Journal of Information Systems and Informatics
Vol. 1, No. 1, March 2019

Published By: DRPM-UBD http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

BITCOIN-USD Trading Using SVM to detect The Current day's Trend in The Market

Ferdiansyah[1], Edi Surya Negara[2], Yeni Widyanti[3]

[1]Informatics Department, Bina Darma University, Palembang, Indonesia
[2]Information System Department, Bina Darma University, Palembang, Idnonesia
[3]Acounting Department, Bina Darma University, Palembang, Idnonesia
Email: [1]ferdi@binadarma.ac.id, [2]e.s.negara@binadarma.ac.id, [3]yeni_widyanti@binadarma.ac.id

Abstract

Cryptocurrency trade now is one of type of investment and the market has been treated like similar to foreign exchange and stock market. The one of unique characteristics of bitcoin is keep rising in the last few years. Bitcoin exchange rate to American Dollar (USD) is $3990 USD on November 2018 and daily pice fluctuations could reach 4.55%2. It is interesting to predict value to ensure profitable investment for investor. However, because of its volatility, there's a need for a prediction tool for investors to help them consider investment decisions for bitcoin or other cryptocurrency trade. Nowadays, Automated based tools are commonly used in stock and foreign exchange market predictions. Therefore, this research studied method to predict the current day's trend one of the most used cryptocurrency, it is Bitcoin. The predict methods will be used on this research is regime prediction to develop model to predict the close value of Bitcoin and use Support vector classifier algorithm to predict the current day's trend at the opening of stock market.

Keywords: Cryptocurrency, Prediction Bitcoin, SVM

1. INTRODUCTION

The one of unique characteristics of bitcoin is keep rising in the last few years. Bitcoin exchange rate to American Dollar (USD) is $3990 USD on November 2018 and daily price fluctuations could reach 4.55%2 [1]. It is interesting to predict value to ensure profitable investment for investor. Because of its volatility, there's a need for a prediction tool for investors to help them consider investment decisions for bitcoin or other cryptocurrency trade. Market technical analysis basically identifies the trend of the market in certain period by using historical market price [2]. They give the candle graph and market indicator to help the analysis. Even though market technical analysis is useful, it requires expert to analyse the technical indicators. Therefore, another automation method is needed[3], because There has been much research about SVM prediction on stocks and foreign exchange as case studies but not on cryptocurrency. Machine learning provides capability produce prediction model more accurately without expert knowledge, in stock and forex domain. In this research we use machine learning and use data history from yahoo-finance, and train with Support Vector Machine, SVM is one of the popular machine learning to used predict future and current trends in stock market [4] [5] and we combine with regime algorithm[6][7] to predict Bitcoin-USD current day's trend on yahoo-finance stock market.

2. METHODS

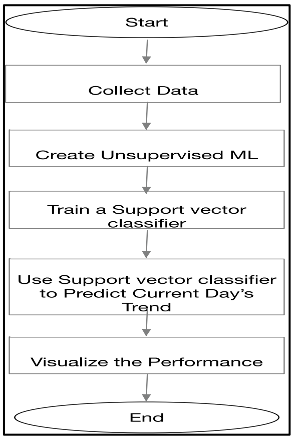
Start
Collect Data
Create Unsupervised ML
Train a Support vector classifier
Use Support vector classifier to Predict Current Day's Trend
Visualize the Performance
End

Figure 1. Methods

2.1. Research Methods

2.1.1 . Dataset

Dataset is from yahoo finance, we take period 4 years, because at 2014 the data is have too many Nan values, so is not good to train the data with many NaN, and will impact to the results. we take the data with symbol "BTC-USD" and saved it as dataframe df, we choose the time period of this data from the year 2015 until 2019.

```
df= web.get_data_yahoo('BTC-USD',start= '2015-01-
01', end='2019-02-28')
df=df[['Open','High','Low','Close']]
```

