



Navigating Bitcoin Panic-Selling using Linear Approach

Agi Prasetiadi*

Faculty of Informatics, Institut Teknologi Telkom Purwokerto
 128 D.I. Panjaitan Street, Purwokerto, Central Java, Indonesia

*Corresponding Email: agi@ittelkom-pwt.ac.id

Received 14 September 2020, Revised 08 October 2020, Accepted 28 November 2020

Abstract — COVID-19 affects significant human activity around the globe, including Bitcoin prices. The Bitcoin price is well known for its volatility, so it is not a big shocker when the panic-selling occurs during the pandemic. However, the mechanism to cope with these breakouts, especially the bearish one, is contentious. The experts give numerous pieces of advice with different conclusions in the end. It is also the same with Machine Learning. Various kernels show different results regarding how the price will move. It depends on the window size, how the data is being preprocessed, and the algorithm used. This paper inspects the best combination that various machine learning can offer with a linear approach to navigate the price prediction based on its depth interval, window size until the algorithms themselves. This paper also proposed a new approach to seeing the prediction range called s-steps ahead prediction using a linear model. The result shows that simple machine learning can herd 99.715% profit even during the bearish breakout.

Keywords – Bitcoin, Panic-Selling, Machine Learning, Linear Model

Copyright © 2020 JURNAL INFOTEL

All rights reserved.

I. INTRODUCTION

Since the Bitcoin market is live 24/7, it gives real-time dynamics that the stock market cannot offer [1]. Moreover, it is easier to access, resulting in high volatility in its movement [2]. For a short time interval, it is quite hard to oversee where the market is heading, but in the long term, it gets even more confusing [3]-[5]. It is understandable because Bitcoin's very nature is not like the stock market, where it has a real ground commodity to offer. Bitcoin offers a free-bank ecosystem in which everyone can transact their money in decentralized ways [6].

People often talk about how the holder behaves when the stress test occurs [7]. During breakouts, the various big players may test the market by pushing it to its resistance level or support level. When it breaks, the market could go in different directions, mostly in a violent way. But there is a catch; in every extreme turn up or turn down, there will always be a dead cat bounce effect.

Based on Fig.1, it can be seen that two dead cat bounce effects happened at the transaction that occurs between March 15th, 2019, to August 29th, 2019. Two dead cat bounce effects happened [8]. There are significant drops that occur during these times. People

who hold great expectations since starting of the bullish trend might get scared seeing a sudden drop on the second wave of the dead cat bounce effect since it might never return to its highest position again. It also allows people who don't buy it at a lower price to sell it again at a higher price in the last period of the dead cat bounce effect, as can be seen in Fig.1.

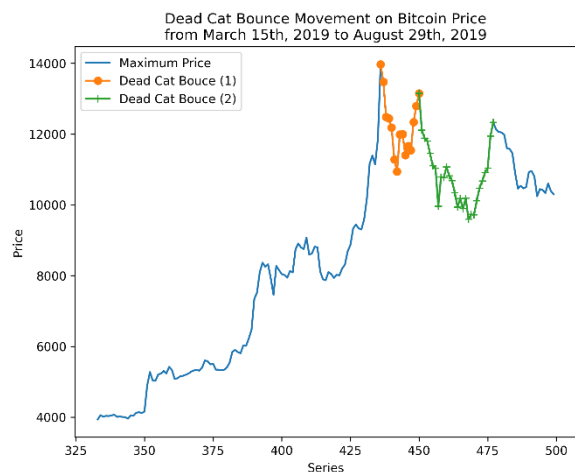


Fig.1. Dead Cat Bounce Effect

COVID-19 certainly causes massive breakout unintentionally, resulting in panic-selling [9]-[11]. This paper investigates various Machine Learning techniques with a linear approach to navigate this unexpected movement [12].

Linearity can be understood as to how the function forms a vector space. Consider f as a function in field F , where it has X, Y as its vector spaces. If $f(a_0x_0 + a_1x_1) = a_0f(x_0) + a_1f(x_1)$ where $a_0, a_1 \in F$ and $x_0, x_1 \in X$, then $f: X \rightarrow Y$ is considered linear [13].

This paper assumes that every time series elements contribute to how people behave and think decide to buy or sell the coin. Based on this same approach, people used Technical Analysis to plan their portfolios. The only problem in Technical Analysis is prone to human subjectivity. If there are patterns in every decision is made on the market, it is not strange that there must be some correlation between the particular position of the price and the next movement of the price.

Research by [14] achieved a 5.36% mean absolute percentage error (MAPE) in predicting Bitcoin price using an autoregressive integrated moving average (ARIMA). On the other hand, research by [15] classified Bitcoin trends using ARIMA, Long Short-Term Memory (LSTM), recurrent neural network (RNN) with accuracies 52.78%, 50.25%, 50.05%, respectively. Research by [16] also found that Twitter significantly affects the next day's trading volume. This paper proposed a new approach to seeing the prediction range we called s -steps ahead prediction using a linear model. This paper also proposed a new approach for preprocessing the raw data based on Fast Fourier Transform and Particle Swarm Optimization behavior. We also conduct window size inspection to get a better understanding of how to analyze the Bitcoin market.

II. RESEARCH METHOD

This paper uses several phases to build the kernel. The first phase is the randomness and slope test to get a basic idea of how the market flow for the past four years. The second phase is a short periodic test to grasp what kind of model is suitable for the predicting. The last stage is deploying full force prediction using hints showed by the second phase called backtracking.

A. Random Test

Four random distributions are used to check the predictability of the market; Uniform, Exponential, Logistic, and Poisson distribution [17]-[20]. Uniform distribution follows the probability density function as $p(x) = (b - a)^{-1}$ within $[a, b]$ and 0 elsewhere. Exponential distribution follows the probability density function as $p(x, \lambda) = \lambda e^{-\lambda x}$. Logistic distribution follows the probability density function as,

$$p(x) = \frac{e^{-\frac{x-\mu}{s}}}{s \left(1 + e^{-\frac{x-\mu}{s}}\right)^2} \quad (1)$$

Last, Poisson distribution follows the probability density function as $p(k, \lambda) = \lambda^k e^{-\lambda} / k!$.

If the market is well predicted using a particular random prediction, then the market trend is followed by these distributions. If all the distributions fail to predict the market, then there is a chance that the market is not random, after all.

