Machine learning in finance: forecasting and trading

Maxime LELIEVRE   Tom MERY   Matteo PEDUTO

***Abstract*—Over the last few years, the cryptocurrencies became increasingly popular among the investors despite their sometimes high volatiliy. Likewise, machine learning has become a popular tool for improving decision making in financial markets.This paper presents the implementations and performances' comparison of different machine learning models on several assets. The results suggest that the combination of models optimized on different assets of the same type reaches the best trading performances.**

I. Introduction

Over the last few years, the cryptocurrencies became increasingly popular as an investment product and for a portfolio diversification strategy. In the period from February 2016 to December 2022, the estimated market capitalization of all cryptocurrencies passed from 27 to 800 billion USD and the 24 hour average trading volume of all cryptocurrencies already reached more than 100 billion USD in 2021. A still increasing body of literature focused on the pertinence of the efficient market hypothesis (EMH), as proposed by Fama (1970) [1], on the famous Bitcoin, see for example Urquhart (2016) [2], Nadarajah and Chu (2017) [3], Bariviera (2017) [4], Sensoy (2018) [5] and Wildi et al.(2019) [6]. In essence, the EMH postulates that efficient markets reflect all past, public or public and private information in market prices. Verification of the EMH is important for market participants as it implies that such information cannot be used to make persistent profits on trading on the market. In summary, recent research on the topic is inconclusive as to whether Bitcoin markets are efficient under the EMH or not. Wildi et al.(2019) results suggest that Bitcoin markets are becoming less rather than more efficient towards the sample end of their data (2019).

In this context, we propose in the continuation of Wildi et al.(2019) to extend their approach to other cryptocurrencies and to commodities and market indices to see weather positive trading performances can be achieved with machine learning models.

We here implement four different machine learning methods applied on three types of assets (cryptocurrencies, commodities and market indices) with three different assets for each. We further analyze weather the combination of several models trained on one data set gives better results than the results of the best model only and weather the combination of the best models trained on several assets if the same type gives better results than the results of the best model trained exclusively on the asset. Therefore we test their performances with several trading performance metrics.

After a presentation of the used data, the pre-processing step and an explanation of some financial prerequisites, we start our analysis by unfolding the four machine learning methods implemented and then the results are discussed.

II. Data analysis

A. Data collection

The project thus focuses on nine different assets divided in three categories (cryptocurrencies, commodities and market indices). The same time period has been considered with respect to the category.

	Cryptocurrencies	Commodities	Market Indices
Assests 1	Bitcoin	Gold	S&P 500
Assests 2	Ethereum	Natural Gas	CAC40
Assests 3	Ripple	Oil	SMI
Period considered	2017-11-09 to 2022-12-13	2012-12-13 to 2022-12-09	2012-12-13 to 2022-12-12

TABLE I
THE NINE ASSETS DIVIDED BY TYPE

The collection of data has been performed on different web sites herafter detailed with the assets' symbol used in the code.
Bitstamp [7]: Bitcoin (BTC-USD), Ethereum (ETH-USD), Ripple (XRP-USD)
Nasdaq [8]: Natural Gas (NYMEX-NG), Gold (LBMA-GOLD), Oil (OPEC-ORB), S&P500 (SP500)
Yahoo Finance [9]: CAC40 (CAC40), SMI (SMI)

In time series machine learning, the data cannot be split in any ways. In fact, it is important to maintain the chronological aspect. The train set must be observed before in time than the test set. This characteristic makes complicated the cross validation process. This is why the data has been split chronologically. The first half for the training, the third quarter for the validation and the last quarter for the testing with the validation set used to tune the hyper-parameters.

B. Data pre-processing

The different data sets extracted give several information about a assets' value on each day (open, close, adjusted close prices,...). The models are trained to forecast the log-returns of the open price of the assets as it is the most realistic one if one wants to take position based on the model's prediction. The open price reflects the price at which an asset first trades when the market opens. The log return at time t is computed as follow:

$$r_t = log\left(\frac{P_t}{P_{t-1}}\right) \tag{1}$$

Going through the log return allows to transform the time series such that the resulting financial data become more stationary from a temporal perspective, meaning even if we shuffle the data order, we will still be able to properly train the model and achieve successful test performance. It also allows to determine right away if the asset's price has increased or decreased over the day, by checking the sign of the log return, which is what matters in a trading perspective.

Moreover, the models are using a determined number of lags. This latter represents the number of consecutive log returns considered to predict the value of the log return and thus the asset's log-return at time t+1. The number of lags is determined by applying the auto-correlation function (ACF) on the log returns, see Fig.1. The ACF defines how data points in a time series are related, on average, to the preceding data points. In other words, it measures the self-similarity of the signal over different delay times. This is performed because, under the assumption that the time-series is gaussian and stationary, a linear forecasting model should reach a better accuracy if a pattern can be identified.

