

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/314933260>

Application of Machine Learning Algorithms for Bitcoin Automated Trading

Chapter · March 2016

DOI: 10.1007/978-3-319-30315-4_14

CITATIONS

17

READS

3,239

1 author:



Kamil Żbikowski

Warsaw University of Technology

11 PUBLICATIONS 161 CITATIONS

SEE PROFILE

Application of Machine Learning Algorithms for Bitcoin Automated Trading

Kamil Żbikowski

Institute of Computer Science
Faculty of Electronics and Information Technology
Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
`kamil.zbikowski@ii.pw.edu.pl`

Abstract. The aim of this paper is to compare and analyze different approaches to the problem of automated trading on the Bitcoin market. We compare simple technical analysis method with more complex machine learning models. Experimental results showed that the performance of tested algorithms is promising and that Bitcoin market is still in its youth, and further market opportunities can be found. To the best of our knowledge, this is the first work that tries to investigate applying machine learning methods for the purpose of creating trading strategies on the Bitcoin market.

1 Introduction

Bitcoin is a peer-to-peer, decentralized electronic cash protocol. It is based on the proof-of-work algorithm that uses cryptographic hash function for confirmation of subsequent blocks of transactions. Block's approval is realised by miners. Skipping the details, the general idea behind is to find such a 32-bit field (called *nonce*) that added to the block will result in a value of a hash function with n leading zeros. Miners compete between each other to earn the incentive for finding such a value. In order to achieve this goal, they have to employ the computational power bearing the costs of hardware and electricity. The more miners are engaged in the described process, the harder it is to find the required value of hash function as more leading zeros are required in order to accept the block. The measure describing this is called *difficulty*, and it is self-regulated by the protocol.

Bitcoin was proposed by Satoshi Nakamoto in 2008 ([1]). Since then the whole Bitcoin ecosystem has been growing rapidly. Since then it also raises questions about some fundamental issues it promises. One of them is the anxiety about the privacy and anonymity. More on this can be found in [2] and [3]. For a long time, Bitcoin's proof-of-work has been perceived as a safe to potential attacks. Recently, there have been some concerns about that as well [4].

Despite some potential threats, Bitcoin community is growing and attracts more and more attention both from the academia and business. In recent years,

plenty of Bitcoin exchanges have been launched. The main advantage of them is a simplification of trading and valuation of Bitcoin. As there is a market there also exist investors, speculators and hedgers ready to exchange bitcoins and fiat currencies. As it is done on traditional financial markets, one may be interested in applying automated trading strategies that would help gain reasonable return on invested funds. In this paper, we would like to test and compare several approaches from traditional markets applied to the task of automated trading on Bitcoin.

The remainder of this paper is organised as follows. Section 2 briefly describes the concept of Support Vector Machines and Volume Weighted Support Vector Machines. We also provide information about indicators used for the purpose of presented experiments. In section 3 we provide information about data that was used for computations. In 4 we analyze experiments' results. Finally, in section 5 we make some concluding remarks.

2 Prediction Models

In order to investigate the existence of profitable automated trading algorithms, we propose to use some of the most well-known approaches. First, we would like to verify basic exponential moving average (EMA) with some simple enhancements. Then, we would like to apply a much more sophisticated method that directly employs machine learning methods like support vector machines (SVM) in combination with box theory. Last but not least, we would like to determine whether some improvements in the machine learning algorithm, validated against data from other financial markets, will also perform well.

2.1 Exponential Moving Average

First model that we use is exponential moving average (EMA) that is widely known technical analysis indicator. It assigns to the more recent quotations, higher weights basing on the assumption that they are more relevant than the older ones. EMA for the moment t is defined recurrently as follows:

$$EMA_t^n = (P_t - EMA_{t-1})\alpha + EMA_{t-1} \quad (1)$$

where P_t is the price for the moment t , α is defined as $2/(n+1)$ and the value of EMA_0 is set to P_0 .

Signals to buy or sell are triggered when the $EMA_t < P_t$ or $EMA_t > P_t$ respectively. Some traders believe that there exist predefined values of n that suit for every case. We took a rather different approach. We applied walk-forward procedure ([5]) in order to re-optimize the model every 100 quotations. During this step, we conduct an extensive search using grid search procedure looking for the value of n for the next moving window basing on last 200 quotations.

Table 1: Indicators used as the inputs of Box SVM and VW-SVM models (n denotes the data length used for computations; t is the index of the element in the time series).

