
7	4.9	4.955	4.9	4.954	1.10%	2.60%	1
8	4.954	4.954	4.954	4.954	0%	0%	2
9	4.955	5.067	4.864	5.062	2.18%	2.18%	2
10	5.058	5.123	5.058	5.121	1.17%	1.17%	3
11	5.12	5.12	4.981	4.981	2.73%	3.90%	3
12	4.95	5.123	4.94	5.123	2.85%	2.85%	4

	Open	High	Low	Close
-	4.883	4.883	4.883	4.883
0	4.856	4.954	4.856	4.857
1	4.856	4.955	4.856	4.954
2	4.954	5.067	4.864	5.062
3	5.058	5.123	4.981	4.981
4	4.95	5.123	4.94	5.123

(a) Table on the left contains the original dataset that is resampled into the dataset on the right.



(b) Left candlestick group is a representation of the original dataset and right candlestick group is arepresentation of the resampled dataset.

Fig. 2. Regrouping process illustration for a percentage of 2%.

for their choice. A more elaborate description of each technical indicator can be found in [8,38,39].

- **Exponential Moving Average (EMA)**: The EMA is used to dampen the effects of short term oscillations, through a smooth line representing the successive average. In an EMA, a body of data to be averaged moves forward with each new trading period. Old data is dropped as new data becomes available which causes the average to move along the time scale. By definition, this indicator is based on past prices, it is a trend following indicator unable to anticipate, only to react, thus, the moving average is a lagging indicator: it is able to follow a market and announce that a trend has begun, but only after the fact.
The major difference between a regular moving average and an EMA, is that the latter assigns a greater weight to recent data, having the ability to react faster to recent price variations. Because in this work the volatile cryptocurrency market is being forecast, the EMA is employed. Its formula is defined as following:

$$EMA_p(n) = EMA_{p-1} + \left(\frac{2}{n+1}\right)[Close_p - EMA_{p-1}]. \tag{7}$$

In Eq. (7), n refers to the number of time periods in the body of data to be averaged and p refers to the current period. Note that the first value coincides with the closing value. In this work six different EMA signals with time periods corresponding to $n = \{5, 10, 20, 50, 100, 200\}$ were implemented.

- **Moving Average Convergence–Divergence (MACD)**: The MACD is a simple momentum oscillator technique calculated using the difference between two exponential moving averages. To calculate the MACD line, traditionally, a 26-period EMA of the price is subtracted from a 12-period EMA, also of the price [8]. The MACD line is usually plotted at the bottom of a price chart along with the signal line. The signal line is an EMA of the MACD, commonly a 9-period EMA is used [8]. Finally, the difference between the two former lines composes the MACD histogram. In Eq. (8) the three components of the MACD indicator are presented.

$$MACD = EMA_{12} - EMA_{26}, \tag{8a}$$

$$Signal\ Line = EMA_9(MACD), \tag{8b}$$

$$MACDhistogram = MACD - Signal\ Line. \tag{8c}$$

The real value of the histogram is spotting whether the difference between the MACD and signal line is widening or narrowing. When the histogram is positive but starts to fall toward the zero line, the uptrend is weakening. Conversely, when the histogram is below negative but starts to move upward towards the zero line, the downtrend is losing its momentum. Although no actual buy or sell signal ought to be given until the histogram crosses its zero line, the histogram turns provide earlier warnings that the current trend is losing momentum. The actual buy and sell signals are given when the MACD line and signal line cross, that is, when the histogram is zero. A crossing by the MACD line above the signal line can be translated into a buy signal. The opposite would be a sell signal. Histogram turns are best used for spotting early exit signals from existing positions.

- **Relative Strength Index (RSI)**: The RSI is a momentum oscillator that measures the speed and change of price movements. This technical indicator is used to evaluate whether a market is overbought or oversold. The formula used on its calculation is:

$$RSI(n) = 100 - \frac{100}{1+RS(n)}, \quad \text{with} \quad RS(n) = \frac{AverageGains}{AverageLosses}. \tag{9}$$

In Eq. (9), n refers to the number of time periods being analysed (traditionally 14 time periods are used [8]), $AverageGains$ refers to the average gain of up periods during the last n periods and $AverageLosses$ refers to the average loss of down periods during the last n periods. The RSI varies between a low of 0 (indicating no up periods) to a high of 100 (indicating exclusively up periods). Traditionally, movements above 70 are considered overbought, while an oversold condition would be a move under 30. A RSI divergence with price is a warning of trend reversal.

- **Rate Of Change (ROC)**: The ROC is a simple momentum oscillator used for measuring the percentual amount that prices have changed over a given number of past periods. Traditionally 10 time periods are used [8]. A high ROC value indicates an overbought market, while a low value indicates an oversold market. The formula for calculating this indicator is as follows:

$$ROC(n) = 100 \times \frac{Close_p - Close_{p-n}}{Close_{p-n}}. \tag{10}$$

In Eq. (10), n refers to the number of time periods being analysed, and p corresponds to the current period.

- **Stochastic Oscillator**: The Stochastic Oscillator's intent is to determine where the most recent closing price is in relation to the price range of a given time period. Three lines are used in this indicator: the %K, the fast %D line and the slow %D line. The %K line, the most sensitive of the three, simply measures percentually where the closing price is in relation to the total price range for a selected time period, typically of 14 periods [8]. The second line, the fast %D is a simple moving average of the %K line, usually of 3 periods [8]. The previously mentioned %K line compared with a three-period simple moving average of itself, the fast %D line, corresponds to the fast stochastic. For the fast stochastic, when the %K line is above the %D line, an upward trend is indicated. The opposite indicates a downward trend. If the lines cross, the trend is losing momentum and a reversal is indicated. However, the %K line is too sensitive to price changes and due to the erratic volatility of the fast %D line, many false signals occur with rapidly fluctuating prices. To combat this problem, the slow stochastic was created. The slow stochastic consists on comparing the original fast %D line with a 3-period simple moving average smoothed version of this same line, called slow %D line [8]. In other words, the slow %D line is a doubly smoothed moving average of the %K line. The formulae for the %K and both %D lines are as follow:

$$\%K_n = 100 \times \frac{Close_t - min(Low)_n}{max(High)_n - min(Low)_n}; \tag{11a}$$

$$Fast\%D = SMA_p(\%K_n); \tag{11b}$$

$$Slow\%D = SMA_p(Fast\%D). \tag{11c}$$

In Eq. (11), $Close_t$ corresponds to the current close, $min(Low)_n$ refers to the minimum Low of the previous n periods, and $max(High)_n$ refers to the maximum High of the previous n periods. SMA_p corresponds to a simple moving average of p periods.
In addition to these 3 signals, the histogram method from the MACD indicator was replicated for the stochastic oscillators to indicate trend reversals (when a sample of this indicator crosses zero) as well as whether the trend is upwards or downwards.

- **Commodity Channel Index (CCI)**: The CCI is an oscillator used to measure the variation of a price from its statistical mean. A high CCI value indicates that prices are unusually high compared to the average price, meaning it is overbought. Whereas a low CCI value indicates that prices are unusually low, meaning it is oversold. Traditionally, high and low CCI values respectively correspond to over 100 and under −100. The formula for calculating this indicator is as follows:

$$CCI(n) = \frac{1}{0.015} \times \frac{TP_p - SMA_n(TP_p)}{\sigma_n(TP_p)}, \quad \text{with} \quad TP_p = \frac{High_p + Low_p + Close_p}{3}, \tag{12}$$

where TP_p is referred to as the typical price and $High_p$, Low_p and $Close_p$ represent the respective prices for the time period p. The item $SMA_n(TP_p)$ is the simple moving average of the typical price for the previous n time periods under consideration, and $\sigma_n(TP_p)$ corresponds to the mean deviation of the SMA during the previous n periods. Commonly, 14 periods are used for the SMA used in this indicator [8]. Lambert, the creator of this indicator, set the constant 0.015 for scaling purposes, to ensure that approximately 70 to 80 percent of CCI values would fall between −100 and +100.

- **On Balance Volume (OBV)**: The OBV indicator is a running total of volume. It relates volume to price change in order to measure if volume is flowing into or out of a market, assuming that volume changes precede price changes.
The total volume for each day is assigned a plus or minus sign depending on whether the price closes higher or lower than the previous close. A higher close causes the volume for that day to be given a plus value, while a lower close counts for negative volume. A running cumulative total is then maintained by adding or subtracting each day's volume based on the direction of the market close. The formula used on its calculation is:

$$OBV_p = \begin{cases} OBV_{p-1} + Volume_p, & \text{if } Close_p > Close_{p-1} \\ OBV_{p-1} - Volume_p, & \text{if } Close_p < Close_{p-1} \\ OBV_{p-1}, & \text{if } Close_p = Close_{p-1}. \end{cases} \tag{13}$$

In Eq. (13), p refers to the current period and $p-1$ refers to the previous period. The OBV line should follow the same direction of the price trend. If prices show a series of higher peaks and troughs (an uptrend), the OBV line should do the same. If prices are trending lower, so should the OBV line. It is when the OBV line fails to move in the same direction as prices that a divergence exists and warns of a possible trend reversal. It is the trend of the OBV line that is relevant, not the actual numbers themselves.

