

Bitcoin Price Forecasting with Support Vector Machine and Autoregressive Integrated Moving Average Model

Benjamin Carpenter, Shuning Jin, Jacob Pauly, Tristan Larsin

University of Minnesota Duluth
1049 University Drive
Duluth, Minnesota 55812

1 Introduction

In this project, we will use machine learning methods to predict bitcoin prices based on historical market data. It would empower businesses and individuals to make better financial investments. It would also assist in understanding the probabilities of risk and reward for specific investments. We are specifically interested in investigating two methods: Autoregressive Integrated Moving Average (ARIMA) and Support Vector Machine (SVM). The objectives of the project are as follows:

- Implementing ARIMA and SVM algorithms for time series forecasting
- Using the algorithms to predict future prices based on previous prices
- Evaluating and comparing the performance of the two algorithms

In the rest of the paper, we will introduce related work in Section 2, our proposed work in Section 3, experimental evaluation in Section 4, and conclusions and future work in Section 5.

2 Related Work

The work proposed by Zhang et al. (Zhang 2003) is a hybrid methodology to combine the linear ARIMA and non-linear ANN models for time series forecasting to improve performance. The empirical results with three real data sets suggest that the hybrid model outperforms its component model, and Canadian lynx data shows 19% decrease in MSE and 8% decrease in MAD.

In a 2014 white paper (Leung, MacKinnon, and Wang 2014), a team from the University of Manitoba uses structural support vector machines (SSVMs) in an attempt to predict future stock market prices. SSVMs are a specialized form of supervised learning algorithm that can perform linear and non-linear regression to predict similar outcomes from past data (Tsochantaridis et al. 2004). In their results, they found the algorithm was performing with 78% accuracy.

In 2017, Facebook developed Prophet (Taylor and Letham 2017), which is a powerful tool for time series forecasting

available in Python and R. Prophet is aimed at efficiently producing high quality forecasts with large number of time series data points. They used the generalized additive model (GAM), which is based on time series decomposition. Three main components are used: trend, seasonality, and holiday. GAM has great flexibility when used non-parametrically.

Autoregressive Integrated Moving Average (ARIMA) was used by Vinay et al. (Vinay et al. 2013) to predict road-traffic volume. ARIMA is used for short-term and long-term historical data. There are several variations of ARIMA that are best suited for given problems. This research tested all of the variations to compare the different accuracies of his road-traffic volume predictions. Some variations have more overhead than others, but lack accuracy. Based purely on accuracy ARIMA-GARCH outperformed all other variations of ARIMA.

In a 2016 paper (Ved et al. 2016), a team used a support vector machine (SVM) and kernel functions to forecast the stock market. They separated the stock market changes into two classifications, rising and declining. When rising, the value at $x - 1$ is less than that at x . When declining, the value at $x + 1$ is less than that at x . The types of SVM kernels used were linear, polynomial, and radial basis. The most accurate SVM kernel performed with an 88.34% accuracy.

Additional methods used by David Sheehan (Sheehan 2017) including the long short term memory (LSTM) model, which is a type of deep learning that can predict cryptocurrency prices. LSTM models are well suited for time series data because loops in neural nets allow for persistent data. In essence, they can remember how time series data has behaved in the past. Sheehan decided to use the same Bitcoin dataset that we have chosen. This dataset includes Bitcoin's details at a one-minute interval from January 2012 through January 2018. This provided him with six years of data. Testing his algorithm from July 1st, 2017 through November 1st, 2017 he received a mean absolute error (MAE) of 0.0392.

In our project, we will apply SVM and ARIMA for bitcoin price prediction. While SVM and ARIMA have been widely applied for stock price forecasting, we want to investigate their predictive power in the context of bitcoin. Also, we are interested in comparing the performance of these two algorithms.

3 Proposed Work

In this section, we will discuss the two proposed algorithms: ARIMA and SVM, as well as evaluation metrics of model performance.

3.1 Autoregressive Integrated Moving Average Model (ARIMA)

ARIMA model is a generalization of ARMA model, with the form:

$$\phi(B)(1 - B)^d X_t = \delta + \theta(B)w_t \quad \text{for } t = 1, 2, \dots$$

where $\delta = \mu(1 - \sum_{i=1}^p \phi_i)$, w_t is white noise with independent and identical distributions, ϕ and θ are vectors of unknown fixed regression coefficients, and B is back-shifting operator defined as $B^d x_t = x_{t-d}$.