Indicator	Formula	Box SVM	VW-SVM
Close price	P_t	+	+
Volume	V_t	+	+
Average Price	$\mu_t = \frac{\sum_{k=0}^t P_{t-k}}{t}$	-	+
Standard Deviation	$\sigma_t = \sqrt{\frac{\sum_{k=0}^t (P_{t-k} - \mu_t)^2}{t}}$	+	+
Relative Strength Index	$RSI_t^n = 100 - \frac{100}{1 + RS_t^n}$ $RS_t^n = \frac{\sum_{k=0}^n U_k}{\sum_{k=0}^n D_k}$ $U_t = \begin{cases} P_t - P_{t-1} & \text{if } P_t \geq P_{t-1} \\ 0 & \text{otherwise} \end{cases}$ $D_t = \begin{cases} P_{t-1} - P_t & \text{if } P_t < P_{t-1} \\ 0 & \text{otherwise} \end{cases}$	+	+
Daily rate of return	$R_t = \frac{P_t}{P_{t-1}}$	-	+
%R Oscillator	$\%R_t^n = 100 \frac{MaxP_t^n - P_t}{MaxP_t^n - MinP_t^n}$ $MaxP_t^n = \max\{P_t, P_{t-1}, \dots, P_{t-n}\}$ $MinP_t^n = \min\{P_t, P_{t-1}, \dots, P_{t-n}\}$	-	+
Average True Range	$ATR_t^n = \frac{1}{n} \sum_{i=0}^n TR_{t-i}$ $TR_t = \max\{P_t^H - P_t^L, P_t^H - P_{t-1}^C, P_t^L - P_{t-1}^C\}$	-	+
Vortex	$V_t^{n(+)} = \frac{\sum_{k=0}^d (P_{t-k}^H - P_{t-k}^L)}{\sum_{k=0}^d TR_{t-k}}$ $V_t^{n(-)} = \frac{\sum_{k=0}^d (P_{t-k}^L - P_{t-k}^H)}{\sum_{k=0}^d TR_{t-k}}$	-	+
On Balance Volume	$OBV_t = \begin{cases} OBV_{t-1} + V_t & \text{if } P_t > P_{t-1} \\ OBV_{t-1} - V_t & \text{if } P_t < P_{t-1} \\ OBV_{t-1} & \text{if } P_t = P_{t-1} \end{cases}$	-	+

2.2 Support Vector Machines with Box Theory

Support Vector Machine (SVM) is a machine learning algorithm that employs the risk minimization principle that was described in [6]. Its general idea is based on maximizing the margin of separating hyperplanes. This operation prevents from building models that are overfitted and directly lowers the error obtained

on the test sample. The general penalty function is defined as follows:

$$\min_{\mathbf{w} \in \mathcal{H}, b \in \mathcal{R}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i, \quad (2)$$

subject to:

$$y_i(\mathbf{w}\mathbf{x}_i - b) \geq 1 - \xi_i, \quad (3)$$

$$\xi_i \geq 0, \quad (4)$$

where C is a penalty term, ξ_i are slack variables, x_i is the i th input value and y_i is the corresponding target value.

The model based on box theory and SVM was proposed in [7]. The box theory assumes that in order to obtain valid buy or sell signals the price should break through the oscillation box constructed over the specified period of time. The input of the model consists of several technical indicators and statistical measures gathered in Table 1.

2.3 Volume Weighted Support Vector Machines

The idea of Volume Weighted SVM was introduced in [5] and further investigated in [8]. It is based on the simple, yet powerful, assumption that latest quotations should have a greater impact while building a prediction model. To achieve this goal, there was proposed a slight modification in a basic SVM formula 2 that takes following form:

$$\min_{\mathbf{w} \in \mathcal{H}, b \in \mathcal{R}} \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m C_i \xi_i, \quad (5)$$

In 5, each C_i term determines the weight assigned to particular input x_i and takes the form:

$$C_i = v_i C, \quad (6)$$

where

$$v_t = \frac{\sum_{k=0}^d V_{t-k}}{\sum_{i=0}^m W_{t-i}}, \quad (7)$$

where V_{t-k} is the real transactional volume for the moment t , k depicts the delay period and d is the data length required for feature calculation.

In [8], the formula 5 was used in combination with several technical indicators. The list of indicators used as an input for VW-SVM for the purpose of the experiment presented in this paper is enlisted in Table 1.