- **Average True Range (ATR)**: The ATR indicator is used as a measurement of price volatility. Strong movements, in either direction, are often accompanied by large ranges (or large True Ranges). Weak movements, on the other hand, are accompanied by relatively narrow ranges. This way, ATR can be used to validate the enthusiasm behind a move or breakout. A bullish reversal with an increase in ATR would show strong buying pressure and reinforce the reversal. A bearish support break with an increase in ATR would show strong selling pressure and reinforce the breaking of support. To calculate the ATR, the True Range must be firstly calculated:

$$TR_p = max\{High_p - Low_p;\ |High_p - Close_{p-1}|;\ |Low_p - Close_{p-1}|\}. \quad (14)$$

In Eq. (14), p refers to the current period and $p-1$ refers to the previous period. Having calculated the True Range, the next step is calculating the Average True Range. The ATR is a simple average of the previous n (traditionally 14 time periods are used [8]) True Range values:

$$ATR_p(n) = \frac{ATR_{p-1} \times (p-1) + TR_p}{n}. \quad (15)$$

In Eq. (15), p refers to the current period, $p-1$ refers to the previous period and n refers to the number of time periods to be analysed.

To summarize this section, Table 2 lists the used technical indicators. An additional column was added to contain the respective time period parameter (previously denoted as n) for all indicators that possess it. For the indicators that do not require an adjustable time period parameter, a '-' is placed instead. In this work, the time period corresponds to the set of older instants to be considered for each specific calculation.

3.4. Machine learning module

This module is the most complex and the main core of the system, as so it was divided into 3 main components, each with a different responsibility. In this system, the resampled data is firstly scaled by standardization, then the target prediction vector is defined and lastly the actual machine learning training and forecasting procedures are executed. Fig. 3 contains a brief illustration of this module's main steps.

3.4.1. Feature scaling

Feature scaling is a data preprocessing step, applied on the technical indicator series outputted from the previous module through a technique called *standardization*. Standardization is the process of centring features by subtracting the features' mean and scaling by dividing the result by the features' standard deviation. The standardized value, of a given sample, x, is calculated as follows:

$$x_{standardized} = \frac{x - \mu}{\sigma}, \quad (16)$$

where μ is the mean of the training samples and σ is the standard deviation of the training samples.

Feature scaling is not only important if we are comparing measurements that have different units, but it is also a general requirement for many machine learning algorithms. When the input dataset contains features with completely different units of measurement, which is the case in this system, using feature scaling avoids the problem in which some features come to dominate solely because they tend to have larger values than others [12].

3.4.2. Target formulation

The objective of the second component of this module is defining the target discrete values to be predicted, the class label, for each data point. This process is a requirement when using methods of supervised learning.

Throughout literature, the bid–ask spread is an issue that is often overlooked. While its impact may be negligible on specific cases, it may also be responsible for harming the system's investments in a real situation if unaccounted for. In less liquid markets, specifically in the most "exotic" cryptocurrency pairs (whose financial worth is typically in the USD cents), the percentual difference between the best buyer and the best seller may become large enough so that the common assumption of utilizing the closing value as both bid and ask values becomes quite far-fetched leading to a disparity with reality that may no longer become acceptable. However, to the author's knowledge, enquiring specific exchanges or sources for the spread of the current instant is the only method of acquiring this data free of charge. Regardless, in order to assemble a decent amount of data, years of data gathering would be required. In this work, due to the unavailability of bid–ask spread value, only transaction fees are considered. Nevertheless, it should be pointed that overlooking the importance of bid–ask spreads is not ideal and prior to applying this system in a real scenario, a database containing the spread values should be built and taken into account to fully validate this strategy.

The buy and sell trading fees were taken into account when determining the ideal long position start and ending points. The trading fees applied in this system are Binance's general fee of 0.1%.

In order to decide whether the signal in the next instant has one of the two possible outcomes, a deterministic binary classification is proposed. In this proposed system, a binary outcome was taken into account: the cryptocurrency market either has a positive or a negative variation in the next instant. In the rare cases where the variation is precisely null, the outcome of the previous instant is duplicated into the current one.

A vector called *vector y* (represented in Eq. (17)) contains the target classification for each sample in the dataset. Each entry follows a binomial probability distribution, i.e., $y \in \{0, 1\}$, where 1 symbolizes a positive or bullish signal variation and 0 symbolizes a negative or bearish signal variation. For a given currency pair, being $close_t$ the closing price for a given time period t, and $close_{t-1}$ the closing price for the previous time period, $t-1$, the target for the given time period, y_t, is defined in a probabilistic way to employ the four learning algorithms as follows:

$$y_t = \begin{cases} 0, & \text{if } Close_{t-1} \times (1 + Fee) < Close_t \times (1 - Fee) \\ 1, & \text{if } Close_{t-1} \times (1 + Fee) > Close_t \times (1 - Fee) \\ y_{t-1}, & \text{if } Close_{t-1} \times (1 + Fee) = Close_t \times (1 - Fee) \end{cases}, \quad (17)$$

Given that the utilized data was retrieved from Binance, to be truly faithful to a real case scenario, only long positions can be adopted by this system, short positions are disabled.

3.4.3. Classifiers fitting and predicting

This is the last component of the Machine Learning module. The objective of this component is, out of a dataset received as input, creating a model with the learned data by training each of the four classification algorithms and, in the end, creating a classifier ensemble with the output of each algorithm by averaging each sample. The four individual learning algorithms can be fit in whatever order.

Before the actual training procedure begins, both the features (matrix X) and the target vector (vector y), must be split into train dataset and test dataset. All training procedures, including cross-validation and grid searching, occur on the training datasets while the test datasets are kept isolated throughout the following procedures mentioned in this section. In this work, 4 splits divide the time series in 5 equal sized intervals, this is, each interval contains the same number of samples. It is worth noting that because the splits divide the number of samples, only the time rearranged series will be split in 5 time intervals with same time duration. While the test interval size is consistent for all iterations, the train set successively increases in size with every iteration, it may be regarded as a superset of the train sets from the previous iterations. Ultimately, when using this method, no

Table 2
List of all indicators and (if applied) their respective parameters, fed as input to the Machine learning algorithms of this work's system.

Technical indicator	Time period value
Exponential moving average	{5, 10, 20, 50, 100, 200} periods
Moving average convergence divergence histogram	[signal EMA 9, fast EMA 12, slow EMA 26] periods
Relative strength index	14 periods
Rate of change	10 periods
Stochastic oscillator %K line	14 periods
Stochastic oscillator fast %D line	3 periods
Stochastic oscillator slow %D line	3 periods
Histogram between %K line & fast %D line	–
Histogram between %K line & slow %D line	–
Commodity channel index	14 periods
On balance volume	–
Average true range	14 periods

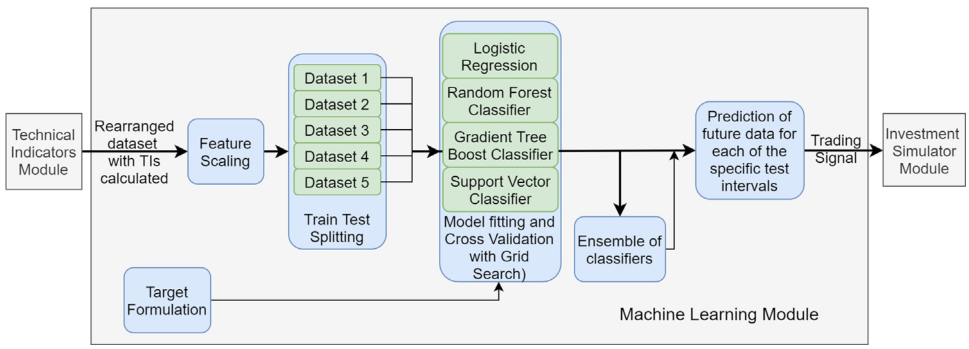


Fig. 3. Machine learning module overview.

future data leaking occurs and forecasting since a much earlier time point is enabled.

As mentioned in Section 2 each learning algorithm requires the initialization of a set of parameters before the actual learning process begins. These parameters are called hyper-parameters [13].

In order to tune the hyper-parameters of each learning algorithm, a simple grid search was carried out during a specific cross-validation procedure: *time series cross-validation* [40], a variation of the commonly used *k-fold* cross-validation destined specifically for time series data samples. In the kth split, the first k folds are returned as train set and the $(k+1)$th fold as validation set. Grid-searching is a simple exhaustive search through a manually specified subspace of values with the purpose of finding the best values for each hyper-parameter [12]. During the model training step, in each fold, various instances of each learning algorithm are trained with all the possible hyper-parameter combinations and tested on the respective validation set. This way, the performance according to the negative log-loss metric is obtained on unseen data for each hyper-parameter combination, the values that generate on average the best negative log-losses were chosen. Table 3 contains the grid employed for each hyper-parameter of each learning algorithm. It is worth adding that not too many parameters were grid searched due to computational and time limitations.

