



HAL
open science

DEEP LEARNING AND TECHNICAL ANALYSIS IN CRYPTOCURRENCY MARKET

Stéphane Goutte, Viet Hoang Le, Fei Liu, Hans-Jörg Mettenheim, Von

► **To cite this version:**

Stéphane Goutte, Viet Hoang Le, Fei Liu, Hans-Jörg Mettenheim, Von. DEEP LEARNING AND TECHNICAL ANALYSIS IN CRYPTOCURRENCY MARKET. 2023. halshs-03917333

HAL Id: halshs-03917333

<https://shs.hal.science/halshs-03917333>

Preprint submitted on 1 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

DEEP LEARNING AND TECHNICAL ANALYSIS IN CRYPTOCURRENCY MARKET

Stéphane Goutte^b, Hoang-Viet Le^{a,b,*}, Fei Liu^c and Hans-Jörg von Mettenheim^{c,a,d}

^aKeynum Investments, France

^bUniversité Paris Saclay, UMI SOURCE, IRD, UVSQ, France

^cIPAG Business School, France

^dOxford-Man Institute of Quantitative Finance, United Kingdom

Abstract

A large number of modern practices in financial forecasting rely on technical analysis, which involves several heuristics techniques of price charts visual pattern recognition as well as other technical indicators. In this study, we aim to investigate the potential use of those technical information (candlestick information as well as technical indicators) as inputs for machine learning models, especially the state-of-the-art deep learning algorithms, to generate trading signals. To properly address this problem, empirical research is conducted which applies several machine learning methods to 5 years of Bitcoin hourly data from 2017 to 2022. From the result of our study, we confirm the potential of trading strategies using machine learning approaches. We also find that among several machine learning models, deep learning models, specifically the recurrent neural networks, tend to outperform the others in time-series prediction.

Keywords: Bitcoin, Technical Analysis, Machine Learning, Deep Learning, Convolutional Neural Networks, Recurrent Neural Network

*Corresponding author: Hoang-Viet Le, viet.le@keynum.fr

1 INTRODUCTION

Even since the success of Google’s AlphaGo against the Go World Champion in 2016, the keywords AI (Artificial Intelligence), machine learning, and deep learning have become the buzzwords that keep being mentioned in almost every industry. As a matter of fact, machine learning has already been slowly yet steadily transforming your everyday activities.

The finance industry, as expected, is one of the pioneers to incorporate AI technologies into its ecosystems. In recent years, more and more studies have been published that were based on the application of machine learning models on forecasting time series and many of them point out the relatively better performances of newer machine learning models compared to classical time series forecasting techniques[9; 4; 11]. Additionally, the widespread and easy-to-access automated electronic trading systems coupled with increasing demand for better alpha keeps forcing trading practitioners and researchers to put more effort into searching for better and more sophisticated models. Not until recently, the technique of deep learning (DL), a subset of machine learning which involves the application of neural networks, started to emerge as one of the best performing techniques within the field of machine learning in a vast majority of implementation areas. It is no exception in the case of financial time series forecasting as a growing number of deep-learning-based models of prediction have been introduced in several conferences and journals in recent years. Gu et al. (2020) used neural networks to forecast stock prices from the US market to find out their superiority over other models such as generalized linear models, principal components regressions, and regression trees[6]. Zhang et al. (2021) utilize the recurrent neural networks, specifically the long short-term memory (LSTM) networks, to predict stock trends report significant positive returns[12]. The work of Bianchi et al. (2021), on the other hand, focus on forecasting bond excess returns and also report significant economic gains[3]. Some other studies target other financial assets such as cryptocurrencies and also often point out the edge of the neural networks model on forecasting future price movements (Anghel, 2020[1]).

In this paper, we tried to put into practice those state-of-the-art deep learning models along with other popular machine learning models to see how well they perform in finding trading signals using the same inputs that was used in traditional technical analysis. While most of the other researches on this topic focus on the traditional stock markets, this paper choose the cryptocurrency market, specifically the Bitcoin market, as the target of research.