B. Slope Test

The slope can be defined as how big the change occurs. In mathematical terms, it can be written as $m = (x_{n+k} - x_n) / k$. We might consider slope at a specific point x of a function $f(x)$ as $f'(x)$. However, it's not feasible to take the derivative or the slope of a market for a time range of nearly 0 seconds, $\lim k \rightarrow 0$. The market's movement for little time depth is adorned by flakes and noises fluctuating temporary price in a short amount of time. Thus, it is more probable to take slope by setting the k in the higher term while minimizing the noise effect in the market [21].

C. Linear Models

Let's assume the movement pattern is consistent all over time. If \bar{x}_i is the series of specific time ranges, then there must be a vector \bar{w} that satisfies $y_i = w_0 + \sum_{j=0}^{|\bar{x}_i|-1} w_{j+1} f(x_{i,j}) + \epsilon_i$, where y_i is the predicted price, f is the function model, and ϵ is the error. The objective is to find \bar{w} , where it holds the minimum ϵ value. Since \bar{w} will be constant, then \bar{w} can also be considered the pattern of the market-pair. In this paper, the model that can satisfy this equation also will be called the kernel [22] [23].

Five kernels will be used to check the regularity of the movements,

1. Multi-task Lasso (MTL),
2. Lasso LARS (LLC),
3. Huber Regressor (HR),
4. Multi-task ElasticNet (MTLENC), and
5. Ridge Regression (Ridge).

These kernels will be tested its accuracy on 4 hours interval market movement. The five kernels' performance will be evaluated progressively on this market data. If the kernel didn't perform well during the test, it would be discharged for further schemes. Then, the kernel with plausible prediction will be chosen to do full force automatic daily trading simulation. In this paper, all kernels used the cross-validation technique, except Huber and Ridge Regression.

- a) *Multi-task Lasso*: In Multi-task Lasso, the objective function to minimize can be expressed as follows.

$$L(\mathbf{W}, \lambda) = \|\mathbf{XW} - \mathbf{Y}\|_{\text{Fro}}^2 + \lambda \|\mathbf{W}\|_{21}. \quad (2)$$

Let say \mathbf{M} is a matrix; then the Frobenius norm can be calculated as $\|\mathbf{M}\|_{\text{Fro}} = (\sum_{i,j} m_{i,j}^2)^{1/2}$,

and $\ell_1\ell_2$ -norm can be calculated as $\|\mathbf{M}\|_{21} = \sum_i (\sum_j m_{i,j}^2)^{1/2}$ [24].

- b) *Lasso LARS*: In Lasso regression, the cost function is modified with ℓ_1 regularization penalty, where

$$L(\mathbf{W}, \lambda) = \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_2^2 + \lambda_1 \|\mathbf{W}\|_1. \quad (3)$$

This Regression can overcome multicollinearity [25]. In the case of this kernel, Least Angle Regression and Shrinkage (LARS) algorithm is used to estimate the Lasso parameters effectively [26] [27]. However, since this algorithm is performed based on residuals iterative computation, it might not be robust with the presence of noise [28].

- c) *Huber Regressor*: Huber Regression treats the sample with two different loss function, squared loss ℓ_2 and absolute loss ℓ_1 : $L(y, \hat{y}) = (y - \hat{y})^2$ when the residual is lesser than or equal to h , and $L(y, \hat{y}) = |y - \hat{y}|$ when the residual is higher than h , where h is the hyperparameter. Thus, for loss that falls into the ℓ_2 category, this Regression behaves similarly with the normal distribution. However, since this regression technique behaves similarly with Laplace distribution for the loss higher than h , this very nature makes Huber Regression robust to outliers [29].

Outliers in the market can be compared to how the market behaves during breakout [30]. And the good news about the breakout is that there will be a dead cat bounce effect after the sudden move. How much the change caused by the dead cat bounce effect can be predicted using the Fibonacci Fan technique. The question is that whether after the dead cat bounce effect, the price will correct itself, or it returns to a new trend phase. Thus, Huber Regression makes an excellent candidate to predict the overall market movement.

- d) *Multi-task ElasticNet*: Multi-task ElasticNet trains the data with a mixture of ℓ_1, ℓ_2 -norm with ℓ_2 regularization. It has the ability to estimates sparse coefficients. The following objective is used to perform this kernel.

$$L(\mathbf{W}, \lambda) = \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_{\text{Fro}}^2 + \lambda_{21}\lambda_1 \|\mathbf{W}\|_{21} + \lambda_{21}(1 - \lambda_1) \|\mathbf{W}\|_{\text{Fro}}^2/2. \quad (4)$$

λ_{21} is the constant used to multiply ℓ_1, ℓ_2 term. Since $0 \leq \lambda_1 \leq 1$, if $\lambda_1 = 0$, this kernel will be calculated with ℓ_1, ℓ_2 penalty. As opposite, if $\lambda_1 = 1$, then this kernel will be calculated with ℓ_1 penalty [31].

- e) *Ridge Regression*: Ridge Regression, also is known as Tikhonov regularization, can solve the multicollinearity problem. It is started by

standardization of the data value by subtracting its means and dividing by its standard deviation. Thus, if $\hat{\mathbf{W}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ can approach \mathbf{W} from $\mathbf{Y} = \mathbf{X}\mathbf{W} + \epsilon$ in ordinary least squares, then Ridge Regression approached \mathbf{W} by adding k value to its diagonal $\mathbf{X}'\mathbf{X}$ elements $\hat{\mathbf{W}} = (\mathbf{X}'\mathbf{X} + k\mathbf{I})^{-1}\mathbf{X}'\mathbf{Y}$ [32].

Ridge Regression is said to have stabler performance when there are small changes in the data. Thus, it is one of the excellent candidates in predicting Bitcoin price movement [33].

D. Fast Fourier Transform Approach

The Fast Fourier Transform (FFT) is one technique to perform the Discrete Fourier Transform (DFT). DFT itself converts a series of complex numbers from time domain into another equal size of complex number series in the frequency domain. The DFT can be described as $X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}$. The lower index of k indicates the lower frequency, and the higher index of k indicates the higher frequency [34].