It is worth adding that the negative log-loss classification metric was employed in this system because investment strategies profit from predicting the right label with high confidence: contrarily to other classification metrics, the negative log-loss takes into account not only whether each predictions is correct or not, but also the probability of each prediction [4].

The results for each hyper-parameter combination are averaged through all validation sets. The instance of the model containing the average highest cross-validated scoring parameters is then refit on the whole training dataset to measure its performance on the whole training dataset continuously. This best instance is also later used as the final model to issue the final predictions on the test dataset.

In this system, even though the grid search is not too extensive, each test dataset utilizes a set of hyper-parameters computed during its respective training procedure on the training dataset, thus no single optimal case was used for all datasets. Besides not compromising the partitioning of future data whatsoever, this method was employed because there are countless cryptocurrencies with entirely different behaviours and characteristics: a single model could be accurate for a specific cryptocurrency but disastrous for another.

This employed cross-validation is identical to the previously explained process of splitting train and test data, the only differences are that rather than only 4 splits, 10 splits are done and instead of splitting train and test data, train and validation data is being split. This is a simple notation difference. In this work, a simple grid search was carried out in this proposed system in order to find the best hyper-parameters according to the negative log-loss metric.

3.4.4. Forecasting and ensemble voting

At this point the second part of supervised learning, the forecasting, commences. For each specific training dataset, all learning algorithms apply the fit model on the respective testing dataset. All 4 algorithms generate a probability estimate of the respective class labels for each sample of the test dataset. The classification odds for each test sample of the 4 models, are now combined through *soft majority voting* into an ensemble. In soft majority voting, the probabilities for the prediction of each class label of the test dataset previously calculated are unweightedly averaged, designated as Ensemble Voting (EV).

At last, this module outputs five different trading signals generated containing the forecasting data. Four trading signals are originated from the individual learning algorithms while the last is originated from the unweighted average of these four. All the trading signals have the same time frame, and will be simulated

Table 3
Respective hyper-parameters utilized for each learning algorithm and its value (or set of values if grid searched).

Learning algorithm	Hyper-parameter	Grid of values
Logistic regression	C	[0.1, 0.01, 0.001, 0.0001]
	Penalty	L2
	Solver	Stochastic average gradient [41]
Random forest	Number of trees	400
	Min. num. of samples required to be leaf node	9
	Min. num. of samples required to split node	9
	Splitting criteria	Gini
	Minimum impurity threshold	10^{-7}
	Amount of features considered	$\sqrt{num\ of\ features}$
Gradient tree	Number of trees	100
Boosting	Max tree depth	3
	Min loss reduction required to make partition	1
	Minimum sum of instance weight	[1, 2]
	L2 regularization strength	[0, 1]
	Step size shrinkage	0.01
Support vector Classifier	C	1
	Kernel	linear

in a real world environment in the next module, the investment simulator.

3.5. Investment simulator module

The investment simulator model is responsible for translating the five trading signals obtained as input from predicted data in the Machine Learning module, into market orders with the purpose of simulating the forecasts' performance in a real market environment.

The trading signal received as input, consists on a series containing the probability of each sample being classified with the label 1. This series containing probabilities is easily converted into a dichotomous class label series according to the largest likelihood. After being converted into a discrete classification series, each entry of this series is interpreted as follows: A class label of 1 means the currency is forecast to be bullish in the next instant representing a *buy* or *hold* signal; A class label of 0 means the currency is forecast to be bearish in the next instant representing a *sell* or *out* signal. Binance trading fees are taken into consideration in the trading simulator with the purpose of reproducing a real market. When a class label 1 signals *buy*, all available capital is applied in opening a long position. Similarly, a class label 0 orders the system to exit the market and retrieve the available capital.

This module starts with a fixed capital for investment (by default it is one unit of quote currency) and invests that capital according to the previously mentioned interpretation of the class label series.

To simulate the market orders, backtest trading is employed in this work. Because this backtest process utilizes historical data, the following two assumptions must be imposed for this proposed system:

1. Market liquidity: The markets have enough liquidity to conclude each trade placed by the system instantaneously and at the current price of their placement.
2. Capital impact: The capital invested by the algorithm has no influence on the market as it is relatively insignificant.

Lastly, as a method to control risk on a trade-by-trade basis, in this system stop-loss orders were implemented.

In the end, a series of metrics, revealed and explained in Section 4.1, are calculated, plotted and stored into memory.

4. Results

This section starts by describing the financial data, evaluation metrics and an additional strategy utilized as comparison baseline for this work's system. Afterwards, the overall results obtained are reported to evince that this system is valid and posteriorly the results for each different type of resampling are presented and analysed in form of case studies. Lastly, follows a discussion and comparison of the results obtained with the time and alternative resampling procedures.

Regarding the utilized financial data, instead of using the nearly 400 currently available pairs in Binance, only the 100 pairs with the most traded volume in USD were selected. This selection filters out many pairs that have been recently listed and do not have a large enough dataset. Moreover, it becomes more likely that the selected markets are in conformity with the two backtest trading assumptions.

In order to use the maximum amount of available data, no common starting date was chosen for all currency pairs. Each currency pair's data begins at the moment Binance started trading and recording the specific currency pair. On the other hand, the ending date, is fixed at 30th October 2018 at 00h00. Out of the used cryptocurrency pairs, the largest pairs originally contain 676 827 trading periods while the smallest pairs contain 68 671 trading periods. Each trading period, as previously mentioned, has the duration of 1 min. The starting date varies from 14th July 2017 (Binance's official launch date) for the oldest pairs, up to 12th October 2018 for the most recently introduced pair.

The original data for the 100 selected currency pairs, prior to its usage in forecasting, must be resampled. As was explained in Section 3.4, to carry out the multiple resampling procedures, a fixed percentual variation threshold must be picked to define the approximate final size of each candle in the final dataset. In the subsequent results, a fixed percentage of 10% was chosen for the resampling procedures.

4.1. Evaluation metrics

As was mentioned before, the main goal of this proposed system is maximizing the negative logarithmic loss and returns while minimizing the associated risk of this work's trading strategy. With this goal in mind, the following metrics of market performance were additionally calculated for each currency pair with the intention of providing a better analysis of the obtained results:

4.1.1. Return on Investment (ROI)

The return on investment measures the amount of return gained or lost in an investment relative to the initially invested amount. The simple standard formula of this metric is represented as follows:

$$ROI = \frac{FinalCapital - InitialCapital}{InitialCapital} \times 100\%, \tag{18}$$

where *FinalCapital* corresponds to the capital obtained from the investment bought with *InitialCapital*.

4.1.2. Maximum Drawdown (MDD)

The maximum drawdown measures the maximum decline, from a peak to a through before a new peak is attained. It is used to assess the relative downside risk of a given strategy [42]. This metric is calculated as follows:

$$MDD = \max_{t \in (StartDate, EndDate)} [\max_{t \in (StartDate, T)} (ROI_t) - ROI_T], \tag{19}$$

where ROI corresponds to the return on investment at the subscript's point in time and $\max_{t \in (StartDate,T)}(ROI_t)$ corresponds to the highest peak from the starting point until the instant T. In this work, as is customary, MDD it is quoted as a percentage of the peak value.

4.1.3. Sharpe ratio

The Sharpe Ratio is a method for calculating the risk-adjusted return. This ratio describes the excess return received for holding a given investment with a specific risk. The sharpe ratio is calculated as follows:

$$SharpeRatio = \frac{ROI - R_f}{\sigma}, \tag{20}$$

where ROI corresponds to the return on investment, R_f is the current risk-free rate, 3.5%, and σ is the standard deviation of the investment's excess return.

4.1.4. Sortino ratio

The Sortino Ratio is a modification of the Sharpe ratio metric. In contrast to Sharpe ratio, the Sortino ratio includes only negative variations. The formula for calculating this metric is identical to the formula represented in Eq. (20), however, the denominator corresponds only to the standard deviation values observed during periods of negative performance.

4.1.5. Additional performance parameters

Besides the four presented metrics used to evaluate the performance of an investor, the following parameters are also used in the classification of this proposed system:

- Percentage of periods in market: Percentage of time periods where a long position was in effect out of all the available time periods of the testing set;
- Percentage of profitable positions: This parameter is calculated by dividing the number of trades that generated a profit (with fees included), by the total number of trades; This probability is complementary to the percentage of non-profitable positions;
- Average Profit per position: Average percentual profit or loss per position;
- Largest Percentual Gain: Most profitable position;
- Largest Percentual Loss: Greatest loss.

With the purpose of validating this system, besides testing with real market data through backtest trading and analysing the results with the just introduced metrics, a previously mentioned investment strategy, the Buy and Hold (B&H) is applied. According to the Efficient Markets theory [43], prices are independent of each other, hence no profit can be made from information based trading. In conformity with this theory, the best strategy is employing a B&H strategy, regardless of market fluctuations. Due to the limited number of existent solutions for trading in multiple cryptocurrency pair markets and to put the Efficient markets theory to the test, B&H was defined as a benchmark strategy intended to be an additional term of comparison for this work's proposed system.