3 Research Data

For the purpose of presented experiment, we used data gather from one of the largest Bitcoin exchanges in terms of average daily volume traded - Bitstamp (www.bitstamp.com). It is available via their API. We have merged all the quotations to the form of 15-minutes aggregates and as a price P_t we have applied

the closing price of an aggregate from the moment t . The spectrum of selected data consists of 2221 aggregates. It starts on January 9th 2015 and ends on February 2th 2015.

Transactions' costs vary depending on exchange and traded volume over time. We assumed transaction fee at the level of 0.2% which seemed to be a reasonable choice.

4 Empirical Study

We applied a walk forward testing[5] procedure that is a recommended method for constructing machine learning models dedicated to time-series data. Each algorithm was able to open both short and long positions. That is not directly possible at the moment having an account open on the Bitstamp exchange. However, several competitors provide such a service for their customers. Therefore, we got to the conclusion that this decision was justified.

Table 2: Backtesting results for EMA, Box SVM and VW-SVM models. R is the rate of return over the whole testing period, DD is the maximum drawdown experienced by the algorithm.

	R[%]	DD[%]
EMA	-59.11	64.30
Box SVM	10.58	25.94
VW-SVM	33.52	22.88
Buy & Hold	4.86 (28.10)	30.90

We run each of the models described in section 2 separately using data that was collected according to section 3. The summary of these experiments is collected in Table 2. We assume that initially, each of the algorithms holds a portfolio of 100,000\$.

In order to measure the degree of risk we provide a drawdown measure that is computed as a maximum loss from the highest value of the equity that algorithm was holding (expressed in percent). As an assessment of model's performance, we provide rate of return computed over the whole testing period. All results are compared to the so-called buy and hold strategy that assumes opening a long position at the beginning of available data and holding up to the end of data. Each time the position is open by the algorithm it engages the whole capital that it is holding at the particular moment. It is rather a weak position size management. However, it is justified if we compare trading result to buy and hold reference strategy.

Simple exponential moving average model lost 59.11%. As it can be seen in the Figure 1 the equity line is declining throughout the whole tested period

of time. The drawdown of this model reached 64.30%. This strategy obtained the worst result in terms of both maximum drawdown and the rate of return. The buy and hold strategy performed significantly better. From this outcome we may conclude that even for such an immature market like it is in case of Bitcoin, simple trading strategies based on technical analysis are not profitable.

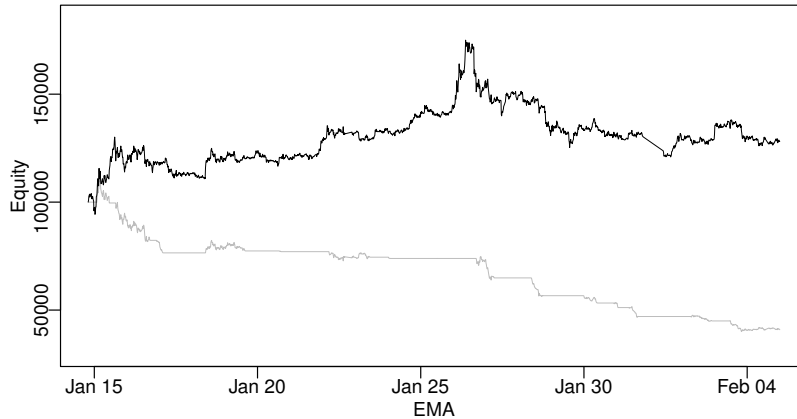


Fig. 1: EMA results. Black – buy & hold strategy, gray – EMA performance.

First of the advanced strategies that we were testing, namely Box SVM, achieved much better results compared to EMA. Its rate of return was 10.58% which is 5.72% improvement over buy and hold (see figure 2). In terms of the maximum drawdown, it did not cross the level of 25.94% which still denotes a high risk of investment. However, compared to buy and hold strategy, it achieved 4.96% improvement.

The second strategy that is based on SVM classifier, namely Volume-Weighted SVM, managed to achieve the rate of return of 33.58%. Its equity curve is shown in the figure 3. It is the highest return ratio among analyzed models. It is 28.66% improvement over buy and hold strategy and 23% over Box SVM. Moreover, VW-SVM exposure to the risk measured by the maximum drawdown was the lowest. It was better than buy and hold by 8.02% and by 3.14% than Box SVM model.