Since the market movement in short depth is dominated by speculator noise, then it is probable that we filter the market price by its frequency. If the intention is for a long-term trend, it might be beneficial to remove the higher frequency altogether. It might be beneficial for the daily trader if the lower frequency is removed, remaining the full spectrum of speculator movement. This technique will be used to preprocess the raw data before being fed to the kernel training.

E. Particle Swarm Optimization Approach

The basic idea of Particle Swarm Optimization (PSO) is combining the local trend and global trend of the swarm. It can be denoted using the following formula.

$$v_{n+1}^i = c_0 v_n^i + c_1 r \left(\frac{p^i - x_n^i}{\Delta t} \right) + c_2 r \left(\frac{p_n^g - x_n^i}{\Delta t} \right), \quad (5)$$

where v_{n+1}^i is the velocity next of the i th particle, compared to the current velocity, denoted as v_n^i . c_0, c_1, c_2 are considered inertia facto, individual confidence, and swarm confidence. r is a random function, p^i is the best position of the i th particle, p_n^g is the best position of all particles, and x_n^i is the i th particle's current position [35] [36].

In this paper, a scheme that mimics this behavior also is applied. The short time interval price movement is what limited eye view can see; however, the long time interval movement is why another trend dominates particular trends. When the price didn't have any meaningful activity for a long time with small exchanges, then the market's liquidity is on the verge of collapse, whether it will go into bullish or

bearish. This collapse can be seen as a conclusion of the market to take another heed. But when the volatility is vast, and the market price is still searching for stability, we can safely say that this movement is remarkably similar in what individual confidence means in the PSO paradigm.

In this paper, the PSO is not used for parameter optimization as previous researches did. This paper extracts the individual confidence and the swarm confidence from the PSO paradigm into a short time interval price window and a longtime interval price window. This technique also will be used to preprocess the raw data before being fed to the kernel training.

F. Data Preprocessing

Bitcoin pairs, just like any other pairs, have five primary data that can be used for elemental analysis:

1. Opening Price (OP),
2. Maximum Price (MxP),
3. Minimum Price (MnP),
4. Closing Price (CP), and
5. Exchange Volume (EV).

Since CP is the next OP of the price, only OP, MxP, and MnP will be evaluated to avoid multicollinearity in this paper.

Let's say $\bar{p} = \langle p_0, p_1, \dots, p_{n-1} \rangle$ is the OP of the Bitcoin through a specific range time. First, the data will be expanded into matrix form, **X** and **Y**. **X** is defined as $\mathbf{X}_{I \times J}$, where $X_{i,j} = p_{i+j}$, $I + J = |\bar{p}|$, and $J = w$, which is the window time. On the other hand, **Y** is defined as $\mathbf{Y}_{I \times 1}$, where $Y_{i,1} = p_{i+w+s-1}$ and s is the number of steps of the price that will be predicted.

For example, if we had data collection from 10 consecutive days within a one-day interval and would like to predict Bitcoin's price at three days after that ten days, then the value of w will be 10, the value of s will be 3. In this paper, the best optimum number of w and s , where $w, s \in \mathbb{N}$, are inspected to yield the best prediction result.

G. Schemes

This paper used the 4 hours interval (4H) within the four-year Bitcoin price movement data range between August 17th, 2017, until July 24th, 2020. Fig. 2 shows the training and testing division. Based on the figure, we can see that the training data set is set to the first 80% of the market movement data.

After that, the data will be evaluated under six massive schemes, as mentioned in Table 1. Scheme A used the raw data sequence based on OP. Scheme B, however, used global maximum based normalization. Hence $\bar{q} = \bar{p} / \max \bar{p}$, where \bar{q} is the normalized sequence. This \bar{q} sequence will be expanded into matrix form too, which is **X**. Scheme C used local maximum within window range for its normalization,

hence its expanded form will be $X_{i,j} = p_{i+j} / \max \langle p_i, p_{i+1}, \dots, p_{i+w-1} \rangle$.

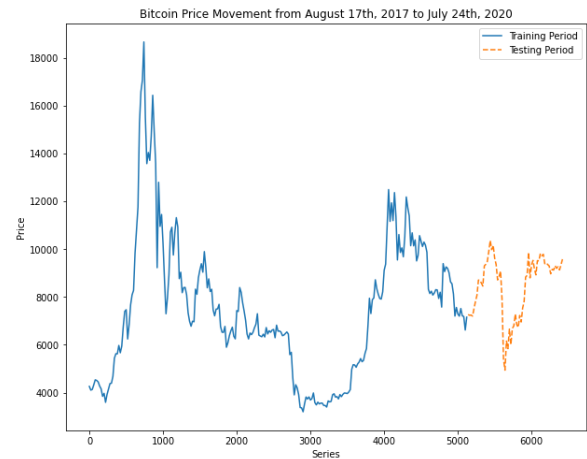


Fig.2. Training and Testing Division

Table 1. Schemes Description

Name	Detail
Scheme A	Using raw data value from OP
Scheme B	Using global max based normalized data value from OP
Scheme C	Using local max based normalized data value from OP
Scheme D	Using combined scheme B and C using multiplication
Scheme E	Using combined scheme B and C using addition
Scheme FFT	Using cropped FFT from OP

Scheme D is the PSO-inspired calculation, where $q_i = p_i^{(g)} p_i^{(l)}$. In this case, $\bar{p}^{(g)}$ is the global max based normalized data, and $\bar{p}^{(l)}$ is the local max based normalized data. Scheme E is similar to the scheme. The only difference is that $q_i = \alpha p_i^{(g)} + (1 - \alpha) p_i^{(l)}$, where α is the weighting rate with range $[0, 1]$.

Schemes A to E will be labeled with the steps s that follow. For example, if scheme A is used to predict the next s -steps of price, it will be labeled as scheme A- s . The steps inspected in this paper are $1 \leq s \leq 4$.