After this, we created indicators that can be used as features for training the algorithm. Before doing that we decided on the look back time period for these indicators. We select look back period of 10 days, we select 10 to check for the past 2 weeks of traiding data and to avoid the noise in smaller back periods.

```
n = 10
t = 0.8
split = int (t*len(df))
```

We move the High, Low and Columns by one, to access the past data, next we created some of technical indicators we created RSI, SMA, ADX, Correlation, SAR and back to the return the past 1- day on an open to open basis. And this is the result :

Table 1. Sample Result from collect Data

	Open	High	Low	Close	...	Corr	SAR	ADX	Return
Date									
2014-12-31	311.269989	319.089996	308.890015	318.239990	...	NaN	NaN	NaN	NaN
2015-01-01	318.239990	321.359985	313.540009	314.890015	...	NaN	NaN	NaN	0.022145
2015-01-02	314.890015	316.399994	313.079987	315.209991	...	NaN	308.890015	NaN	-0.010582
2015-01-03	315.209991	315.829987	284.890015	287.130005	...	NaN	311.384009	NaN	0.001016
2015-01-04	287.130005	289.940002	255.869995	264.720001	...	NaN	321.359985	NaN	-0.093304

There are many Nan Values. We need to drop them. we droppend the NaN values with this command algorithm.

```
    df = df.dropna()
```

2.1.2. Create Unsupervised Machine Learning

We continue on used a Standard Scaler function and make an unsupervised learning algorithm to create regime prediction[8]. We printed the mean and Covariance value for all the regimes and we plotted the regimes and the new output with indicator as feature and this is the results:

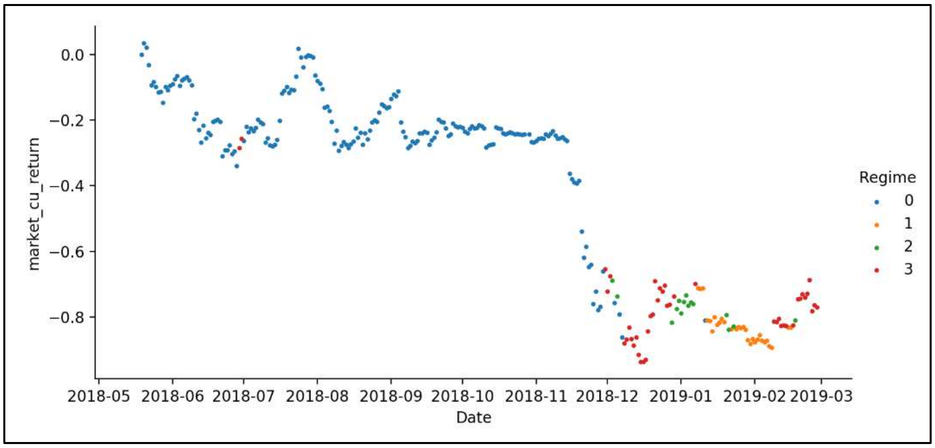

Figure 2. Regime prediction

Based on fig.2

1. The red zone is the low volatility or the sideways zone
2. The purple zone is high volatility zone or panic zone.
3. The green zone is a breakout zone.
4. The blue zone: Not entirely sure but let us find out.

```
('Mean for regime 0: ', 1.42634789312060095)
('Co-Variance for regime 0: ', 1.2409273396350549)
('Mean for regime 1: ', -0.519033358676531)
('Co-Variance for regime 1: ', 0.15086596332070093)
('Mean for regime 2: ', -0.48168648538272163)
('Co-Variance for regime 2: ', 0.16917162258686008)
('Mean for regime 3: ', -0.4206341985600407)
('Co-Variance for regime 3: ', 0.2583432643923261)
```