5 Conclusion

In this paper, we compare three different algorithms that are applied to the task of trading on Bitcoin exchange. One of them was a simple technical indicator based on the exponential moving average. Two others, namely Box SVM algorithm and Volume-Weighted SVM, were selected from the literature. Both em-

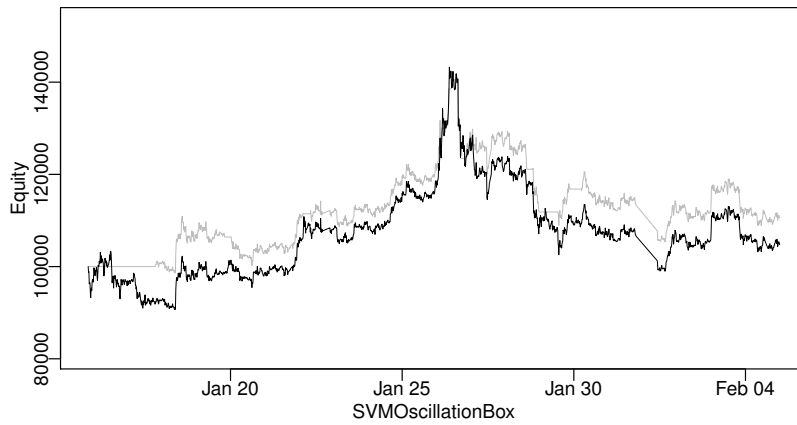


Fig. 2: Box SVM results. Black – buy & hold strategy, gray – Box SVM performance.

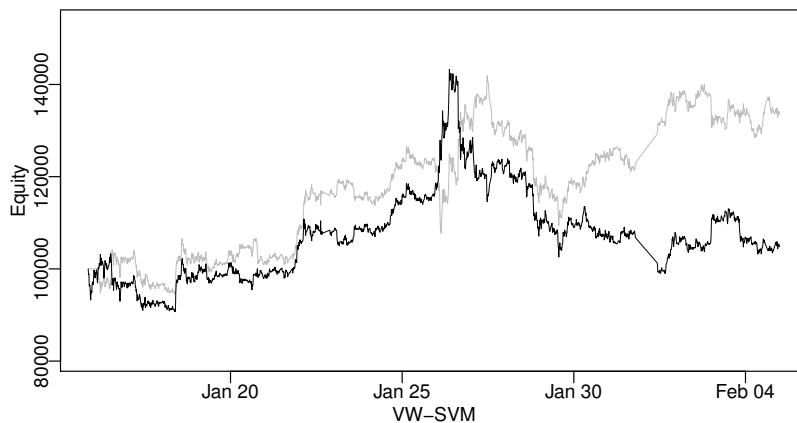


Fig. 3: VW-SVM results.0 Black – buy & hold strategy, gray – VW-SVM performance.

ploy SVM classifier and proved to be effective on other types of capital markets. As it was shown in the empirical analysis, applying machine learning modelling enhances the overall trading performance.

The obtained results are very promising. Assuming achieved scores, the compound return over a one-year period would exceed reasonable levels known from financial markets. It is a direct evidence that Bitcoin market is still young and tends to be ineffective.

Future work will address the issue of position sizing and minimizing the overall risk expressed in this paper by the mean of the maximum drawdown. It

is also worth investigating other methods known from traditional markets. One of them can be acquiring data from social networks and use it as an input for machine learning algorithms in order to predict substantial price movements.

References

1. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Consulted **1**(2012) (2008) 28
2. Androulaki, E., Karame, G., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In Sadeghi, A.R., ed.: *Financial Cryptography and Data Security*. Volume 7859 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2013) 34–51
3. Koshy, P., Koshy, D., McDaniel, P.: An analysis of anonymity in bitcoin using p2p network traffic. In Christin, N., Safavi-Naini, R., eds.: *Financial Cryptography and Data Security*. *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2014) 469–485
4. Eyal, I., Sirer, E.: Majority is not enough: Bitcoin mining is vulnerable. In Christin, N., Safavi-Naini, R., eds.: *Financial Cryptography and Data Security*. *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2014) 436–454
5. Żbikowski, K.: Time series forecasting with volume weighted support vector machines. In Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., Kostrzewa, D., eds.: *Beyond Databases, Architectures, and Structures*. Volume 424 of *Communications in Computer and Information Science*. Springer International Publishing (2014) 250–258
6. Cortes, C., Vapnik, V.: Support-vector networks. *Machine learning* **20**(3) (1995) 273–297
7. Wen, Q., Yang, Z., Song, Y., Jia, P.: Automatic stock decision support system based on box theory and svm algorithm. *Expert Systems with Applications* **37**(2) (2010) 1015–1022
8. Żbikowski, K.: Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. *Expert Systems with Applications* **42**(4) (2015) 1797–1805