Table 2. Scheme FFT Description

Name	Detail
Scheme FFT-A	Zeroing high frequencies
Scheme FFT-B	Uniforming high frequencies
Scheme FFT-C	Zeroing low frequencies
Scheme FFT-D	Uniforming low frequencies

As shown in Table 2, several schemes are used to inspect the probability of riding the wave based on the FFT definition. Let's assume \bar{p} is the frequency domain version of \bar{p} , and t is the index between $0 < t < |\bar{p}|/2$, where we can consider any index within the range $[0, t)$ as lower frequencies, and any index within the range $[t, |\bar{p}|)$ as higher frequencies. Scheme FFT-A defines $\langle P_t, P_{t+1}, P_{t+2}, \dots, P_{|\bar{p}|/2} \rangle$ as zeros vector. Scheme FFT-B defines $\langle P_{t+1}, P_{t+2}, \dots, P_{|\bar{p}|/2} \rangle$ with the same values as P_t . Scheme FFT-C defines $\langle P_0, P_1, P_2, \dots, P_t \rangle$ as zeros vector. Scheme FFT-D defines $\langle P_0, P_1, P_2, \dots, P_{t-1} \rangle$ with the same values as P_t . This selective cutting is applied with various cutting percentages.

Schemes FFT will be assigned with its respective cutting percentage. For example, if scheme FFT-A sacrifices 25% of its frequencies' elements, it will be labeled as scheme FFT-A 25% cut. The cutting variants are 25%, 50%, and 75%.

III. RESULT

A. Random Test

Random Test is conducted by assuming the price movement is in random walk mode, which is increased or decreased by r_i or random sequences following four different distributions mentioned before. Thus, $p'_{i+1} = p_i + r_i$, where p'_{i+1} is the next price prediction, and p_i is the previous real price. In this test, we just want to predict the movement trend, whether bullish or bearish, resulting in $p'_{i+1} - p_i = r_i$. This result predicted difference price will be compared with the actual difference price, $p_{i+1} - p_i$.

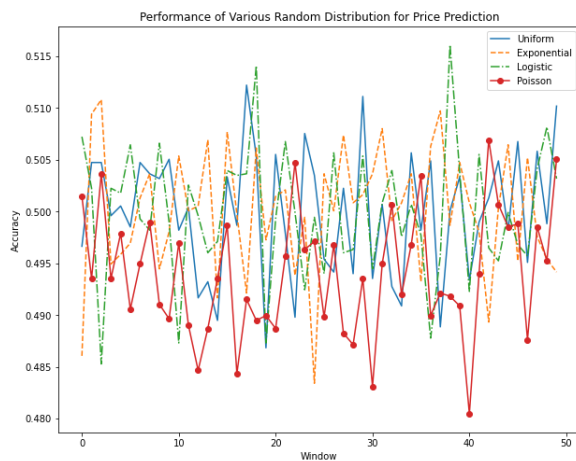


Fig.3. Random Prediction Accuracy

First, the up and down movement of Bitcoin is calculated from 4 years Bitcoin price movement, up for $p_{i+1} - p_i \geq 0$, and down for $p_{i+1} - p_i < 0$. Then, each random distribution will generate the corresponding random walk movement for four years. This process is iterated 100 times. For the Exponential and Poisson distribution, the random walk movement is generated by combining them with Uniform

distribution with a range from $[-1, 1)$ to ensure the generated number has negative and positive values.

The result can be seen in Fig.3, where Uniform, Exponential, Logistic, and Poisson distribution has accuracy 0.5001, 0.4999, 0.5001, and 0.4949, respectively. None of these distributions can even guess higher than 0.51 accuracy. Thus, the Bitcoin market didn't work on a random process. In other words, patterns are dominating the Bitcoin market.

B. Slope Test

In this slope test, two different variables are being tested. The variables used to determine the best combination to predict the next price movement, the number of k steps behind the price that will be observed its difference, and the number of s steps ahead of the price that will be predicted. If $(p_{n+k} - p_n)/k \geq 0$, then $p_{n+k+s+1}$ is considered up, and vice versa. Fig.4 shows the dynamic between two variables in predicting its accuracy with $1 \leq k < 20$ and $1 \leq s < 5$.

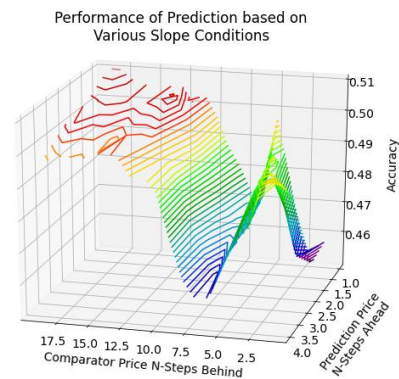


Fig.4. Slope Prediction Accuracy

It can be seen that it seems like the smaller the s value, and the higher the k value, the higher the accuracy it gets. Thus, a further test is performed for $s = 1$ and $20 \leq k < 100$ to verify this tendency.

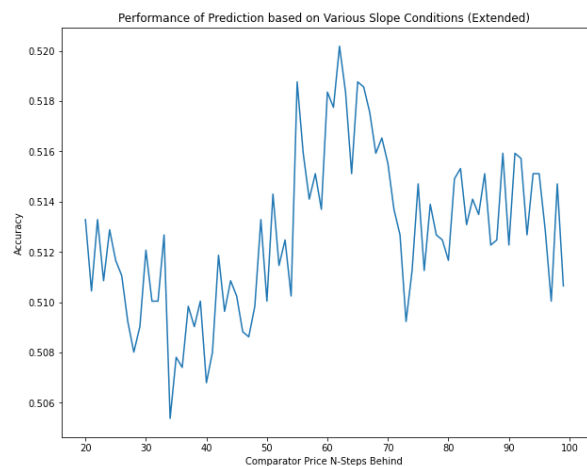


Fig.5. Slope Prediction Accuracy - Extended

Interestingly, Fig.5 has accuracies above 0.5, all of them. It even gets better for 60-ish k , where it reaches

its prediction peak at 0.5202. Compare it with Fig. 4, where most of the accuracies lie below 0.5. The value seems to indicate that, indeed, there is a pattern buried within market noise movement. This result suggests that it doesn't need a lot of samples to determine the next movement. We need to amplify this accuracy.

C. Schemes

The Linear Model test gives insightful results regarding how the market behaves. Fig.6 shows Scheme A of Linear Models performance over four prediction cases. All kernels perform quite excellent in detecting the pattern only within 20-ish window size. Multi-task ElasticNet Regression, however, shows stable yet not so high accuracy in predicting the market movement. Except for Scheme A-4, Multi-task ElasticNet Regression indicates that the higher the window value, the lower the accuracy to predict the next movement. It can be seen that, within Scheme A, Multi-task Lasso Regression and Ridge Regression gives quite an outstanding result in detecting movement regardless of market noise.