Table 2. Mean Regime results

Date	Regime	...	Return	market_cu_return
0.2018-05-19	0	...	0.000161	0.000161
1 2018-05-20	0	...	0.033846	0.034007
3 2018-05-21	0	...	-0.013396	0.020611
3 2018-05-22	0	...	-0.052036	-0.031425
4 2018-05-23	0	...	-0.062863	-0.094288

2.1.3. Train in Support Vector Classifier

Next, we scaled the Regimes data frame, excluding the Date and Regimes columns, we make in the earlier piece of code and saved it back in the same columns. By doing that, we will not be losing any other features but the data will be scaled and ready for training with the algorithm support vector classifier. After that, we created the signal column which serves as the prediction values. The algorithm would train on the features set to predict the signal.

```
Regimes.loc[Regimes['Return']>0,'Signal']=1
Regimes.loc[Regimes['Return']<0,'Signal']=-1
Regimes['return'] = Regimes['Return'].shift(1)
Regimes=Regimes.dropna()
```

We continue with a support vector classifier. For this, I used the same SVC model that represented by sklearn.

```
cls=SVC(C=1.0,cache_size=200,class_weight=None,c
oef0=0.0,decision_function_shape=None, degree=3,
gamma='auto',kernel='rbf',max_iter=-
1,probability=False,random_state=None,
shrinking=True,tol=0.001, verbose=False)
```

Support Vector Machine (SVM) is a discriminative classifier that generates a separating hyper-plane. Error tolerance budget is included to make separating hyperplane robust in case of inseparable class data. Linear decision boundaries are augmented to more complex boundary shape through kernel implementation (e.g. polynomial, Gaussian and radial kernel). SVMs have received attention for the impressive performance in classification. Below is formulation of the optimization and constraint model for SVM [9].

$$min_{y,w,b}\ \frac{1}{2}||\omega||^2 + \mathbb{C}\sum_{\mathrm{i}=1}^{\mathrm{m}}\xi\mathrm{i}$$

$$s.t. y^{\mathrm{i}}\left(\omega^T x^{\mathrm{i}} + b\right) \geq 1 - \xi_{\mathrm{i}\,,} i = 1, \dots, m\ \xi_i \geq 0, i = 1, \dots, m. \qquad (1)$$

We split the test data of the unsupervised regime algorithm into train and test data. We use this new train data to train our support vector classifier algorithm, to create the train data we dropped the columns that are not a part of the feature set:

```
‘Signal’,’Return’,’market_cu_return’,’Date’
```

Then we fit the X and y data sets to the algorithm to train it on

```
split2= int(.8*len(Regimes))
X =
Regimes.drop(['Signal','Return','market_cu_retur
n','Date'], axis=1)
y= Regimes['Signal']
cls.fit(X[:split2],y[:split2])
```

We calculated the size of test sets and predictions indexed accordingly to the data frame. the original return values of 'BTC-USD' are stored in df, while those in Regimes is scaled hence, won't be useful for taking a cumulative amounts to check for the performance.

```
p_data=len(X)-split2.
```

3. RESULTS AND DISCUSSION

We store the predictions made by the SVC in the column. Based on these signals we calculated the returns of the strategy by multiplying the signal at the beginning of the day with the return at the opening.

Table 3. Sample Data Prediction result signal

Date	Signal
2019-02-17	1
2019-02-18	1
2019-02-19	1
2019-02-20	1
2019-02-21	1
2019-02-22	1
2019-02-23	1
2019-02-24	1
2019-02-25	1
2019-02-26	1
2019-02-27	1
2019-02-28	1
2019-03-01	1

Finally, we calculated cumulative strategy returns and the cumulative market returns and stored them in df. Then, we calculated the sharpe ratio to make sure and measure the performance.