4.2. Case studies

In this section, the case studies and the main results obtained through the application of the described strategies are presented.

4.2.1. General overview

First of all, a general idea of the overall performance obtained with each different methodology independently of the resampling method utilized is provided. To achieve this, whilst utilizing the B&H strategy as benchmark, the five trading signals generated by the four individual learning algorithms and the ensemble voting method are individually averaged and subsequently compared against each other. The results are represented in Table 4.

Through analysis of Table 4, it is clear that the B&H strategy yields the worst results. Out of all individual learning algorithms, the SVC generates the highest ROI and is the least risky, as so, it can be considered as the best individual learner. LR follows in terms of the metric ROI but is quite risky as a forecasting method. Both RF and GTB performed averagely and both yielded modest ROIs. Nonetheless, EV is by far the most robust alternative. As can be seen, the trading signal obtained from EV on average outperforms the remaining ones according to the majority of metrics.

The obtained accuracies are on par or, for the case of EV, exceed the accuracy of most papers regarding cryptocurrency exchange market forecasting throughout the state-of-the-art (Table 1). The risk measures and returns on investment are also superior on average.

In conclusion, there is no clear performance hierarchy for each of the 4 individual learning algorithms. Nonetheless, the trading signals generated by the EV methodology obtain the top performance out of all tested methods. In any case, all five methodologies clearly outperform the plain B&H strategy.

4.2.2. Time resampling

Time resampling is the most widely used method of resampling, therefore, it is the first sampling method analysed to be considered as the comparison baseline.

Firstly, a temporal graph showing the evolution of the average ROI per market for each trading signal is represented in Fig. 4. In other words, this figure, contains for each time instant, the respective average ROI of all 100 analysed markets. In this figure it can be observed that from May until September 2018 the analysed market's price drops considerably, as can be confirmed by B&H's signal. EV and SVC were the methodologies that suffered the smallest losses in this period. Overall, most methodologies are quite conservative when the market suddenly rises, hence a B&H strategy earns more in these periods. On the other hand, this conservative behaviour is also responsible for minimizing losses in sudden price drops, contrarily to the B&H strategy.

Secondly, Table 5 contains the general statistics obtained for the time resampling method. In this table, a better method cannot be clearly defined. Relatively to EV, SVC achieved a worse predictive power and is slightly more risky, yet obtained nearly twice the profits. Nonetheless, EV stands out due to the better accuracies and NLL as well as due to the remarkably small percentage of periods in market and top values in the risk metrics, thus suggesting that this is clearly the least risky alternative.

Table 4
Comparison between the Buy & Hold strategy and each of the five methodologies employed.

Parameter	B&H	LR	RF	GTB	SVC	EV
Average obtained results (for all markets and resampling methods)						
Final ROI	−10.5%	519%	295%	335%	538%	**615%**
Accuracy	40.20%	53.50%	53.62%	53.51%	53.38%	**56.28%**
Negative log-loss	−20.6	−0.7031	−0.6992	−0.6918	−0.6975	**−0.6829**
Periods in market	100%	56.0%	52.9%	55.0%	50.4%	**39.9%**
Profitable positions	19.5%	**60.2%**	56.2%	58.3%	58.8%	57.6%
Profit per position	−10.6%	0.57%	0.31%	0.33%	0.61%	**0.69%**
Largest gain	**35.6%**	17.6%	17.7%	17.3%	18.4%	15.0%
Largest loss	−46.0%	−14.4%	−15.0%	−15.1%	−14.5%	**−13.2%**
Max drawdown	79.9%	57.6%	60.6%	62.0%	54.7%	**49.3%**
Annual sharpe ratio	−0.164	0.769	0.413	0.312	0.848	**0.945**
Annual sortino ratio	0.169	2.374	1.665	1.407	2.568	**2.821**

Table 5
Average obtained results for the Buy & Hold and each of the five methodologies employed for time resampling.

Parameter	B&H	LR	RF	GTB	SVC	EV
Average obtained results (all markets are considered)						
Final ROI	−27.9%	25.6%	3.92%	3.30%	**39.5%**	18.7%
Accuracy	37.40%	54.77%	55.58%	54.84%	54.70%	**59.26%**
Negative log-loss	−21.6	−0.6931	−0.6896	−0.6890	−0.6757	**−0.6746**
Periods in market	100%	51.4%	44.4%	48.9%	44.9%	**27.7%**
Profitable positions	15.0%	**57.6%**	50.8%	53.4%	56.0%	55.0%
Profit per position	−27.9%	0.05%	0.01%	−0.02%	**0.09%**	0.06%
Largest gain	20.8%	18.1%	**20.8%**	19.6%	18.4%	15.3%
Largest loss	−48.7%	−15.2%	−14.5%	−14.9%	−16.2%	**−12.3%**
Max drawdown	77.0%	55.6%	56.8%	59.6%	52.5%	**42.5%**
Annual sharpe ratio	−0.352	0.075	−0.135	−0.327	0.228	**0.288**
Annual sortino ratio	−0.137	0.608	0.330	−0.074	0.985	**1.102**

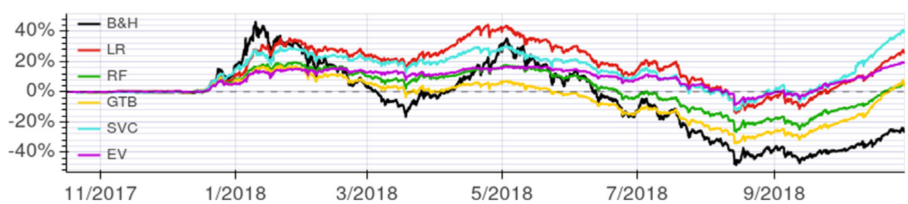


Fig. 4. Average accumulated ROI [%] for each instant from the starting until the last test point with time resampled data.

Thirdly, two specific markets and their return on investment evolution are represented in Figs. 5 and 6 and their statistics are shown in Table 6. In these figures, the top subfigure contains a candlestick representation of the utilized resampled data for each cryptocurrency pair. As can be seen, the background is divided into five different coloured periods. Each coloured period represents a different sub-dataset of the train-test splitting procedure. This splitting procedure divides the data in five intervals with the same amount of samples. Additionally, it is also visible that the ROI line only begins at the start of the second dataset, which is in accordance with the fact that the first sub-dataset is only used during the training procedure. Note that, for the trading signals where stop-loss orders are active (all except the ones originated by B&H) throughout this whole chapter, the largest loss peaks at approximately −20%, the stop-loss activation percentage.

Fig. 5 contains the example of a market where this system performed remarkably well. From this figure and the upper half of Table 6 it is clear that the trading signal from the LR outperformed the remaining signals. EV obtained a clear second place with mostly above average results. Fig. 6, on the other hand, contains a market with one of the worst performances obtained with this system. From this figure and the lower half of Table 6 it can be concluded that EV yielded the best results with lowest risk. However, if the trading signals from which it is derived from do not perform well, there is only so much EV can do. In any case, the B&H strategy was outperformed by the remaining trading signals.

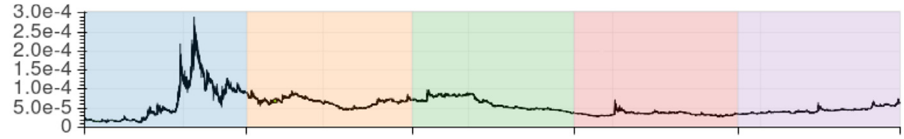
4.2.3. Percentage resampling

The results from this type of resampling, were only narrowly outperformed by amount resampling in terms of predictive power and final returns. Nevertheless, on average these results are noticeably less risky. Due to its low risk and only slightly lower profits and predictive power, this resampling method originates a more desirable outcome, hence, it is worth carrying out a more in depth analysis of the obtained results.

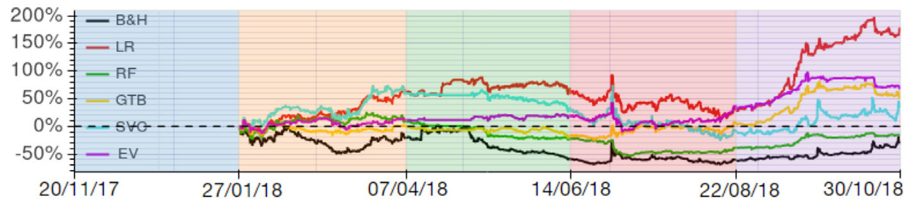
Firstly, a temporal graph showing the instantaneous average ROI of the 100 analysed markets for each of the methodologies is presented in Fig. 7. In this figure, although not as significant as the previous resampling method, the drop in the markets from May until September 2018 is also visible in this case. Once again, the signal from B&H was clearly surpassed by all remaining trading signals in terms of profits.