1.1 Data Source

Due to the borderless nature of cryptocurrency market, the prices of cryptocurrencies are not the same everywhere but vary between exchanges. In this research, we chose the Binance exchange as it is the largest cryptocurrency exchanges based on its trading volume as of August 2022. The Binance exchange publicly offers historical data dating back to 2017. Transaction details are stored continuously and are accessible at via its API. The exchange houses several sets of high-frequency data including limit order quotes and trading logs (or market order logs). Due to the goal of our analysis, only the trading logs (or trade data) are needed. Table 2 list out all the fields of the trading logs table and their meanings.

Trade data table records every single transaction that had taken place. Each row refers to a transaction processed by the Binance exchange and obviously, it is possible that many transaction can happen at the same time.

Table 1. Trade data table

	Field name	Type	Description
1	timestamp	datetime	The moment at which the transaction occurs, with precision to millisecond.
2	symbol	string	Ticker representing trading instrument.
3	side	string	The side of the trade, either Buy or Sell.
4	size	integer	The transaction size of the trade
5	price	float	The price that the trade was executed

As every transaction is recorded in this table, the total number of transactions is provided during our chosen period from the end of 2017 until August 2022 is upto a few billions. This is an enormous amount of data and is required to be processed for further analysis.

1.2 Data Preprocessing

The first step of the data preprocessing is to gather all of the needed data values from different data sources into a single collection of databases. In this study, as all of our data come from the historical trading database of Binance, there is no need for other sources of data. However, if any other types of alternative data are considered such as order book data, sentimental or fundamental data, etc., they should be included in this step. In this study, during these steps, several trivial data such as the ID of a single trade, type of order, etc. will also be removed. In the end, all of the trades that happened in every hour are resampled and grouped by their timestamps to create a new table of time series data with the following header: timestamps, open, high, low, close, number of trades and volume.

Table 2. The Preprocessed Trade data

	Field name	Type	Description
1	timestamp	datetime	The beginning time of the given period.
2	symbol	string	Ticker representing trading instrument, we only concern the ticker BTCUSD.
3	open	float	The latest price that happened before the beginning of the period
4	high	float	The maximum price that happened during the period
5	low	float	The minimum price that happened during the period
6	close	float	The latest price that happened before the end of the period
7	trades	integer	The total number of transactions happened during the period
8	volume	float	The total volume of currency that were traded during the period

After the resampling procedure, the length of our dataset is reduced to only 43,463 data points. For the purpose of training our machine learning models, the data will be divided into 3 parts: training set, validation set, test set with the proportion of 50%, 25%, and 25% respectively.

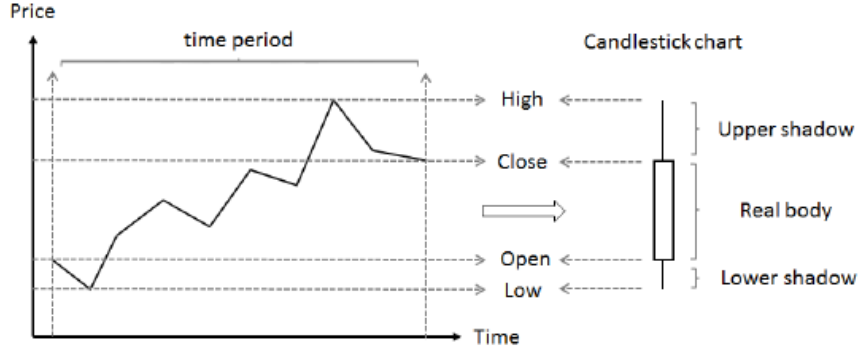
2 METHODOLOGY

2.1 Feature Engineering

2.1.1 Candlestick Data

Generally, the asset price data of a specific time period can be represented by four features: the open price (the price at the beginning of the period), close price (price at the end of the period), high price (the maximum price during the period), and low price (the minimum over the interval). However in technical analysis, the candlestick can be visually encoded in a different way (Figure 1): the candle body denotes the open and close prices, and the lines protruding from the bar (the candle's shadow) denote the extrema over the interval. Finally, the color of the bar implies the relative return of the period: a white (sometimes green) candle denotes a positive return over the interval (close price > open price) and a black (sometimes red) candle denotes a negative return (close price < open price).