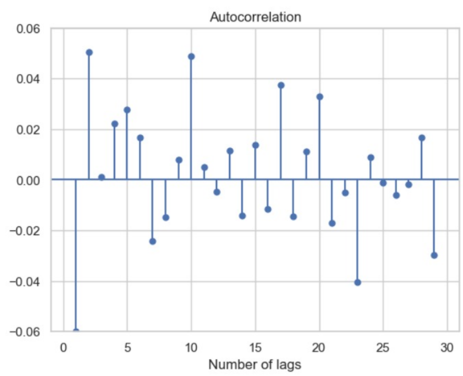

Fig. 1. Auto-correlation function applied on the log-returns of Bitcoin

Thus the data set is first transformed with the log returns before being split between the train, validation and test set. Finally, it is gathered in sequences of a chosen number of lags. For example, for 6 lags, one input is a sequence of 6 consecutive log returns and the output is the following log return.

III. Financial prerequisites

Before unfolding the implemented models, some financial basics must be explained to ensure the reader's understanding of the methodology used. Trading is the buying and selling of financial instruments in order to make a profit. These instruments range from a variety of assets that are assigned a financial value that goes up and down – and you can trade on the direction they take. Trading is opposed to investing, which suggests a buy-and-hold strategy and we will use it to evaluate our trading strategy. A good trade then consists of correctly predicting if the financial asset's value will increase or decrease and take the corresponding action. If one predicts it to increase, one will buy the asset to sell it later at a higher price. Conversely, if one predicts the value to decrease, one will sell the asset -it is called to short- and buy it later for a lower price.

We here give more details about the efficient market hypothesis (EMH) mentioned in the introduction. The EMH postulates that efficient markets reflect all past information (weakform), public information (semi-strong form), or public and private information (strong form) in market prices. Some findings suggest that Bitcoin markets, while inefficient in their early days, transitioned into efficient markets recently. Others find support for the adaptive market hypothesis (AMH), an alternative theory that builds on evolutionary principles and assumes markets and market efficiency evolve over time. Verification of the EMH is important for market participants as it implies that such information cannot be used to make persistent profits on trading on the market.

The trading strategy followed during this analysis consists of selling or buying based on the predictions output by the implemented models in a horizon of one day. It is useless in financial prediction to measure a model performance through the value of the loss function used during the training (here MSE). Thus, the performances of the models are assessed using three financial metrics hereafter explained.

A. Hit rate

The hit rate can be considered as the accuracy of the models. In trading, the hit rate is typically defined as the number of winning or profitable trades over a period of time for a trading strategy, divided by the total number of trades over the same period, and expressed as a percentage. It is determined by checking the sign of the predicted log return with respect to the sign of the target, expressed as a log return too. So if the predicted and the target log-returns have the same sign then it is considered as a profitable trade.

$$Hit\ Rate = \frac{\sum(sign(output) \cdot sign(target) >= 0)}{len(output)} \qquad (2)$$

B. Annualized Sharpe ratio

The Sharpe ratio compares the return of an investment with its risk. Sharpe ratios above 1 are generally considered “good,” offering excess returns relative to volatility.

To calculate the Sharpe ratio, investors first subtract the risk-free rate R_f from the portfolio's rate of return R_p, often using U.S. Treasury bond yields as a proxy for the risk-free rate of return. In this analysis, it is set up to 0 for simplicity. Then, they divide the result by the standard deviation of the portfolio's excess return σ_p. For the annualized sharpe ratio, it is almost the same but the numerator is multiplied by the square root of the number of periods (365 for the cryptocurrencies and 252 for the commodities and market indices due to the fact that the markets of these two latest are open only during the working days).

$$Annualized\ Sharpe\ Ratio = \sqrt{nb\_periods}\frac{R_p - R_f}{\sigma_p} \qquad (3)$$

C. Maximum Drawdown

The maximum drawdown (MDD) is the maximum observed loss from a peak to a trough of a portfolio, before a new peak is attained. MDD is an indicator used to assess the relative riskiness of one's trading strategy versus another. A low MDD is preferred as this indicates that losses from investment were small.