Secondly, a table containing the general statistics obtained for percentage resampling method follow in Table 7. In this table, even though, on average EV's returns are slightly inferior to SVC, the former distinctly obtains the highest predictive power and the lowest risk out of all other trading signals, thus it may be concluded that the EV method clearly generates the trading signals with the top results. Once again, the B&H strategy clearly obtained the worst average performance.

Thirdly, the results for two specific markets and their return on investment evolution are represented in Figs. 8 and 9 and their statistics are shown in Table 8. These figures follow the same structure and contain the same markets as the figures presented time resampling.

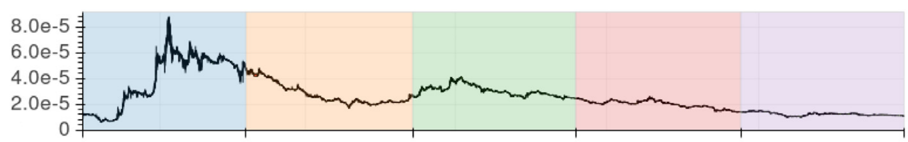


(a) OHLC representation with time resampled data.

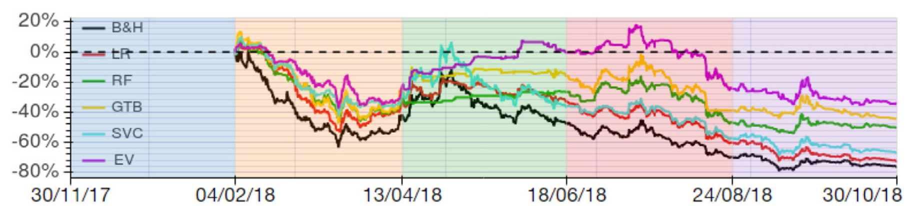


(b) Chronological evolution of the ROI [%] for the 5 calculated trading signals and B&H strategy.

Fig. 5. ROI variations for currency pair POEETH (Po.et/Ethereum) with time resampling applied.



(a) OHLC representation with time resampled data.



(b) Chronological evolution of the ROI [%] for the 5 calculated trading signals and B&H strategy.

Fig. 6. ROI variations for currency pair ADABTC (Cardano/Bitcoin) with time resampling applied.

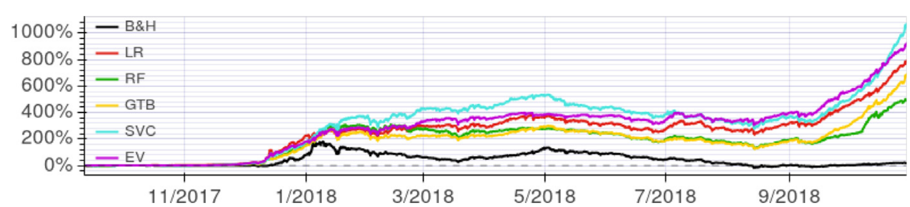
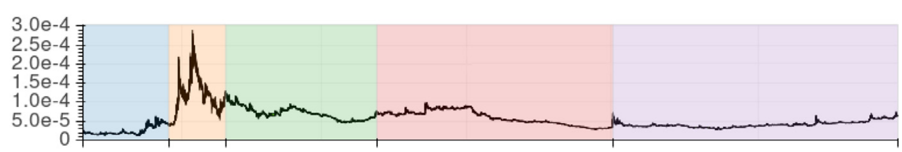
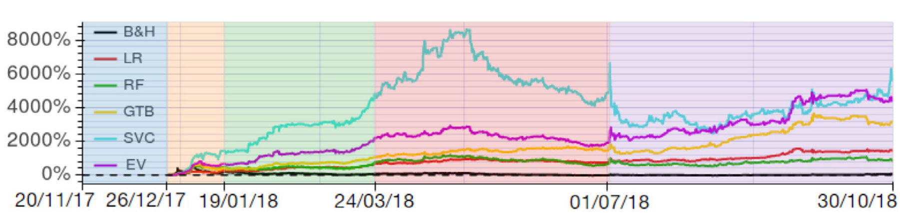


Fig. 7. Average accumulated ROI [%] for each instant from the starting until the last test point with percentage resampled data.



(a) OHLC representation with percentage resampled data.



(b) Chronological evolution of the ROI [%] for the 5 calculated trading signals and B&H strategy.

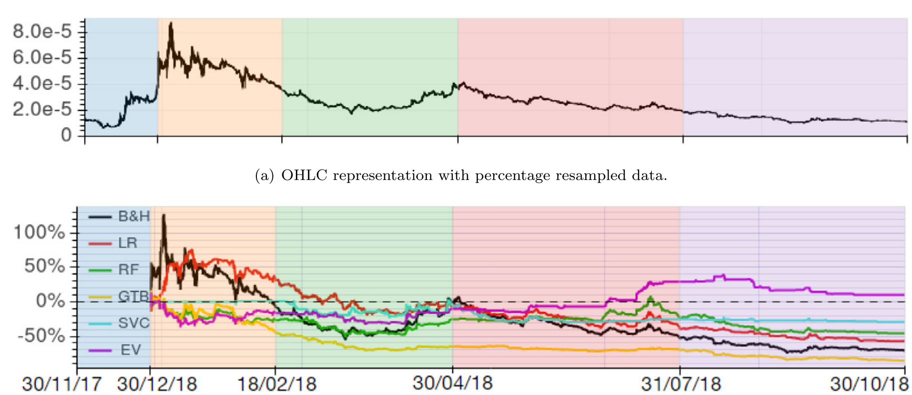
Fig. 8. ROI variations for currency pair POEETH (Po.et/Ethereum) with percentage resampling applied.

Table 6
Results obtained with time resampling for the POEETH and ADABTC markets.

Parameter	B&H	LR	RF	GTB	SVC	EV
POEETH market with time resampling						
Final ROI	−28.0%	**180%**	−15.6%	51.8%	39.7%	71.3%
Accuracy	39.02%	55.51%	58.75%	58.36%	50.99%	**60.06%**
Negative log-loss	−21.06	−0.6851	−0.6732	−0.6840	**−0.6660**	−0.6730
Periods in market	100%	60.9%	35.2%	41.4%	73.3%	**31.0%**
Profitable positions	0%	**63.5%**	52.2%	54.4%	55.3%	55.7%
Profit per position	−28.0%	**0.205%**	−0.02%	0.060%	0.048%	0.102%
Largest gain	NA	3.55%	21.3%	16.7%	**25.9%**	16.7%
Largest loss	−28.0%	−20.6%	−20.3%	**−7.7%**	−21.0%	−20.5%
Max drawdown	70.5%	27.14%	63.51%	**19.16%**	57.27%	26.93%
Annual sharpe ratio	0.222	**2.190**	−0.057	1.174	0.900	1.471
Annual sortino ratio	0.368	**3.405**	−0.152	1.869	1.383	2.130
ADABTC market with time resampling						
Final ROI	−76.3%	−72.6%	−50.3%	−44.4%	−67.1%	**−34.1%**
Accuracy	35.54%	45.31%	55.21%	52.78%	43.41%	**57.22%**
Negative log-loss	−22.26	−0.7053	−0.6905	−0.6925	**−0.6632**	−0.6826
Periods in market	100%	80.0%	50.7%	54.2%	84.7%	**44.5%**
Profitable positions	0%	48.3%	44.8%	39.1%	49.1%	**51.2%**
Profit per position	−76.3%	−0.19%	−0.13%	−0.13%	−0.21%	**−0.08%**
Largest gain	NA	8.12%	10.8%	**21.8%**	19.4%	10.8%
Largest loss	−76.3%	−20.9%	**−14.4%**	−20.4%	−15.4%	−15.2%
Max drawdown	79.30%	74.20%	57.61%	53.87%	71.34%	**45.58%**
Annual sharpe ratio	−2.057	−2.350	−1.541	−1.051	−1.842	**−0.872**
Annual sortino ratio	−3.005	−3.372	−2.370	−1.719	−2.743	**−1.380**

Table 7
Average obtained results for the Buy & Hold and each of the five methodologies employed for percentage resampling.

Parameter	B&H	LR	RF	GTB	SVC	EV
Average obtained results (all markets are considered)						
Final ROI	16.4%	792%	494%	692%	**1063%**	923%
Accuracy	41.38%	53.03%	53.00%	53.06%	52.64%	**55.23%**
Negative log-loss	−20.2	−0.7112	−0.7028	−0.6928	−0.7187	**−0.6866**
Periods in market	100%	56.7%	55.8%	58.0%	52.3%	**43.3%**
Profitable positions	25.0%	**61.9%**	55.2%	58.0%	59.7%	59.0%
Profit per position	**16.4%**	0.82%	0.51%	0.67%	1.17%	0.97%
Largest gain	**56.3%**	20.9%	15.0%	16.3%	19.1%	13.6%
Largest loss	−39.9%	−13.2%	−14.2%	−14.2%	−13.9%	**−12.7%**
Max drawdown	80.5%	56.6%	59.9%	60.5%	55.4%	**50.1%**
Annual sharpe ratio	0.066	1.297	0.868	0.824	1.360	**1.496**
Annual sortino ratio	0.605	3.907	2.911	2.689	3.765	**4.325**



(a) OHLC representation with percentage resampled data.