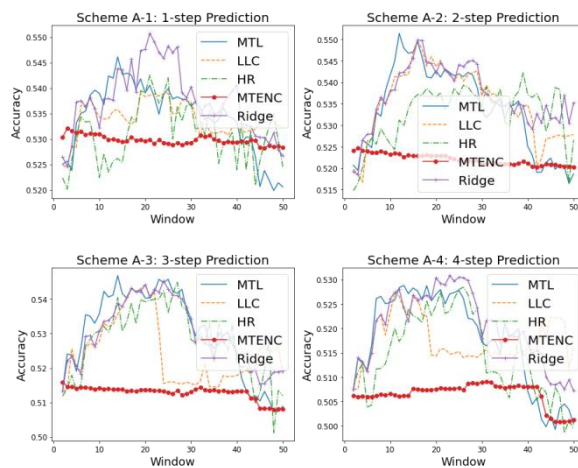


Fig. 6. s-steps Prediction Performance of Scheme A-1, A-2, A-3, and A-4

The case of Multi-task Lasso Regression for higher w value enforces the Multi-task ElasticNet Regression result, which indicates that it doesn't need a high window to understand how the market behaves. The 20-ish window size in 4H interval translates to nearly three days range view. Most of the traders in this time scale, dominated by daily traders, only trade based on a small range of historical data movement, usually until three days ago.

Fig.7 shows the Scheme B performance after the data is normalized with maximum global value; in other words, the input data is restricted only with range $[0,1]$. It can be seen that Multi-task Lasso experiences a significant accuracy drop when the value is normalized. On the contrary, Multi-task ElasticNet keeps its high consistency with higher accuracy. It even gets better for Ridge Regression for higher window value within 2-steps scope prediction.

This result gives us an insightful understanding that even though there is a lot of noise within a short time interval, Ridge Regression provides us with a hint that there is a stabler pattern within a longer window and time interval. Ridge Regression understands it; daily traders dominate even the data.

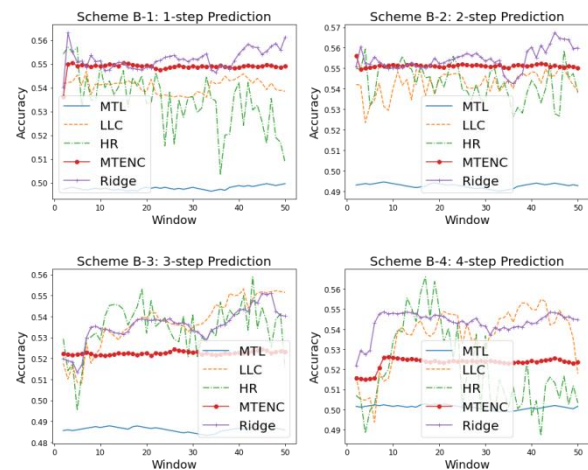


Fig.7. s-steps Prediction Performance of Scheme B-1, B-2, B-3, and B-4

Multi-task ElasticNet result suggests that this kernel is stable, whether normalized or not. Its accuracy is increased for normalized data. Its performance graph even indicates a limit in detecting patterns because of its significant change in accuracy performance. Scheme B-4 performance shows that, for a concise amount of data, 7 window, it didn't give the same accuracy when the window is higher than 7. It can be understood that the highest accuracy reaches its 0.5674 level by Ridge Regression. The lowest accuracy reaches its 0.4832 level by Multi-task Lasso.

Fig.8 shows the Scheme C performance after the data is normalized with maximum local value. Now, the result indicates an even exciting result. Overall, the data has higher accuracy than the previous scheme, especially the 2-steps ahead prediction. All kernels, excluding Multi-task Lasso, move toward the same direction along with a higher window value. Suppose scheme A suggests that we don't need a high window value to predict. In that case, this specialized normalized version of data reveals that more dominating holders exist in a more extended time interval domain.

In other words, based on scheme A, B, and C result, the following deduction can be drawn,

1. Scheme A: the daily trader saw historical data from 3 days ago without analyzing overall movement.
2. Scheme B: there are traders considering the highest and lowest Bitcoin value ever in overall historical movement, yet make the decision based on 7 to 8 days movement.

3. Scheme C: some more dominating holders decide based on maximum local value with historical data for more than eight days.

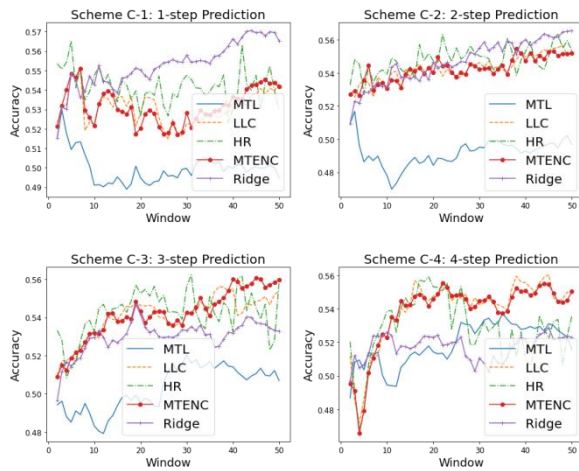


Fig.8. s-steps Prediction Performance of Scheme C-1, C-2, C-3, and C-4

This result gives us a better strategy for navigating the price movement, which we should monitor most, and which one we can consider has the lower benefit. Based on these schemes, it suggests that 2-steps ahead prediction is the most plausible. It can be understood that the highest accuracy reaches its 0.5705 level. The lowest accuracy reaches its 0.4656 level by Multi-task Lasso.

Fig.9 and Fig.10 shows the performance of the weighted price inputted. In this regard, the first scheme is scheme D, where the local max based normalized price is multiplied by its global max based normalized price. The inputs are guaranteed to have a range within [0, 1]. Since scheme D incorporates the short view with the long view of the price movement, it is not surprising that scheme D's performances achieve higher accuracy performance than the previous schemes, shown in Fig.9.

Multi-task Lasso still has the lowest prediction accuracy, which confirms that normalized inputs don't fit well with this Regression. On the other hand, Ridge Regression keeps its performance at the highest accuracy. Not only that, but it also gains the highest accuracy with 0.5734, beating all the previous regression schemes.