Figure 1. Candle stick example



Source:[10]

The specific formulas to transform the price variables into a candlestick are as follows:

$$Color = \begin{cases} White & \text{if } Close - Open > 0 \\ Black & \text{if } Close - Open < 0 \end{cases} \quad (1)$$

$$Candle\ Body = |Close - Open| \quad (2)$$

$$Upper\ Shadow = \begin{cases} (High - Close) & \text{if } Close - Open > 0 \\ (High - Open) & \text{if } Close - Open < 0 \end{cases} \quad (3)$$

$$Lower\ Shadow = \begin{cases} (Open - Low) & \text{if } Close - Open > 0 \\ (Close - Low) & \text{if } Close - Open < 0 \end{cases} \quad (4)$$

2.1.2 Technical Indicators

Despite its popularity, in practice, candlestick patterns are often used in combination with several other qualitative indicators such as support and resistance, Fibonacci retracement, etc. or even popular statistical indicators like the moving average, etc. In our study, aside from the raw features

mentioned in the previous section, we also introduced an extended dataset that includes several popular technical analysis indicators to reflect the actual practice of technical analysis. The results from both the "raw" dataset and the "extended" data set will also be compared to see the effectiveness of technical indicators to the model.

In the extended dataset, we include simple moving average (SMA) and relative strength index (RSI) of multiple time steps, and other more complex indicators such as the moving average convergence/divergence oscillator (MACD), the Williams percent range (Williams %R), the stochastic oscillator, and the money flow index (MFI).

2.2 Data Labeling

In our study, our labels are chosen to be binary ($[Buy, Not Buy]$ or $[Sell, Not Sell]$) which are similar to trading signals used in technical analysis. The horizon of prediction is the direction of the next 3 hours or the next 3 candlesticks so that a trend can be formed. Finally, no threshold for the return is used so that positive return will be labeled as the signal for buy and vice versa.

Table 3. Data labeling

Dataset	Proportion	Long	Short
Training	50.00%	51.29%	48.71%
Validation	25.00%	51.54%	48.46%
Test	25.00%	49.72%	50.28%

2.3 Model Specifications

In this paper, seven popular machine learning algorithms were used which is the simple Logistic regression, two decision tree-based gradient boosting models (XGBoost and LightGBM), and four neural networks models (MLP, GRU, LSTM, and CNN).

For logistic regression model, we choose the logistic regression algorithm provided by the Scikit-learn library using the default specification with the L2 regularization to prevent overfitting. For both XGBoost and LightGBM, we used their official Python libraries. For this two algorithms, we use 100 trees as it is a popular parameter used in several pieces of researches [2; 8; 13; 7] and all others were set as default.

Regarding neural networks models, in this study, all of our models are built and trained using the Python library Keras running on top of the TensorFlow framework. The architecture of our neural networks will generally follow the work of Roberts, S. et al (2020)[5]. In his work, he built an MLP model and extended it to other deep learning models based on it. The architecture of the MLP model as follows:

Table 4. MLP architecture

#	Layer	Units	Activation Function	Dropout
0	Input	-	-	-
1	Dense	64	ReLU	0.5
2	Dense	64	ReLU	0.5
3	Dense	1	Sigmoid	-

For the 2 recurrent neural network models, LSTM and GRU, to keep the model to be comparable with the MLP model, the new recurrent layer with only 8 neurons will be added before the 2 fully-connected layers of the MLP. Table 5 illustrates the architecture of the LSTM and GRU model.

Table 5. LSTM/GRU architecture

#	Layer	Units	Activation Function	Dropout
0	Input	-	-	-
1	LSTM/GRU	8	ReLU	0.5
2	Dense	64	ReLU	0.5
3	Dense	64	ReLU	0.5
4	Dense	1	Sigmoid	-

Finally for the CNN, instead of the recurrent layers like LSTM or GRU, two convolutional layers are added along with two max-pooling layers and 0.5 dropouts each. Each of the convolutional layers has 8 neurons, similar to the recurrent neural network architecture.