$$MDD = \frac{Trough\ Value - Peak\ Value}{Peak\ Value} \tag{4}$$

IV. Models and Methods

This section unfolds the models and methods used to analyze weather positive trading performance can be achieved with machine learning. The first part of the analysis optimized the four methods, namely Neural Networks (NN), Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM) and Random Forest (RF), on every single asset. Optimization of machine learning models in financial trading is a complicated topic where a lot of research is still ongoing. In addition, the main discoveries remain secretes for obvious economical reasons. Thus, the goal of the optimization performed in this project was not to obtain the best absolute trading metrics performances. In fact, the analysis of the architecture of the methods -mentioned hereafter- was not the main focus of the tuning. The structure of the methods is then the same for each single asset and was determined with satisfying results for the trading metrics, and in order to have approximately the same number of parameters independently of the method (103 for NN, 109 for CNN, 91 for LSTM). It was found that the hyper-parameter that impacts the most the performance of a method with a determined architecture is the learning rate. Thus, the optimization is determined by finding the learning rate, via grid search for ten values evenly distributed in the interval [0.0001 ; 0.001], that gives the best hit rate on the validation set. It is, indeed, the most relevant metric in this project because the hit rate is often proportional to the sharpe ratio (it is usually not the case for strong variations) and the maximum drawdown mostly reflects the riskiness of one's overall trading strategy. It is also important to notice that the hit rate oscillates depending on the number of training's epochs. A convergence, however, is noticed if at least 500 epochs are computed, see Fig.2. The training is then performed on this number of epochs to optimize the different methods. For the random forest, the tuned hyper-parameter is the number of maximum features, tuned via grid search again, taken during the bootstrapping step. The computation of the hit rate value to compare is obtained as the median of 10 models for time's sake. However, for the testing, the performances are obtained as the median of 100 models. XXXXXXXXXXXXX

Once the best models found for each asset, the analysis focused on one asset per category, in particular Bitcoin for cryptocurrencies, gold for commodities and CAC40 for market indices. The implementations of four different methods on each time series previously mentioned end up with four different models per asset. The goal here is to analyze weather the combination of several models trained on the same data set with different accuracy allows to make better predictions compared to the predictions of the best of the four models. One could think that each model learns something from the data in a possible different way that could contribute to improve the predictions when the results are combined. So instead of decreasing the overall accuracy by combining models of lower accuracy with the best model, the combination will output better predictions.

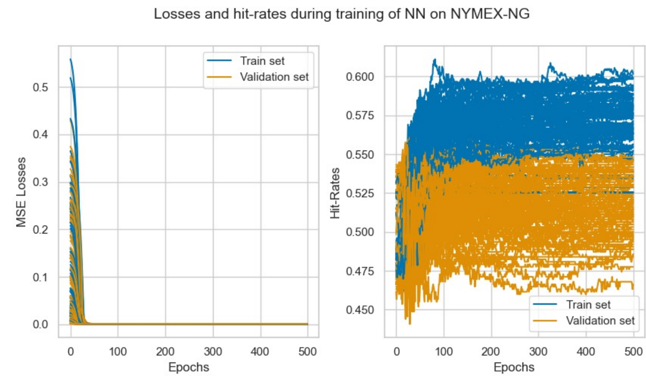

Fig. 2. Plot showing the convergence of the hit rate when computing 500 epochs

In a second part, the analysis focused on all the nine assets to investigate weather the models' predictions are improved when one combines the best model trained on the assets of the same type. In other words, if what is learnt on a different asset of the same type (cryptocurrencies, commodities or market indices) can improve the predictions on one asset in particular. For example, let's say model A is the best model for the market indices and one wants to trade on the SMI, one could wonder weather the combination of three models A each trained on CAC40, S&P500 and SMI respectively has a greater performance than the one of a model A only trained on SMI.

The log return of the combination of the models on each data set is then measured by taking the mean, the median and absolute maximum of the log returns of each model. The results reported in the Table II and III are the best values obtained between the latter three strategies.

The four methods implemented are explained hereafter. As stated before, the same architecture has been used no matter which assets was being predicted. The first section tries to reproduce the analysis of Wildi et al.(2019) on neural networks but applied on a broader time period. The three others analyze weather positive trading performances can be achieved with other methods.

A. Neural Networks

The neural networks implemented are non-linear feedforward neural networks with two hidden layers of dimensions six and three. Each node in the hidden layers corresponds to a neuron with Relu-activation function and the weights are updated with the mean-square error at each epoch. This architecture creates a model with 103 parameters.

B. Convolutional Neural Networks

To expand the analysis further, a convolutional neural network is implemented with two hidden layers before being flattened to end up with a fully connected neural network. There are two convolutional layers. The first one with 8 channels, a stride of 1, a kernel size of 3 and no-padding. The second one with 4 channels, a stride of 1, a kernel size of 1 and no-padding. The Relu-activation function is used after each convolutional layer and the weights are updated with the mean-square error at each epoch. This architecture creates a model with 109 parameters.

C. LSTM

A Long Short Term Memory (LSTM) algorithm is implemented. Two LSTM are concatenated and each algorithm has two hidden features. Here again, the weights are updated with the mean-square error at each epoch. This architecture creates a model with 91 parameters.