(b) Chronological evolution of the ROI [%] for the 5 calculated trading signals and B&H strategy.

Fig. 9. ROI variations for currency pair ADABTC (Cardano/Bitcoin) with percentage resampling applied.

Fig. 10. Entry and exit points for the TRXBTC (Tron/Bitcoin) pair during 17 and 18 January 2018 for percentage resampling.

Fig. 11. Entry and exit points for the IOTABTC (Internet of Things Application/Bitcoin) pair from mid September 26 until mid October 2 2018 for percentage resampling.

Table 8
Results obtained with percentage resampling for the POEETH and ADABTC markets.

Parameter	B&H	LR	RF	GTB	SVC	EV
POEETH market with percentage resampling						
Final ROI	64.2%	1436%	850%	3145%	**5747%**	4416%
Accuracy	42.59%	56.60%	52.33%	**56.71%**	53.92%	56.54%
Negative log-loss	−19.83	−0.6860	−0.7058	−0.6894	**−0.6824**	−0.6859
Periods in market	100%	46.5%	64.4%	**43.8%**	59.5%	44.9%
Profitable positions	100%	60.9%	54.1%	61.0%	**62.3%**	61.3%
Profit per position	**64.2%**	1.6%	1.0%	3.7%	2.1%	4.9%
Largest gain	**64.2%**	2.6%	2.5%	4.8%	6.4%	7.3%
Largest loss	NA	**−2.2%**	−8.9%	−3.7%	−8.2%	−4.3%
Max drawdown	89.98%	**16.60%**	54.60%	20.03%	70.94%	23.92%
Annual sharpe ratio	1.041	3.969	2.661	3.868	3.404	**3.970**
Annual sortino ratio	2.510	9.326	6.125	10.192	10.829	**11.927**
ADABTC market with percentage resampling						
Final ROI	−71.1%	−56.8%	−45.8%	−85.7%	−29.1%	**10.1%**
Accuracy	40.0%	52.04%	54.88%	50.24%	55.97%	**59.02%**
Negative log-loss	−20.72	−0.6921	−0.6903	−0.6941	**−0.674**	−0.6837
Periods in market	100%	58.4%	52.5%	61.9%	**23.6%**	29.4%
Profitable positions	0%	**58.6%**	47.7%	48.4%	47.1%	53.8%
Profit per position	−17.1%	−0.14%	−0.08%	−0.20%	−0.19%	**0.03%**
Largest gain	NA	9.5%	9.4%	**14.4%**	12.5%	9.2%
Largest loss	−71.1%	−20.3%	−11.9%	−24.2%	**−8.8%**	−11.8%
Max drawdown	88.64%	75.77%	57.4%	87.7%	**32.7%**	35.7%
Annual sharpe ratio	−1.483	−1.190	−1.028	−3.168	−1.013	**0.471**
Annual sortino ratio	−2.303	−1.812	−1.637	−4.061	−1.682	**0.662**

Fig. 8 contains the example of a market where this system performed well. From this figure and Table 8 it can be concluded that the signal from EV only obtained top results for the risk ratios. Results regarding ROI and especially predictive power are slightly inferior to the top values. In this example the top values are dispersed throughout the different trading signals, which goes to show that when utilizing varied learning algorithms, one's weakness can be overcome by another learning algorithm and, ultimately only the best traits of these algorithms are hopefully noticeable in the EV, which in this case, did not happen.

Fig. 9, on the other hand, contains the same market who underperformed from the previous case study. From this figure and Table 8 it can be concluded that EV obtained top predictive results with a low risk, and even achieved a positive ROI, contrarily to the remaining trading signals. In fact, almost the opposite to what happened in the previous market is verified here. The trading signal from EV outdid most metrics of the remaining trading signals. This occurrence does not happen every time, but is one potentiality of the EV. Obviously the contrary, where the trading signal produced by EV is the worst out of all, does happen as well, but it is a much rarer occurrence in this system as can be confirmed by the tables containing the average overall results for any resampling method, where the EV trading signal on average excels the remaining.

By now, it is evident that this resampling method is able to generate on average much higher ROIs than time resampling. However, in order to achieve the more thorough analysis of this work's system, specific entry and exit points from periods with different characteristics are represented in Figs. 10 and 11. Fig. 10 contains a volatile period of the cryptocurrency pair TRXBTC. Here it is visible that the system is mostly inside the market as the overall trend is bullish. Fig. 11, on the other hand, contains a relatively calm period of the cryptocurrency pair IOTABTC. By comparing the two figures, it is clear that the system is more inactive during calm intervals, which is in accordance with the intention of taking advantage of highly volatile situations.

4.2.4. Amount resampling

This case study contains an analysis over the results obtained when the system utilizes amount resampled data. This type of resampling procedure generates on average the best performance in terms of the ROI metric for the trading signals generated by EV. Hence, similarly to the previous case study, a more in depth analysis will be carried out. Firstly, a temporal graph showing the instantaneous average ROI of the 100 analysed markets for each of the methodologies is represented in Fig. 12. In this figure, similarly to percentage resampling, a slight drop is still visible from around May until September 2018. This resampling method generates an overall higher ROI relatively to the percentage resampling method. Once again, the B&H signal was clearly surpassed by all remaining trading signals in terms of returns.

Secondly, a table containing the general statistics obtained for amount resampling method follow in Table 9. In this table, yet again, the trading signal produced by EV clearly outperforms the remaining methodologies. This method obtains the highest profits, the best predictive power and the lowest risk out of all available trading signals. The B&H strategy visibly performed the worst as usual. Relatively to the previous case studies, it may be concluded that on average, this resampling method generates the highest profits, but is riskier than both logarithmic amount and percentage resampling.

Thirdly, the results for two specific markets are represented in Figs. 13 and 14 and their statistics are shown in Table 10. Likewise, these figures follow the same structure and contain the

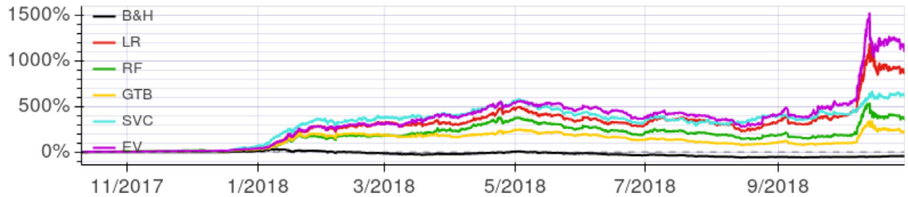


Fig. 12. Average accumulated ROI [%] for each instant in the test dataset with amount resampled data.

Table 9
Average obtained results for the Buy & Hold and each of the five methodologies employed with amount resampled data.

Parameter	B&H	LR	RF	GTB	SVC	Ensemble voting
Average obtained results (all markets are considered)						
Final ROI	−46.92%	859.1%	354.2%	215.5%	617.8%	**1100.3%**
Accuracy	40.84%	53.24%	53.12%	53.20%	53.58%	**55.61%**
Negative log-loss	−20.4	−0.7021	−0.7012	−0.6921	−0.6868	**−0.6839**
Periods in market	100%	57.75%	55.28%	55.56%	50.96%	**44.41%**
Profitable positions	13.00%	**59.72%**	52.80%	55.49%	57.42%	57.10%
Profit per position	−46.92%	0.919%	0.357%	0.210%	0.671%	**1.22%**
Largest gain	8.66%	18.26%	16.08%	**18.94%**	19.12%	16.44%
Largest loss	−55.6%	−16.28%	−16.31%	−17.00%	−14.48%	**−14.10%**
Max drawdown	81.6%	61.0%	63.7%	65.3%	55.5%	**53.7%**
Annual sharpe ratio	−0.440	0.531	0.276	0.114	0.786	**0.816**
Annual sortino ratio	−0.411	1.500	1.410	0.744	2.479	**2.565**

same markets as the figures presented in the previous case studies. Fig. 13 and Table 10 contain the example of a market where this system performed notably well. Once more, the trading signal from the EV overcomes in all aspects the remaining signals. Nonetheless, it is observable that in this resampling method worse results were obtained for this market relatively to when percentage resampled data was used. The fact that lower values were obtained can be up to some degree blamed on the lower predictive performance and the higher riskiness, as can be seen by comparing the metrics from this table with Table 8. Fig. 14, on the other hand, contains the market who underperformed. From these figures and their respective values in Table 10, a conclusion similar to the one taken in the previous case study may be taken. The SVC's signal performance, one more time, exceeded the remaining trading signals. However, tn this case, similarly to the logarithmic amount case study, the EV was unable of excelling both SVC and the remaining trading signals.