Table 6. CNN architecture

#	Layer	Units	Activation Function	Dropout	Filter size	Padding
0	Input	-	-	-	-	-
1	Convolutional	8	ReLU	-	3	same
2	MaxPooling	-	-	0.5	2	same
3	Convolutional	8	ReLU	-	3	same
4	MaxPooling	-	-	0.5	2	same
5	Dense	64	ReLU	0.5	-	-
6	Dense	64	ReLU	0.5	-	-
7	Dense	1	Sigmoid	-	-	-

All of the neural network models are training in 500 training epochs with batch size equaling to 1024. Early stopping and model checkpoints based on the validation loss are also used to prevent overfitting so that we can get the best performing model on the validation set.

2.4 Model Evaluation

2.4.1 Precision and Recall

Precision and recall are also popular metrics that usually go together for evaluating classifier performance and will be used as the main metrics in this paper. The increases in either precision or recall are often achieved at the expense of the other, i.e. high precision is achieved at the expense of recall, and high recall is achieved at the expense of precision. Usually, an ideal model would love to have both high recall and high precision but this is usually hard to be achieved in practice. Due to the fact that a buy signal does not necessarily lead to a good return and investors usually look for a good return rather than normal return, in combination with the existence of transaction costs, a good trading signal system generally prefers a high precision with a reasonable recall.

2.4.2 Return and Average Return

As this study focuses on the feasibility of trading strategy based on several signal generators, it is always a good idea to include the financial metrics such as return into the study. For the sake of simplicity, we do not incorporate the transaction cost into the study. The return, therefore, will be calculated based on the compounded return formula:

$$Total\ Return = \prod_{i=1}^N (1 + R_i S_i) \quad (5)$$

where R_i is the return of every time step, and S_i is the binary signal generated by the model at each specific time step.

Using this total return as a metric, however, does not incorporate the transaction cost and therefore the higher return may come from many small trades instead of the quality of every single trade (again recall versus precision). As a result, in this study, we will primarily use the metrics of average return to reflect the quality of the signal instead.

$$Average\ Return = \frac{Total\ Return}{\sum_{i=1}^N S_i} \quad (6)$$

3 RESULT AND DISCUSSION

3.1 General Discussion

As discussed in the previous section, the goal of our study is to not only to find out the potential application of machine learning in generate trading signals but also to find out whether or not the representation of candlestick and other popular technical indicators are necessary for the improvement of the models. Therefore, we applied several subsets of features are compared to each other to compared their usefulness.

The first dimension for the set of features is about the representation of the candlestick. In section 2.1, we have mentioned that the candlestick has 3 main characteristics: the candle body, the upper shadow, and the lower shadow. In combination with the close price, we use them to create the "candle" features dataset to compare with the traditional "ohlc" dataset, which contains open, high, low, close price, to see which approach serve better as the input for the machine learning models.

The second dimension is about the inclusion of additional technical indicators or not. The feature dataset with the inclusion of additional technical indicators is called the "extended" feature dataset. For the original set of features (only 4 candlestick features, trades, and volume and the cyclical), we called it the "raw" feature dataset.