It can be understood that the highest accuracy reaches its 0.5734 level by Multi-task ElasticNet. The lowest accuracy reaches its 0.4806 level by Multi-task Lasso. In this scheme, Ridge Regression didn't hint like scheme C, where the higher the window, the more accurate the result. This condition might be caused by the fact that the short time interval is being incorporated, resulting in negating the longer time interval tendencies to be more accurate for a higher window.

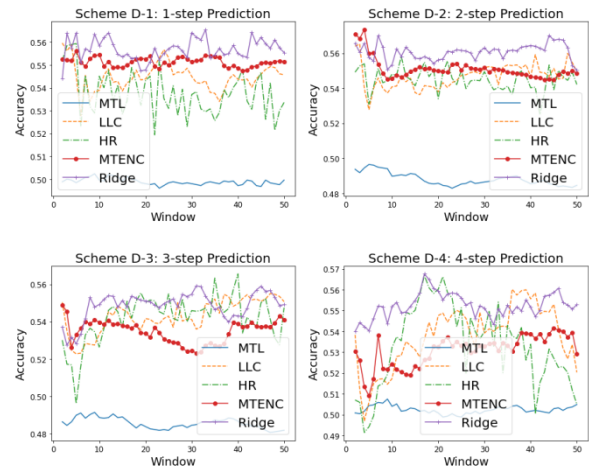


Fig.9. s-steps Prediction Performance of Scheme D-1, D-2, D-3, and D-4

Since longer time interval dominates more than a short time interval, scheme E used $\alpha = 0.75$ value to calculate its formula, $q_i = \alpha p_i^{(g)} + (1 - \alpha)p_i^{(l)}$. Since Multi-task Lasso didn't perform well within the normalized value, this scheme excluded this kernel. The performance of each kernel can be seen in Fig.10.

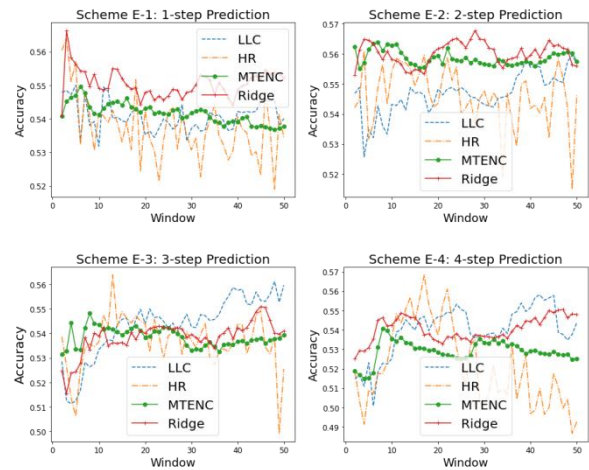


Fig.10. s-steps Prediction Performance of Scheme E-1, E-2, E-3, and E-4

The highest accuracy reaches 0.5684 value, and the lowest accuracy reaches 0.4866 value. The same kernel, Huber Regressor, possesses both numbers. Based on the overall result, Huber Regressor is the most volatile in terms of giving accuracy performance. Ridge Regressor still keeps its stability in its execution. On the other hand, Lasso LARS provides another hint for E-2 and E-3 that a higher window might better predict when the kernel learns historical data higher than eight days. Although this scheme doesn't perform as well as scheme C or D, this scheme gives the best lowest accuracy compared to the others.

For the last scheme, the scheme FFT, where frequencies on certain intervals are being cut, the next price prediction is calculated by checking the last sequence's slope within the selected window. Let \bar{P}' as

a cut version of \bar{P} . The next price tendency is determined by the sign of $p'_{|\bar{p}'|-1} - p'_{|\bar{p}'|-2}$. This scheme FFT uses 1-steps ahead prediction.

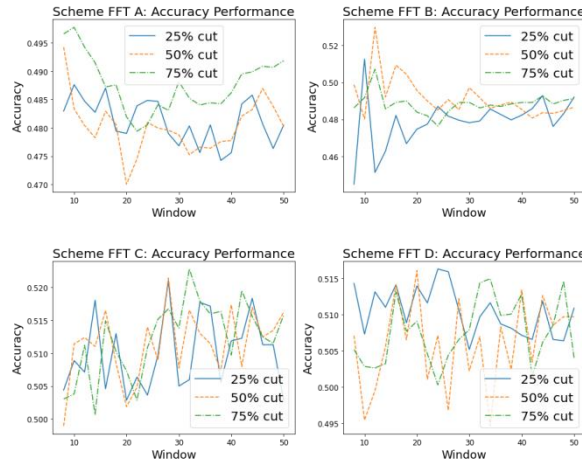


Fig.11. Accuracy Performance of Scheme FFT-A, FFT-B, FFT-C, and FFT-D

As can be seen in Fig.11, this scheme, among many other schemes, gives the lowest accuracy performance. The highest accuracy reaches its 0.5297 level, and the lowest accuracy reaches its 0.4449 level. It has a similar result with the slope test. Thus, it is clear that the majority trend didn't guarantee the next movement.

Table 3. Schemes' Highest and Lowest Accuracy

Name	Highest Accuracy	Lowest Accuracy
Scheme A	0.5514	0.4780
Scheme B	0.5674	0.4832
Scheme C	0.5705	0.4656
Scheme D	0.5734	0.4806
Scheme E	0.5684	0.4866
Scheme FFT	0.5297	0.4449

Table 3 shows the overall result of each scheme. Predicting bitcoin next movement based on weighting local max based normalized data with global max based normalized data through multiplication returns the highest prediction among any schemes. On the other hand, determining the next price based on random thought is proven as a lousy strategy.

IV. DISCUSSION

Since scheme D has the highest accuracy performance, this scheme will be used as the base for our automatic market trader. But first, let's check its hit accuracy performance. All these schemes are predicting whether the next opening price is up or down. But the price movement is not that simple. Even the next opening price is higher than the previous opening price, there is always a probability

that the price went down for a moment before its rising. This kind of movement is captured inside the minimum price and maximum price range. Thus, it begets another question, what is the appropriate price position we should take for the next prediction. The predicted next price should be checked to ensure the decision's safety, whether it is inside the minimum price and maximum price range.

Fig.12 shows the hit accuracy of scheme D. The highest hit accuracy, 80.3%, is attained by Huber Regressor at 2-steps ahead scheme with two window size. The higher the window size, the lower the hit accuracy gets. On the other hand, the lowest hit accuracy, 13.2%, is attained by Huber Regressor at 4-steps ahead scheme with 49 window size.