For comparison purposes, the entry and exit points of the same periods and markets used in the previous case study, but with data resampled according to an amount, are represented in Figs. 15 and 16. Fig. 15 contains a highly volatile period of the cryptocurrency pair TRXBTC. Relatively to the percentage case study (Fig. 10), it is clear that with amount resampled data, the trading signal typically does not keep the same long position active for as much time. With this type of resampling, relatively to percentage resampling, the triggered long positions are of shorter durations. Therefore, this strategy is prone to sustain heavier losses, namely due to transfer fees, which explains the larger risk verified. Fig. 16, on the other hand, contains a relatively calm period of cryptocurrency pair IOTABTC. Note that all candles have durations of above 2 h due to the reduced price variation. The largest variation seen in this figure is of around 5% in the course of 12 h, this same pair earlier in 2018 on a regular basis had this same percentual variation in a matter of minutes. Once again, the judgement that this system is clearly more active in volatile moments is upheld.

4.2.5. Results and discussion

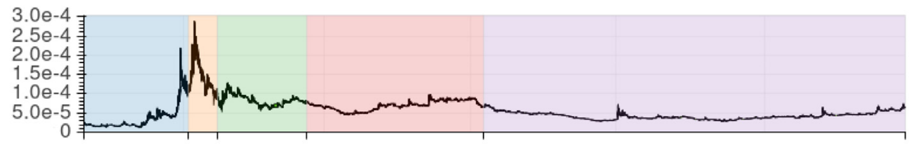
First of all, like all works mentioned in the state-of-the-art, the results obtained in this work were obtained through back-test simulations, hence all the characteristic limitations of this technique also apply to this work [4]. Nevertheless, the system developed in this work attempts to reproduce as closely as possible a realistic environment in order to extrapolate its results into an actual active trading system.

The obtained accuracies in this work are on par or, for the case of ensemble voting, exceed the accuracy of most papers regarding cryptocurrency exchange market forecasting throughout the state-of-the-art (Section 2 and Table 1). The risk measures and returns on investment are also superior on average. Only Mallqui and Fernandes [34] obtained an absolute superior accuracy. However, contrarily to testing only one specific market, this system was tested against 100 different markets. The levels of accuracy seen in Mallqui and Fernandes's work were exceeded by a few markets tested in this work (as can be discerned by observing Table 5).

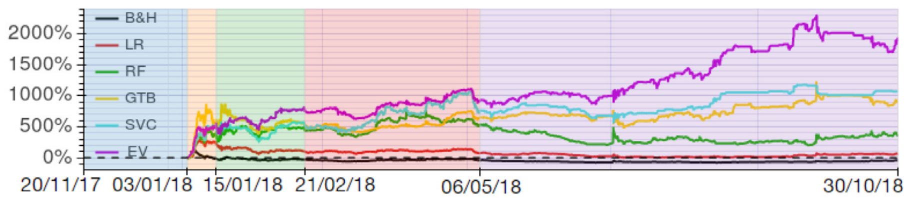
In relation to the different resampling procedures, Table 11 contains a comparison showing how frequently the results from the two alternative resampling procedures exceed the results obtained with time resampling. It is worth noting the values displayed in this table were obtained by comparing the same given market and respective machine learning algorithm among the different resampling procedures. All markets and machine learning algorithms were taken into account to compute the values shown in this table.

Through analysis of the results obtained and Table 11, the following observations can be made:

- Relatively to the results provided by the time resampled procedure it can be concluded that the developed strategy is not flawlessly suited for this type of resampling. Neither B&H nor any of the five remaining models achieved absolutely brilliant results. In any case, it may be concluded the EV methodology is most definitely the superior one. It is worth praising the high predictive performances as can be perceived by the high accuracies and NLLs, particularly with the trading signals provided by EV, implying that this system has predictive potential for this resampling method. This suggests that if this same machine learning formulae were to be combined with a more fine-tuned strategy, there is potential to achieve more impressive results, risk and return-wise.

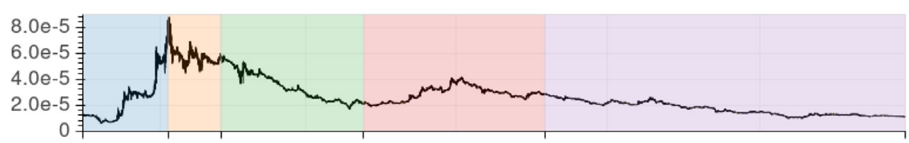


(a) OHLC representation amount resampled data.

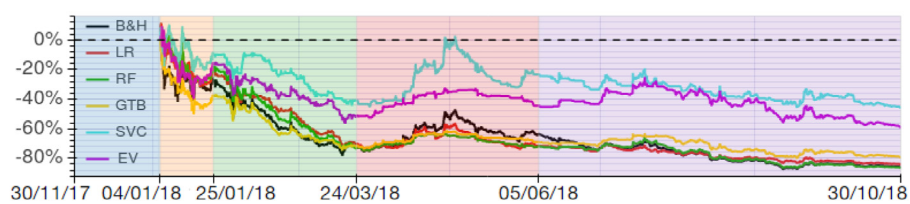


(b) Chronological evolution of the ROI [%] for the 5 calculated trading signals and B&H strategy.

Fig. 13. ROI variations for currency pair POEETH (Po.et/Ethereum) with amount resampling applied.



(a) OHLC representation with amount resampled data.



(b) Chronological evolution of the ROI [%] for the 5 calculated trading signals and B&H strategy.

Fig. 14. ROI variations for currency pair ADABTC (Cardano/Bitcoin) with amount resampling applied.

Table 10
Comparison between the Buy & Hold strategy and each of the five methodologies employed for the amount resampling in POEETH and ADABTC markets.

Parameter	B&H	LR	RF	GTB	SVC	Ensemble voting
POEETH market with amount resampling						
Final ROI	−39.35%	67.34%	367.7%	904.0%	1070.0%	**1930.8%**
Accuracy	42.31%	52.96%	52.41%	56.33%	54.90%	**57.18%**
Negative log-loss	−19.92	−0.6881	−0.6970	−0.6870	**−0.6793**	−0.6817
Periods in market	100%	63.4%	67.3%	49.4%	52.9%	**45.9%**
Profitable positions	0%	56.9%	53.4%	57.7%	58.6%	**59.2%**
Profit per position	−39.35%	0.060%	0.39%	0.98%	1.2%	2.0%
Largest gain	NA	4.11%	2.4%	4.1%	3.7%	**11.6%**
Largest loss	−39.35%	−16.7%	−7.7%	−0.84%	**−0.66%**	−1.7%
Max drawdown	90.3%	75.3%	65.0%	37.1%	40.4%	**24.9%**
Annual sharpe ratio	0.2394	0.9534	1.855	2.051	2.799	**3.092**
Annual sortino ratio	0.4224	1.861	4.854	7.718	7.222	**11.979**
ADABTC market with amount resampling						
Final ROI	−85.87%	−84.06%	−86.45%	−79.24%	**−45.78%**	−58.85%
Accuracy	39.37%	51.28%	49.51%	51.23%	52.96%	**54.66%**
Negative log-loss	−20.94	−0.6916	−0.7176	−0.6935	**−0.6732**	−0.6882
Periods in market	100%	70.0%	71.1%	60.9%	64.3%	**59.6%**
Profitable positions	0%	**52.1%**	46.6%	49.3%	50.4%	50.7%
Profit per position	−85.87%	−0.135%	−0.213%	−0.200%	−0.084%	**−0.113%**
Largest gain	NA	12.2%	13.5%	10.2%	**16.4%**	15.4%
Largest loss	−85.87%	−12.9%	−21.6%	−11.5%	**−8.4%**	−9.7%
Max drawdown	88.58%	85.5%	87.5%	79.9%	**50.6%**	61.9%
Annual sharpe ratio	−2.224	−2.802	−3.018	−2.475	**−0.960**	−1.394
Annual sortino ratio	−3.197	−3.845	−4.025	−3.341	**−1.645**	−2.015

- Looking at the results obtained by the percentage resampled data, it may be concluded that return and risk-wise it is clearly preferable relatively to the time resampled procedure. Nonetheless, even though the returns were plainly superior, the predictive power is inferior.

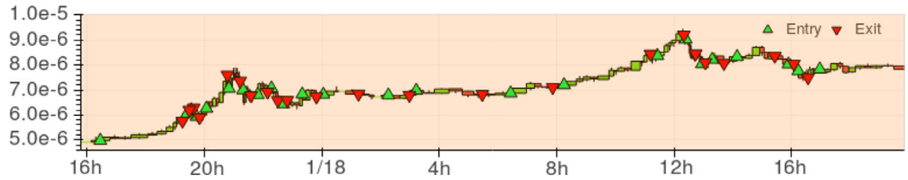


Fig. 15. Entry and exit points for the TRXBTC (Tron/Bitcoin) pair during most of 17 and 18 January 2018 for amount resampling.

Fig. 16. Entry and exit points for the IOTABTC (Internet of Things Application/Bitcoin) pair from mid September 26 until mid October 2 2018 for amount resampling.

Table 11
Proportion of markets resampled according to alternative methods that surpass the results of time resampling (out of 500 market-Machine learning algorithm combinations).