Table 7. Performance of All Long-Only Models

Model	Candlestick	Feature	Strategy	Validation			Test		
				Precision	Recall	Avg return	Precision	Recall	Avg return
GRU	candle	extended	long	61.74%	10.44%	0.17%	58.39%	10.87%	0.04%
GRU	candle	raw	long	58.86%	11.27%	0.13%	58.34%	11.57%	0.03%
GRU	ohlc	extended	long	58.08%	11.63%	0.08%	57.11%	10.76%	0.07%
CNN	ohlc	extended	long	57.82%	10.31%	0.08%	57.02%	11.85%	-0.02%
LSTM	ohlc	raw	long	58.87%	10.96%	0.10%	56.72%	12.40%	-0.02%
MLP	candle	extended	long	59.82%	10.47%	0.11%	56.03%	11.82%	0.01%
MLP	candle	raw	long	60.69%	10.20%	0.12%	55.90%	10.85%	0.04%
Logistic	candle	extended	long	59.00%	9.73%	0.10%	55.83%	9.30%	0.01%
CNN	candle	extended	long	58.74%	9.60%	0.09%	55.81%	9.56%	-0.03%
CNN	candle	raw	long	57.93%	11.89%	0.12%	55.71%	12.08%	0.01%
MLP	ohlc	raw	long	57.56%	11.05%	0.07%	55.68%	11.57%	-0.02%
LSTM	candle	raw	long	60.20%	10.87%	0.15%	55.64%	10.39%	-0.02%
LGBM	candle	raw	long	56.51%	11.72%	0.06%	55.62%	11.92%	-0.01%
CNN	ohlc	raw	long	56.93%	9.44%	0.07%	55.60%	11.59%	0.00%
Logistic	ohlc	extended	long	57.94%	9.93%	0.07%	55.53%	10.11%	-0.02%
Logistic	candle	raw	long	59.85%	9.08%	0.14%	55.19%	8.98%	-0.02%
GRU	ohlc	raw	long	58.06%	10.13%	0.08%	55.19%	12.68%	-0.02%
LGBM	candle	extended	long	60.03%	9.28%	0.14%	55.12%	9.09%	0.00%
LGBM	ohlc	extended	long	59.74%	11.49%	0.12%	54.89%	10.64%	0.01%
LGBM	ohlc	raw	long	61.72%	9.75%	0.13%	54.67%	9.49%	-0.01%
XGBoost	candle	raw	long	57.14%	10.18%	0.08%	54.59%	10.04%	-0.01%
MLP	ohlc	extended	long	59.04%	11.00%	0.08%	54.38%	11.06%	-0.01%
LSTM	candle	extended	long	60.88%	9.24%	0.13%	54.29%	8.79%	-0.04%
LSTM	ohlc	extended	long	58.80%	10.73%	0.07%	54.21%	11.78%	-0.02%
XGBoost	ohlc	raw	long	54.91%	9.48%	0.08%	54.10%	10.39%	0.03%
Logistic	ohlc	raw	long	59.04%	9.04%	0.08%	54.09%	9.19%	-0.02%
XGBoost	ohlc	extended	long	56.97%	9.48%	0.05%	52.64%	9.23%	-0.01%
XGBoost	candle	extended	long	57.48%	9.26%	0.08%	52.20%	8.24%	0.01%

Table 8. Performance of All Short-Only Models

Model	Candlestick	Feature	Strategy	Validation			Test		
				Precision	Recall	Avg return	Precision	Recall	Avg return
LGBM	ohlc	raw	short	54.00%	11.23%	-0.02%	60.05%	11.97%	0.07%
MLP	candle	extended	short	57.93%	9.19%	0.04%	59.16%	9.31%	0.04%
LGBM	candle	extended	short	55.59%	9.33%	0.04%	59.02%	9.36%	0.05%
LGBM	candle	raw	short	54.01%	11.35%	-0.02%	58.77%	11.35%	0.03%
GRU	ohlc	raw	short	55.30%	11.40%	-0.01%	58.31%	10.59%	0.05%
Logistic	ohlc	extended	short	54.91%	10.35%	0.02%	58.02%	10.75%	0.03%
LSTM	candle	raw	short	59.55%	11.40%	0.01%	57.84%	9.54%	0.01%
MLP	ohlc	extended	short	55.24%	10.14%	0.01%	57.77%	12.08%	0.00%
LSTM	ohlc	raw	short	57.54%	11.32%	0.00%	57.66%	10.16%	0.01%
LGBM	ohlc	extended	short	53.98%	11.92%	0.01%	57.66%	11.28%	0.01%
Logistic	candle	extended	short	56.32%	9.09%	0.01%	57.63%	8.90%	0.02%
LSTM	ohlc	extended	short	56.43%	10.94%	0.00%	57.45%	10.94%	-0.01%
CNN	ohlc	raw	short	55.86%	10.75%	0.02%	57.29%	11.32%	0.03%
GRU	candle	raw	short	57.91%	11.04%	-0.02%	57.28%	8.19%	-0.01%
Logistic	ohlc	raw	short	53.76%	10.85%	-0.03%	57.18%	10.75%	0.02%
CNN	candle	extended	short	56.20%	9.90%	0.03%	57.10%	9.20%	0.03%
LSTM	candle	extended	short	55.01%	11.87%	-0.01%	56.52%	10.91%	-0.01%
GRU	ohlc	extended	short	52.02%	45.35%	-0.02%	56.15%	46.05%	0.02%
MLP	candle	raw	short	54.44%	10.04%	-0.04%	56.02%	9.15%	0.00%
XGBoost	candle	raw	short	52.96%	10.19%	0.00%	55.92%	10.48%	0.05%
CNN	ohlc	extended	short	52.17%	44.80%	-0.03%	55.89%	44.75%	0.02%
GRU	candle	extended	short	51.91%	46.11%	-0.02%	55.83%	47.91%	0.02%
XGBoost	candle	extended	short	52.22%	10.35%	-0.03%	55.78%	10.27%	0.02%
CNN	candle	raw	short	55.29%	9.92%	-0.02%	55.42%	9.36%	-0.01%
MLP	ohlc	raw	short	56.72%	10.83%	0.00%	54.55%	10.43%	-0.01%
Logistic	candle	raw	short	51.95%	47.20%	0.00%	54.44%	47.24%	0.02%
XGBoost	ohlc	extended	short	53.08%	11.44%	-0.02%	54.04%	10.87%	0.01%
XGBoost	ohlc	raw	short	52.31%	9.43%	-0.04%	53.02%	9.45%	0.00%