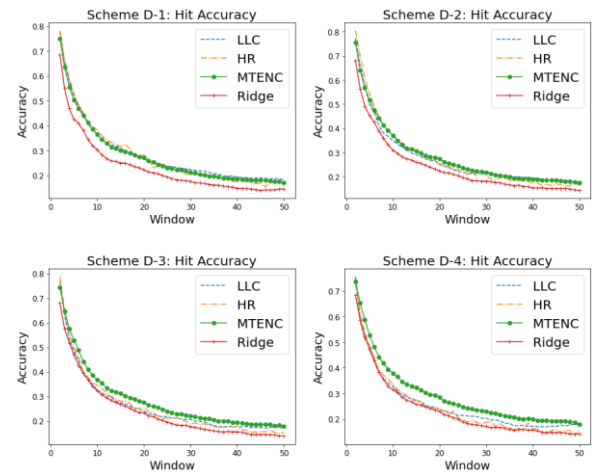


Fig.12. Hit Accuracy of Scheme D-1, D-2, D-3, and D-4

However, this result contradicts our former assumption, where the higher the window, the higher the accuracy becomes. Most Bitcoin markets implement transaction fees, called maker and taker fees. Maker is the condition where the order is processed until when another trader agreed and executed our order. Taker is the condition when the order is immediately executed with the current market liquidity. Since taker order is much more spontaneous than maker order, some markets apply higher fees. For example, Binance used a 0.10% fee for all maker and taker orders. Thus, to make a profit in our trading, if we buy the coin at a price p_n , we should sell our coin at a price p_{n+1} where it should be higher than $p_n / 0.999^2$. In other words, when the bitcoin price moves higher than 0.2003%, then at that point, we can gain profit.

Since this 0.2003% increment is a tricky move for a short time interval, a high hit accuracy rate for a short time interval doesn't guarantee the movement already gives a profit to the trader. Further spot trading simulation is needed to inspect the real profitability of this scheme D-2.

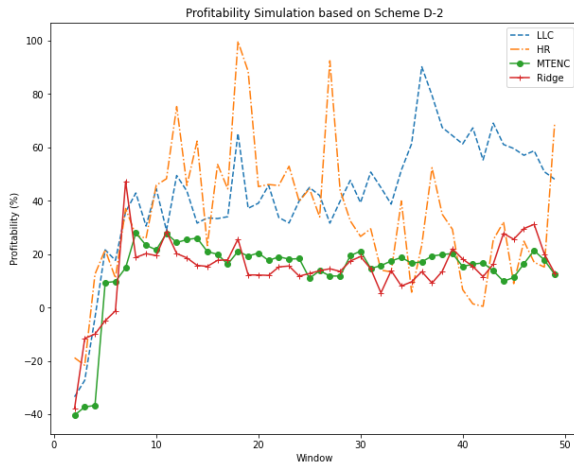


Fig. 13. The Profitability of Scheme D-2

Fig. 13 shows all four kernels' profitability when being a test to do automatic trading based on their kernel result in predicting the bitcoin movement. Each trading simulation is started by zero coins and a certain amount of capital. The automated trading simulation is conducted from around December 26th, 2019, until July 24th, 2020.

The highest performer is claimed by Huber Regressor, where it reaches 99.715% profitability, with 2-steps ahead prediction and window size as 18. In other words, if the trader has \$1000 as his initial capital, with this scheme at the end of the day, he will get \$1997.15 as his final capital. However, the loss will occur for all kernels if the window size is smaller than 4.

Table 4. Scheme D-2 Profitability Statistics

Kernel	μ Profitability	σ Profitability	Max Profitability
LLC	42.893%	22.525%	90.189%
HR	33.945%	25.363%	99.715%
MTENC	14.33%	14.396%	27.942%
Ridge	14.357%	12.489%	47.2304%

Table IV shows that Huber Regressor herd the highest profitability among four kernels, yet the most volatile. On the other hand, Ridge Regressor has the most stable performance for all window size schemes, followed by Multi-task ElasticNet. Thus, Ridge Regressor could be the best choice for low-risk taker stability, although the gain is not that high compared to others. For the high-risk taker, Huber Regressor might be the best choice. Finally, let's look at how Huber Regressor, with its 18 window size and 2-steps ahead scenario, keeps capital growth during Bitcoin panic-selling due to the COVID-19 outbreak.

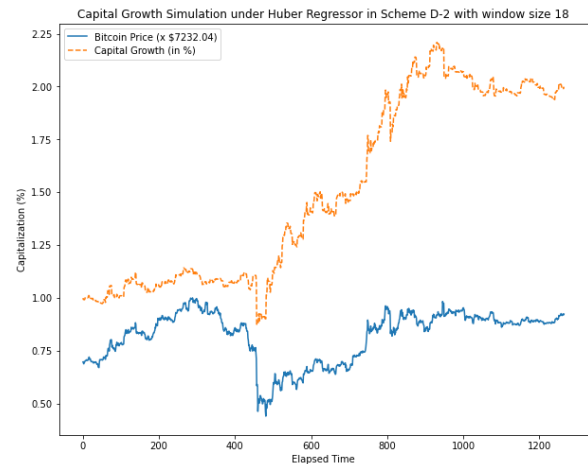


Fig. 14. Capital Growth with Huber Regressor on Scheme D-2 with 18 Window Size

Fig. 14 shows how this scheme guard the capital loss during COVID-19 panic-selling. Even the Bitcoin price fell more than 50%; the initial capital didn't lose more than 75% from its original value—this scheme minimizing the loss during the panic-selling period. After the panic-selling period is over, this scheme boosts the trading even more during a bullish or recovery period.

V. CONCLUSION

It combined a short time with a long interval market trend movement by multiplying their normalized prices. Similar to the PSO approach, a high window period yields promising results for growing capital value in the Bitcoin market using Lasso Lars, Huber, Multi-task Elastic Net, and Ridge Regressor. The best kernel design is not being made for predicting the next price, but for predicting the next second price. This result might be caused by the daily trader's fluctuation that dominates a short time and the trend dominated by traders who decide based on historical data for more than eight days. This combination can be tackled well by Huber Regressor with a relatively medium window size, resulting in protecting capital during the panic-selling and boosting capital during the bullish period.