Alternative resampling:	Percentage	Amount
Markets that obtained ROI higher than time resampling	69.6%	61.4%
Markets that obtained ROI at least 10x higher than time resampling	12.2%	55.4%
Markets that obtained Sharpe ratio higher than time resampling	72.4%	63.8%
Markets that obtained Sortino ratio higher than time resampling	72.8%	63.8%

- Finally, the results generated by the amount resampled data, once again, clearly outperform the time procedure return and risk-wise, but the predictive power is certainly inferior, likewise the percentage procedures. Relatively to percentage resampled data, generally this resampling procedure yields a better ROI, in fact almost all markets who acquired a higher profit relatively to time resampling got a profit at least 10 times higher. This type of resampling also got a slightly superior predictive power, yet this procedure clearly has a higher risk associated when compared to percentage resampling.
- Regarding the fact that, time rearrangement on average, holds the lowest ROI, but contradictorily also obtains the best accuracies and negative log-losses out of all the resampling methods employed, a possible explanation is because most investors utilize similar investment strategies and prediction algorithms based on similarly time sampled data. Furthermore, market manipulation is entirely permitted in cryptocurrency exchange markets. Because the algorithms and strategies utilized by investors are related and due to their collective influence, the current ongoing time series is constantly being heavily impacted. Hence, when an algorithm similar to the ones who collectively crafted and whose influence is deeply embedded in a given time series intends to backtest trade with this same time series, it seems plausible that its overall predictive power is enhanced. Nevertheless, to assign this phenomenon as the unambiguous source of this issue would require further investigation.

Independently of the resampling method, this system is clearly more active in volatile periods than calm periods as was pretended. Throughout these case studies, it was verified that specific markets, such as ADABTC, will invariably yield losses independently of the resampling procedure. A portfolio management could possibly withdraw these markets from the list of tradeable markets in order to prevent these types of losses. On the upside, it was verified that if this work's system was able of turning a profit with a time resampled financial series, most likely, the same given market attains far larger returns with any of the two alternative resampling procedures. One important takeaway worth noting is that, independently of the utilized strategy, higher accuracies or NLL do not inevitably translate into a higher ROI or a lower risk. To conclude, as a rule, both alternative resampling procedures consistently provide a higher ROI relatively to time resampling, suggesting that financial time series resampled in accordance to an alternative metric, rather than time, do in fact have the potential of creating a signal more prone to earn larger profits with less risk involved.

5. Conclusion

In this work a system combining several machine learning algorithms with the goal of maximizing predictive performance was described, analysed and compared to a B&H strategy. All aspects of this system, namely the target formulation, were designed with the objectives of maximizing returns and reducing risks always in mind. To validate the robustness of this system's performance, this system was tested on 100 cryptocurrency exchange markets through backtest trading.

Based upon this work, it may be concluded that all four distinct learning algorithms consistently bared positive results, better than random choice or the B&H strategy. Nonetheless, the trading signal generated by the ensemble voting method produced by far the best results.

Independently of the utilized learning algorithm, the outcome of utilizing data resampled according to alternative metrics, namely data resampled according to a fixed percentage as well as a fixed amount, proved to generate significantly higher returns than the commonly used time sampled data. The overall results were in accordance with what was anticipated: rearranging financial time series according to an alternative metric does in fact construct a series prone to generating larger returns, which is in accordance with one of the main intentions of this work.

It is worth noting that the return on investment obtained in this system with the alternative procedures is in some markets excessive. This fact is attributed to the absence of bid–ask spread data, as was mentioned in Section 3. Putting it differently, this work shows that the three alternative methods offer more profitable ROIs relatively to time resampling, however, it is unlikely that this system would do as well, return-wise, in an actual real scenario in face of bid–ask spreads.

In future work, among other improvements, the following suggestions seem to be the most promising: incorporation of bid–ask spreads in the system to have a more realistic system; in this work static resampling thresholds were applied, but these could be dynamically adjusted as a function of specific indicators; an alternative set of technical indicators or indicators (such as fundamental indicators or data extracted from social media) could be experimented with; the utilized set of machine learning algorithms could be perfected with the aid of statistical testing and additional machine learning procedures namely evolutionary algorithms should be tested in an attempt to address the shortcomings of the ones applied.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2020.106187.

Acknowledgement

This work is funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/EEA/50008/2020. .

References

[1] G. Hileman, M. Rauchs, Global Cryptocurrency Benchmarking Study, Cambridge Centre for Alternative Finance, Vol. 33, 2017.
[2] E. Hadavandi, H. Shavandi, A. Ghanbari, Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting, Knowl.-Based Syst. 23 (8) (2010) 800–808.
[3] J. Nadkarni, R.F. Neves, Combining neuroevolution and principal component analysis to trade in the financial markets, Expert Syst. Appl. 103 (2018) 184–195.
[4] M.L. De Prado, Advances in Financial Machine Learning, John Wiley & Sons, 2018.
[5] S. Nakamoto, et al., Bitcoin: A Peer-to-Peer Electronic Cash System, Working Paper, 2008.
[6] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
[7] T. Klein, H.P. Thu, T. Walther, Bitcoin is not the New Gold–A comparison of volatility, correlation, and portfolio performance, Int. Rev. Financ. Anal. 59 (2018) 105–116.
[8] J.J. Murphy, Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications, Penguin, 1999.
[9] R.C. Cavalcante, R.C. Brasileiro, V.L. Souza, J.P. Nobrega, A.L. Oliveira, Computational intelligence and financial markets: A survey and future directions, Expert Syst. Appl. 55 (2016) 194–211.
[10] Y.S. Abu-Mostafa, A.F. Atiya, Introduction to financial forecasting, Appl. Intell. 6 (3) (1996) 205–213.
[11] D.W. Hosmer Jr, S. Lemeshow, R.X. Sturdivant, Applied Logistic Regression, Vol. 398, John Wiley & Sons, 2013.
[12] S. Raschka, Python Machine Learning, Packt Publishing Ltd, 2015.
[13] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
[14] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[15] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[16] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, in: Springer Series in Statistics, Springer New York, 2009.
[17] G. Louppe, Understanding random forests: From theory to practice, 2014, arXiv preprint arXiv:1407.7502.
[18] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794.
[19] Y. Freund, R.E. Schapire, et al., Experiments with a new boosting algorithm, in: Icml, Vol. 96, Citeseer, 1996, pp. 148–156.
[20] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[21] J.H. Friedman, Stochastic gradient boosting, Comput. Stat. Data Anal. 38 (4) (2002) 367–378.
[22] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.
[23] R. Mitchell, E. Frank, Accelerating the xgboost algorithm using GPU computing, Peer J. Comput. Sci. 3 (2017) e127.
[24] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[25] H.-T. Lin, C.-J. Lin, R.C. Weng, A note on Platt's probabilistic outputs for support vector machines, Mach. Learn. 68 (3) (2007) 267–276.
[26] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, Chapman and Hall/CRC, 2012.
[27] J.a.M.R. Cardoso, R. Neves, Investing in credit default swaps using technical analysis optimized by genetic algorithms, 2017.
[28] Z. Jiang, J. Liang, Cryptocurrency portfolio management with deep reinforcement learning, in: 2017 Intelligent Systems Conference, IntelliSys, IEEE, 2017, pp. 905–913.
[29] M. Nakano, A. Takahashi, S. Takahashi, Bitcoin technical trading with artificial neural network, Physica A 510 (2018) 587–609.
[30] S. McNally, J. Roche, S. Caton, Predicting the price of Bitcoin using Machine Learning, in: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP, IEEE, 2018, pp. 339–343.
[31] A. Greaves, B. Au, Using the bitcoin transaction graph to predict the price of bitcoin, 2015, No Data.
[32] J. Yao, C.L. Tan, A case study on using neural networks to perform technical forecasting of forex, Neurocomputing 34 (1–4) (2000) 79–98.
[33] K. Żbikowski, Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy, Expert Syst. Appl. 42 (4) (2015) 1797–1805.
[34] D.C. Mallqui, R.A. Fernandes, Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques, Appl. Soft Comput. 75 (2019) 596–606.
[35] E. Akyildirim, A. Goncu, A. Sensoy, Prediction of cryptocurrency returns using machine learning, 2018.
[36] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 161–168.
[37] L.A. Teixeira, A.L.I. De Oliveira, A method for automatic stock trading combining technical analysis and nearest neighbor classification, Expert Syst. Appl. 37 (10) (2010) 6885–6890.
[38] C.D. Kirkpatrick II, J.A. Dahlquist, Technical Analysis: The Complete Resource for Financial Market Technicians, FT Press, 2010.
[39] S.B. Achelis, Technical Analysis from A to Z, McGraw Hill New York, 2001.
[40] R.J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, OTexts, 2018.
[41] A. Defazio, F. Bach, S. Lacoste-Julien, SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives, in: Advances in Neural Information Processing Systems, 2014, pp. 1646–1654.
[42] M. Magdon-Ismail, A.F. Atiya, An Analysis of the Maximum Drawdown Risk Measure, Citeseer, 2015.
[43] B.G. Malkiel, E.F. Fama, Efficient capital markets: A review of theory and empirical work, J. Finance 25 (2) (1970) 383–417.