In combination with the primary two dimensions of long and short strategy, in the end, each type of machine learning models are fitted 8 different times as the combination of the 3 dimensions. In the end, we have in total 56 different models and all of their result are shown in Table 7 and Table 8.

The two tables show the list of all models with their performance metrics calculated on the validation and the test set. The order of how they are listed is based on their precision on the test set, from the highest to the lowest. From just a quick view of the two tables, we can see that most of the top models are deep learning models with the exception of the LightGBM in shorting strategy. Additionally, all of the machine learning models overall outperform almost all of the candlestick patterns in both precision and average return, given the same range of recall. Given the fact that the market is quite bearish in the test set and the shorting is generally more difficult in practice, we believe the performance for the long-only strategy is more reliable compared to the short-only strategy

The best performing model here in both validation and test set is the GRU long-only model which is trained on the extended candle dataset with 58.36% precision and the average return of 0.04%. The performance of this strategy is shown in figure 2 and 3 where it is compared with the buy-and-hold strategy of Bitcoin for reference.

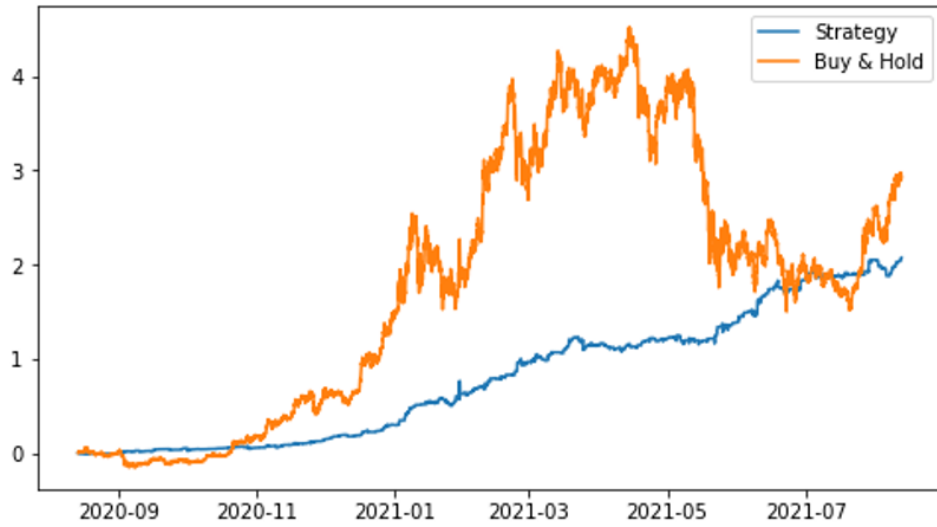


Figure 2. Model Performance in Validation set

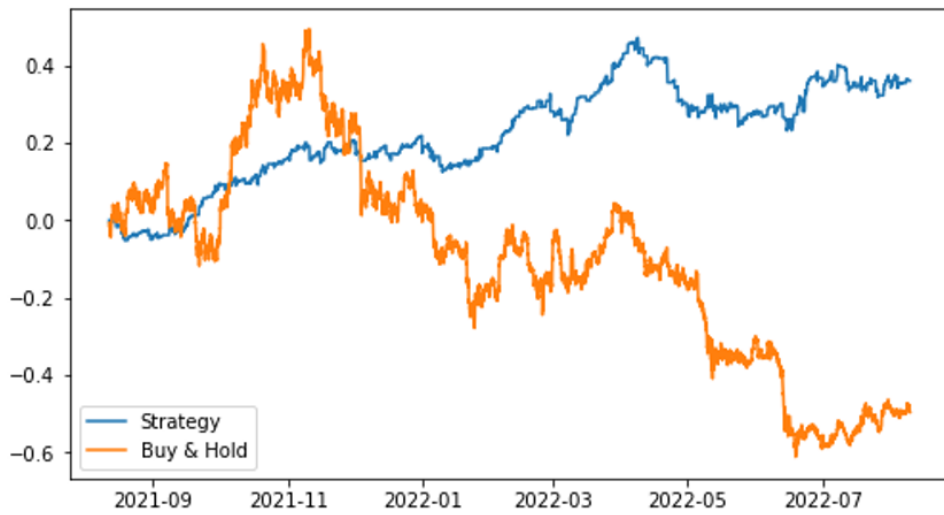


Figure 3. Model Performance in Test set

From the two figures, we can see the model outperformed the benchmark and there are way less downtrends and drawdowns during the both period. The cumulative return of the strategy also increase steadily and does not come from only a few single trades or some specific period.

3.2 Best Candlestick Features

Table 9. Comparison between 2 candlestick representation methods

Data	Validation			Test		
	Precision	Recall	Avg return	Precision	Recall	Avg return