(i) Model Specification

The ARIMA(p, d, q) model consists of three parts:

Autoregression of order p : AR assumes current value x_t is correlated with p past values, where p denotes the number of time lags. The autoregressive process models the trend of x_t .

Moving average of order q : MA assumes current value x_t is correlated with q recent white noises. The moving average smoother is to remove white noise in data and helps discover underlying long-term trend.

Differencing of order d : time series are usually modeled by a stationary component and a nonstationary component. Performing differencing is to remove trend and produce stationarity. First difference is defined as $\nabla x_t = x_t - x_{t-1}$.

(ii) Model Fitting

Given p, d, q order for ARIMA model, we shall firstly do d -order differencing and reduce it to ARMA(p, q). Secondly, we focus on estimating parameters ϕ and θ in ARMA model. There are two typical techniques for such estimation: Yule-Walker estimation and maximum likelihood estimation.

3.2 Support Vector Machine (SVM)

SVM constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for classification, regression. It allows flexible mapping of high dimensional features to capture non-linear relationships with regularization to avoid over-fitting. Our goal is draw a dividing line between classes which maximizes the the space between the line and the nearest points (known as the 'margin'). In sigma notation, we want to minimize $W(\alpha)$:

$$W(\alpha) = \sum_{i=1}^l \alpha_i + \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l y_i y_j \alpha_i \alpha_j (x_i \cdot x_j)$$

where α is the vector of l non-negative Lagrange multipliers to be determined, and C is a constant.

In order to construct W , Lagrange multiplication technique is applied to an optimization problems bounded by a constraint. Here, the constraint that must be satisfied is:

$$\sum_{i=1}^l y_i \alpha_i = 0 \quad 0 \leq \alpha_i \leq C \quad \forall i$$

Note that only the closest points to the hyperplane (dividing line) are important for the problem. We could remove all other points and arrive at the same solution. Those points and their derived vectors are the 'support vectors' in SVM. By adding dimensions, we can apply SVM to many independent variables.

3.3 Evaluation Metrics

(i) Goodness of Fit

Akaike's Information Criterion (AIC)

$$AIC = \log \hat{\sigma}_k^2 + \frac{n + 2k}{n}$$

AIC, Bias Corrected (AICc)

$$AICc = \log \hat{\sigma}_k^2 + \frac{n + k}{n - k - 2}$$

Bayesian Information Criterion (BIC)

$$BIC = \log \hat{\sigma}_k^2 + \frac{k \log n}{n}$$

Mean Square Error (MSE)

$$MSE = \frac{SSE(k)}{n - k}$$

where $\hat{\sigma}_k^2$ is the maximum likelihood estimator of variance, SSE is residual sum of squares, k is the number of parameters, and n is the number of observations.

(ii) Prediction Accuracy

Mean Absolute Percentage Error (MAPE)

$$M = \frac{100}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

where A_t is the actual value and F_t is the forecast value.

4 Experimental Evaluation

4.1 Dataset

We use a dataset of historical exchanges of Bitcoin from January 1, 2012 to January 8, 2018 (Kaggle). The data is based on every minute update, and contains 3,161,057 observations in total. It has 8 attributes: timestamp (in Unix time), open, high, low, close, volume in BTC, volume currency, and weighted price. The csv file has a size of 212.8 MB.

4.2 Preprocessing

The original dataset is based on every minute data from 2012-01-01 to 2018-01-08. Initially, we transform it to daily data by adopting the observations on 00:00 for each day. We use the close price as both the response variable and the explanatory variable. Figure 1 shows the trend in early years is relatively trivial. Thus, we decide to only use data from 2017-01-01 to 2018-01-01, which has 366 observations. To stabilize the variance, log transform is applied. The data after preprocessing is shown in Figure 2.

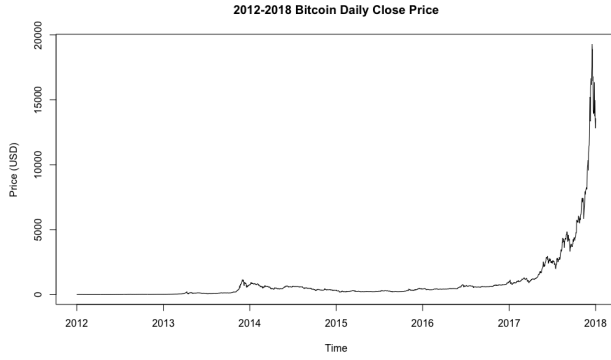


Figure 1: 2012-2018 Bitcoin Daily Close Price

4.3 Evaluation Method

To evaluate model performance, we divide the data into training set and testing set using a 90:10 split. Due to the nature of time series data, we could not do random sampling here. We make forecasts based on historical data, and test error is measured by MSE and MAPE. There are two types of forecast: one-step-ahead and m-step-ahead (long range). In one-step-ahead forecast, each next point is predicted based on past p (autoregressive order) true values. In m-step-ahead forecast, only the first 90% points are used for fitting, and the future points are predicted by recursive one-step-ahead forecast. In conjunction with past true values, newly predicted values are also used to prophesy the next immediate point, and the procedure repeats m times.