Published By: DRPM-UBD http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

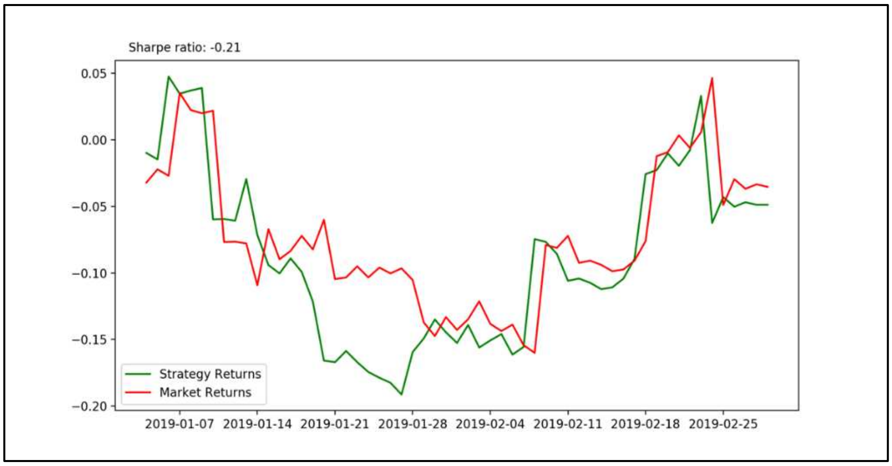

Figure 3. Strategy Return and Market Return

Based on Fig.3 , we look and the result based on Strategy Return is -0,05 and market return is in -0,03, and prediction for 01-03-2019 is still minus -0,04 so is not for investment on bitcoin for this day because the value is fall from 24-02-2019 it means negative on stock market. [10]

4. CONCLUSION

SVM and regime prediction can be use to predict current day's trend of Bitcoin on the market, but we need more knowledge to read the data, by expert on reader signal market and strategy return on the trading market. Future research we suggest to use deep learning such as ANN and other technical analytic ex. Market Technical Analysis that we mention on introduction to improve and provide the best results.

REFERENCES

[1] B. B. Worldwide, "The Bitcoin Volatility Index," 2018. [Online]. Available: https://www.buybitcoinworldwide.com/volatility-index/. [Accessed: 04-Dec-2018].

[2] J. J. Murphy, "Technical Analysis Of The Financial Markets," *Pennsylvania Dental Journal*. 1999.

[3] A. Radityo, Q. Munajat, and I. Budi, "Prediction of Bitcoin exchange rate to American dollar using artificial neural network methods," in *Advanced Computer Science and Information Systems (ICACSIS), 2017 International Conference on*, 2017, pp. 433–438.

[4] B. J. de Almeida, R. F. Neves, and N. Horta, "Combining Support Vector

Machine with Genetic Algorithms to optimize investments in Forex markets with high leverage," *Appl. Soft Comput.*, vol. 64, pp. 596–613, 2018.

[5] L. Kašćelan, V. Kašćelan, and M. Jovanović, "Hybrid support vector machine rule extraction method for discovering the preferences of stock market investors: Evidence from Montenegro," *Intell. Autom. &amp; Soft Comput.*, vol. 21, no. 4, pp. 503–522, 2015.

[6] E. Su, "Stock index hedging using a trend and volatility regime-switching model involving hedging cost," *Int. Rev. Econ. Financ.*, vol. 47, pp. 233–254, 2017.

[7] E. Panopoulou and T. Pantelidis, "Regime-switching models for exchange rates," *Eur. J. Financ.*, vol. 21, no. 12, pp. 1023–1069, 2015.

[8] Varun Divakar, "Trading Using Machine Learning In Python – SVM (Support Vector Machine)," 2017. [Online]. Available: Trading Using Machine Learning In Python – SVM (Support Vector Machine). [Accessed: 10-Feb-2019].

[9] A. Greaves and B. Au, "Using the bitcoin transaction graph to predict the price of bitcoin," *No Data*, 2015.

[10] G. Saras, "Sekilas Tentang CAPM dan Cara Perhitungannya," 2017. [Online]. Available: https://rumahsaraswati.co/capm-dan-cara-perhitungannya/. [Accessed: 01-Mar-2019].