candle	57.15%	12.86%	0.056%	56.26%	12.67%	0.010%
ohlc	56.39%	13.04%	0.038%	55.96%	13.36%	0.006%

Table 9 presents the average performance metrics of all machine learning models using either the "ohlc" features or the "candle" features. From the table, we can see the reason why the candle representation is more used in technical analysis. The performance of the candle data is slightly better than the performance of the OHLC data (0.8% for the validation set and only 0.3% for the test set).

3.3 Inclusion of Technical Indicators

The difference caused by the addition of technical indicators is even less noticeable compared to the previous case. Table 10 illustrates the performance of the extended features in comparison with the raw features.

Table 10. Comparison between the raw dataset and extended dataset

Feature	Validation			Test		
	Precision	Recall	Avg return	Precision	Recall	Avg return
extended	56.75%	14.05%	0.049%	56.12%	14.13%	0.009%
raw	56.78%	11.86%	0.045%	56.10%	11.90%	0.008%

The addition of 12 new technical indicators almost does not substantially increase the performance of all machine learning models, at least in the quality of the signals. The inclusion of those indicators only led to an increase of measly 0.02% in precision for the test set. However, those indicators surely do increase the numbers of signals with more than 2% improvement in recall in both dataset.

3.4 Models Comparison

Finally, we would like to compare the general performance of every machine learning model that we used. Table 11 shows the average performance metrics of all models.

Table 11. Average Performance of all models

Model	Validation			Test		
	Precision	Recall	Avg return	Precision	Recall	Avg return
GRU	56.73%	19.67%	0.048%	57.07%	19.83%	0.025%
LGBM	56.95%	10.76%	0.056%	56.97%	10.64%	0.018%
LSTM	58.41%	10.92%	0.059%	56.29%	10.61%	-0.011%
CNN	56.37%	14.58%	0.045%	56.23%	14.96%	0.003%
MLP	57.68%	10.36%	0.048%	56.19%	10.78%	0.008%
Logistic	56.60%	14.41%	0.048%	55.99%	14.40%	0.005%
XGBoost	54.63%	9.98%	0.026%	54.03%	9.87%	0.011%