D. Random forest

A random forest regressor is implemented because of its ability to process large data sets while being very simple to implement and understand. In order to speed up the optimization, the number of trees is set to 1 000 for the optimization of the hyper-parameter and to 10 000 for the testing.

V. Results

The following Table II and Table III resume the hit rates and sharpe ratios of the best models for each asset as well as the comparison with the *Buy & Hold* strategy and two combinations of models. Combination 1 states for the combination of the 4 best models trained on the same asset. Combination 2 stands for the combination of the 3 best models trained on the asset of the same type. The results in bold represents the best model for each asset. The metrics of the combinations are highlighted in green when their performances are better than the best model alone. The buy-and-hold is in red when the models' and combination metrics performances could not reach higher values.

	B&H	*NN*	*CNN*	*LSTM*	*RF*	*C1*	*C2*
Bitcoin	0.473	0.457	0.446	**0.479**	0.459	0.451	0.505
Ethereum	0.490	0.457	**0.490**	0.481	0.475	0.488	0.497
Ripple	0.490	0.497	0.497	**0.532**	0.470	0.514	0.525
Nat. gas	0.516	0.518	**0.519**	0.494	0.498	0.505	0.502
Gold	0.493	0.502	0.511	0.477	**0.512**	0.509	0.514
Oil	**0.562**	0.492	0.508	**0.527**	0.522	0.525	0.536
S&P500	0.564	0.553	0.554	**0.558**	0.548	0.572	0.533
CAC40	**0.548**	0.520	0.526	0.528	0.498	0.504	0.520
SMI	0.535	0.507	**0.511**	0.506	0.488	0.517	0.546

TABLE II
Hit Rate comparison for each asset

VI. Discussion

The first result to notice is that positive trading performances can be achieved with one optimized model but it depends on the asset. CNN and LSTM are the ones that perform the best in general and random forest works well for gold, see Table II and III. The combination 1, which combines the best optimized models of one asset, performs worse than the best model alone, except for the S&P500. This can be explained by the fact that some model learn nothing new and thus worsen the overall performance of the combination. However, the combination 2 shows significant improvement compared to the best model. Indeed, for five assets out of nine, it is the one with the best performances. Here again it depends on the type of assets. For the cryptocurrencies and the commodities, the combination 2 performs the best in two cases out of three, and one out of three for the market indices. This might be explained by the difference of inter-dependence of the corresponding assets. The cryptocurrencies and the commodities have both a worldwide scale whereas the market indices depends on a region, USA for S%P500 and Europe for CAC40 and SMI.

	B&H	*NN*	*CNN*	*LSTM*	*RF*	*C1*	*C2*
Bitcoin	-1.163	-0.981	-2.066	**-0.140**	-2.020	-1.856	0.413
Ethereum	-0.870	-0.560	-0.327	**1.123**	0.245	0.308	-0.404
Ripple	-0.947	-0.143	0.521	**1.191**	-1.453	0.531	1.497
Nat. gas	0.743	0.523	**1.107**	-0.308	0.352	0.577	0.615
Gold	0.034	-0.056	-0.960	-0.468	**0.305**	-0.390	0.596
Oil	0.686	0.389	0.652	0.994	**1.012**	0.810	1.049
S&P500	0.518	**0.833**	0.801	0.768	0.090	1.136	0.551
CAC40	0.597	-0.192	**0.702**	0.499	-0.674	0.169	0.685
SMI	0.219	-0.588	**-0.321**	-0.367	-0.805	-0.400	0.522

TABLE III
Sharpe Ratio comparison for each asset

VII. Summary

To conclude this project, the analysis showed

Acknowledgements

The authors thank Marc Wildi for hosting our project in his lab and his helpful suggestions.

References

[1] Fama EF (1970) Efficient capital markets: A review of theory and empirical work. The Journal of Finance 25(2):383–417
[2] Urquhart A (2016) The inefficiency of bitcoin. Economics Letters 148:80 – 82
[3] Nadarajah S, Chu J (2017) On the inefficiency of bitcoin. Economics Letters 150:6 – 9
[4] Bariviera AF (2017) The inefficiency of bitcoin revisited: A dynamic approach. Economics Letters 161:1 – 42
[5] Sensoy A (2018) The inefficiency of bitcoin revisited: A high-frequency analysis with alternative currencies. Finance Research Letters
[6] Bundi N, Wildi M (2019) Bitcoin and Market-(In)Efficiency: a Systematic Time Series Approach.
[7] For Bitcoin, Ethereum and Ripple.
[8] For Natural Gas
For gold
For oil
For S&P500
[9] For CAC40 and SMI.