REFERENCES

- [1] Kristoufek, Ladislav. "On Bitcoin markets (in) efficiency and its evolution." *Physica A: Statistical Mechanics and its Applications* 503 (2018): 257-262.
- [2] Lahmiri, Salim, Stelios Bekiros, and Antonio Salvi. "Long-range memory, distributional variation and randomness of bitcoin volatility." *Chaos, Solitons & Fractals* 107 (2018): 43-48.
- [3] Brandvold, Morten, et al. "Price discovery on Bitcoin exchanges." *Journal of International Financial Markets, Institutions and Money* 36 (2015): 18- 35.
- [4] Brauneis, Alexander, et al. "A High-Frequency Analysis of Bitcoin Markets." Available at SSRN 3478514 (2019).

- [5] Wu, Liang, and Shujuan Chen. "Long memory and efficiency of Bitcoin under heavy tails." *Applied Economics* (2020): 1-12.
- [6] De Filippi, Primavera, and Benjamin Loveluck. "The invisible politics of bitcoin: governance crisis of a decentralized infrastructure." *Internet Policy Review* 5.4 (2016).
- [7] Baqer, Khaled, et al. "Stressing out: Bitcoin "stress testing"." in *International Conference on Financial Cryptography and Data Security*. Springer, Berlin, Heidelberg, 2016.
- [8] Johannsen, Kolja. "Dead Cat Bounce-Demand Reversal Following the Bursting of a Bubble." Available at SSRN 2961304 (2016).
- [9] Baker, Scott R., et al. "The unprecedented stock market reaction to COVID-19." *The Review of Asset Pricing Studies* (2020).
- [10] Al-Awadhi, Abdullah M., et al. "Death and contagious infectious diseases: Impact of the COVID-19 virus on stock market returns." *Journal of Behavioral and Experimental Finance* (2020): 100326.
- [11] Koehn, Maximilian-Benedikt, and Andrejs Cekuls. "A Behavioural Finance Explanation of Speculative Bubbles: Evidence from The Bitcoin Price Development." *Scientific Programme Committee* (2019): 410.
- [12] Pavlyshenko, Bohdan. "Using stacking approaches for machine learning models." in *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. IEEE, 2018.
- [13] Hover, Franz S., and Michael S. Triantafyllou. "System Design for Uncertainty." Cambridge, MA: MIT Center for Ocean Engineering, 2010.
- [14] Bakar, Nashirah Abu, and Sofian Rosbi. "Autoregressive integrated moving average (ARIMA) model for forecasting cryptocurrency exchange rate in high volatility environment: A new insight of bitcoin transaction." *International Journal of Advanced Engineering Research and Science* 4.11 (2017): 237311.
- [15] McNally, Sean, Jason Roche, and Simon 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.
- [16] Shen, Dehua, Andrew Urquhart, and Pengfei Wang. "Does twitter predict Bitcoin?" *Economics Letters* 174 (2019): 118-122.
- [17] Claessen, Koen, Jonas Duregård, and Michał H. Pałka. "Generating constrained random data with uniform distribution." *Journal of functional programming* 25 (2015).
- [18] W. Feller, "An introduction to probability theory and its applications", 2, Wiley (1971).
- [19] Johnson, N. L.; Kotz, S.; N., Balakrishnan. "Continuous Univariate Distributions." Vol. 2 (2nd ed.), (1995).
- [20] Haight, Frank A., *Handbook of the Poisson Distribution*, New York, NY, USA: John Wiley & Sons, (1967).
- [21] Varberg, Dale E., Edwin Joseph Purcell, and Steven E. Rigdon. *Calculus with differential equations*. Pearson/Prentice Hall, 2007.
- [22] Shao, Jun. "Linear model selection by cross-validation." *Journal of the American statistical Association* 88.422 (1993): 486-494.
- [23] Godin, Benoît. "The linear model of innovation: The historical construction of an analytical framework." *Science, Technology, & Human Values* 31.6 (2006): 639-667.
- [24] Lee, Seunghak, Jun Zhu, and Eric P. Xing. "Adaptive multi-task lasso: with application to eqtl detection." *Advances in neural information processing systems*. 2010.
- [25] Januaviani, Trisha Magdalena Adelheid, et al. "The best model of LASSO with the LARS (least angle regression and shrinkage) algorithm using Mallow's Cp." *World Scientific News* 116 (2019): 245-252.
- [26] Chand, Sohail, Sarah Ahmad, and Madeeha Batool. "Solution path efficiency and oracle variable selection by Lasso-type methods." *Chemometrics and Intelligent Laboratory Systems* 183 (2018): 140-146.
- [27] Zhang, Long, and Kang Li. "Forward and backward least angle regression for nonlinear system identification." *Automatica* 53 (2015): 94-102.
- [28] Madigan, David, and Greg Ridgeway. "Discussion of" Least angle regression" by Efron et al." *arXiv preprint math/0406469* (2004).
- [29] P. J. Huber, *Robust Regression: Asymptotics, Conjectures, and Monte Carlo*. *Ann. Statist.* 1 (1973), 799-821.
- [30] R. W. Hill, *Robust Regression When There Are Outliers in the Carriers*. Ph.D. Dissertation, Harvard University, Boston, MA, 1977.
- [31] Rahmanl, Raziur, et al. "Adaptive Multi-task Elastic Net based feature selection from Pharmacogenomics Databases." in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018.
- [32] Kibria, B. M., and Shipra Banik. "Some ridge regression estimators and their performances." *Journal of Modern Applied Statistical Methods* 15.1 (2016): 12.
- [33] Exterkate, Peter, et al. "Nonlinear forecasting with many predictors using kernel ridge regression." *International Journal of Forecasting* 32.3 (2016): 736-753.
- [34] Garrido, Mario, et al. "Hardware architectures for the fast Fourier transform." *Handbook of Signal Processing Systems*. Springer, Cham, 2019. 613-647.
- [35] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." in *Proceedings of ICNN'95-International Conference on Neural Networks*. Vol. 4. IEEE, 1995.
- [36] Du, Ke-Lin, and M. N. S. Swamy. "Particle swarm optimization." *Search and optimization by metaheuristics*. Birkhäuser, Cham, 2016. 153-173.