From the table, it is easy to observe that the recurrent neural networks (GRU and LSTM) are better in both precision and average return for this specific classification task with the exception of the LightGBM. They are ranked from the highest to the lowest in the precision calculated on the test set. Among them, the GRU model is better in almost every metrics compared to other models. For the LightGBM models, they come second but only perform well for the short-only strategy in the bearish market. Their long-only versions are quite worse compared to the other. As a result, we do not value their performance that high. The LSTM model comes third but with the best performance

in validation set and negative return in test set. It is likely that the LSTM models are overfitted in the training phase. The logistic model, due to its simplicity, performs worse than most models but still surprisingly better than the XGBoost. The XGBoost, despite its similar nature with the LightGBM, actually performed way worse than all other models, with 2% precision lower (a noticeable margin) than the 2nd worst one.

4 CONCLUSION

Our analysis has pointed out that it is not inconceivable that the Bitcoin price history might contain some predictive information. That information can even be enhanced through the use of several popular technical indicators so that they can be exploited by quantitative methods to create a profitable trading strategy. Our detailed results show that our prediction models using the machine learning approach trained on the Binance cryptocurrency exchange can create the profits that are better than the classic Buy-and-Hold strategy and provide much better precision than random guesses.

Our study has shown how several popular machine learning methods such as neural networks, decision trees, and logistic regression performed on Bitcoin price trend prediction. The results turned out that the neural networks did the best compared to other models, especially in the case of fewer features. However, among several neural network architectures, the state-of-the-art CNNs did not outperform other architectures with comparable complexity as we expected [5]. The recurrent neural network, on the other hand, shows superior performance with the GRU architectures and, to a lesser extent, the LSTM architectures. Between the two, the GRU performed better than LSTM. However, due to the limited sample size, we can not arbitrarily conclude that the GRU is better than the LSTM.

REFERENCES

- ANGHEL, D.-G. A reality check on trading rule performance in the cryptocurrency market: Machine learning vs. technical analysis. *Finance Research Letters* 39 (2021), 101655.
- BASAK, S., KAR, S., SAHA, S., KHAIDEM, L., AND DEY, S. R. Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance* 47 (2019), 552–567.
- BIANCHI, D., BÜCHNER, M., AND TAMONI, A. Bond risk premiums with machine learning. *The Review of Financial Studies* 34, 2 (2021), 1046–1089.
- COSENZA, D. N., KORHONEN, L., MALTAMO, M., PACKALEN, P., STRUNK, J. L., NÆSSET, E., GOBAKKEN, T., SOARES, P., AND TOMÉ, M. Comparison of linear regression, k-nearest neighbour and random forest methods in airborne laser-scanning-based prediction of growing stock. *Forestry: An International Journal of Forest Research* 94, 2 (2021), 311–323.
- GHOSHAL, S., AND ROBERTS, S. Thresholded convnet ensembles: neural networks for technical forecasting. *Neural Computing and Applications* (04 2020).
- GU, S., KELLY, B., AND XIU, D. Empirical asset pricing via machine learning. *The Review of Financial Studies* 33, 5 (2020), 2223–2273.
- HANCOCK, J., AND KHOSHGOFTAAR, T. M. Performance of catboost and xgboost in medicare fraud detection. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)* (2020), IEEE, pp. 572–579.
- NOBRE, J., AND NEVES, R. F. Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets. *Expert Systems with Applications* 125 (2019), 181–194.
- RODRIGUEZ, P. N., AND RODRIGUEZ, A. Predicting stock market indices movements. *WIT Transactions on Modelling and Simulation* 38 (2004).
- WANG, Z., AND OATES, T. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks.
- YUN, K. K., YOON, S. W., AND WON, D. Prediction of stock price direction using a hybrid ga-xgboost algorithm with a three-stage feature engineering process. *Expert Systems with Applications* 186 (2021), 115716.
- ZHANG, Y., CHU, G., AND SHEN, D. The role of investor attention in predicting stock prices: The long short-term memory networks perspective. *Finance Research Letters* 38 (2021), 101484.
- ZOLOTAREVA, E. Aiding long-term investment decisions with xgboost machine learning model. In *International Conference on Artificial Intelligence and Soft Computing* (2021), Springer, pp. 414–427.