4.4 ARIMA Experiment

4.4.1 Model Building

There are three hyperparameters in $ARIMA(p, d, q)$ model. We want to identify the differencing order d , and sample ACF $\hat{\rho}(h)$ serves a criterion - that is, slow decay in ACF indicates a need for differencing. Since the ACF of our data (Figure 3) shows such pattern, we do first differencing. After that, the ACF (Figure 5) suffices and no further differencing is needed. Hence, we decide $d=1$. The log first difference $\nabla \log(x_t)$ is also called return or growth rate, shown in Figure 4.

We also need to identify the autoregressive order p and moving average order q . Above all, we need to determine if $p=0$ or $q=0$, which correspond to $MA(q)$ or $AR(p)$ respectively. We use the sample ACF and PACF behavior as a guide. Since there are no cut off after certain lags, and both ACF (Figure 5) and PACF (Figure 6) tail off, we can conclude p and q are nonzero, leading to $ARMA(p, q)$ model. For simplicity, we assign $p=1$ and $q=1$. That is, we adopt $ARMA(p=1, q=1)$ for the first differenced data, which corresponds to $ARIMA(p=1, d=1, q=1)$ for the original log transformed data. The model adequacy will be more rigorously examined below.

4.4.2 Model Checking

With all three hyperparameters specified, we carry out preliminary model checking by residual analysis. Applying the

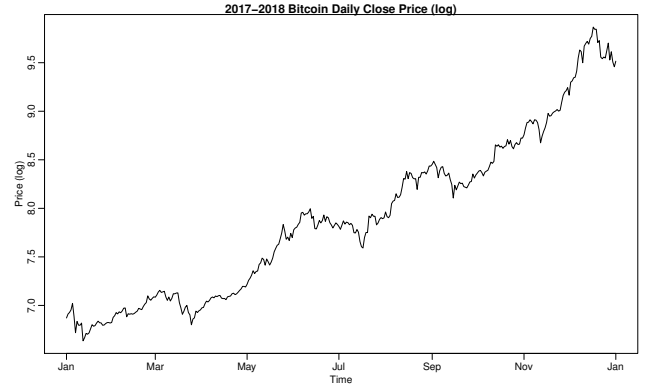


Figure 2: 2017-2018 Bitcoin Daily Close Price with Log Transformation

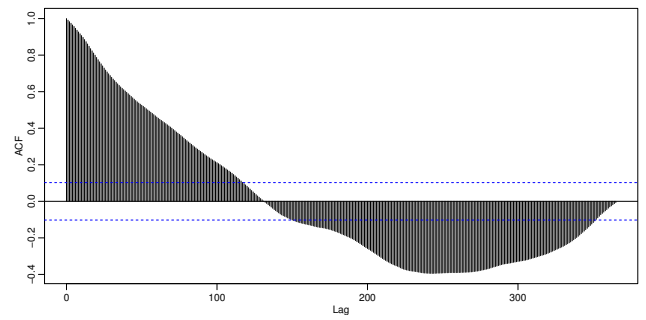


Figure 3: ACF - Without Differencing

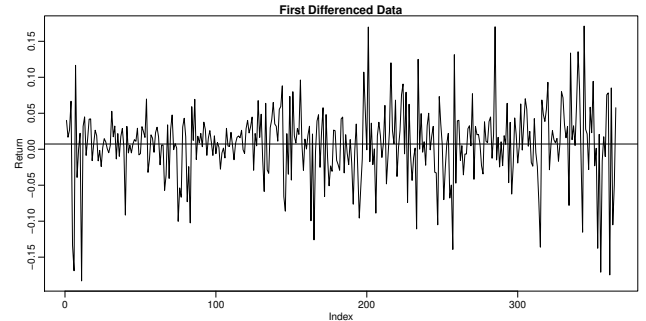


Figure 4: Log First Differenced Price

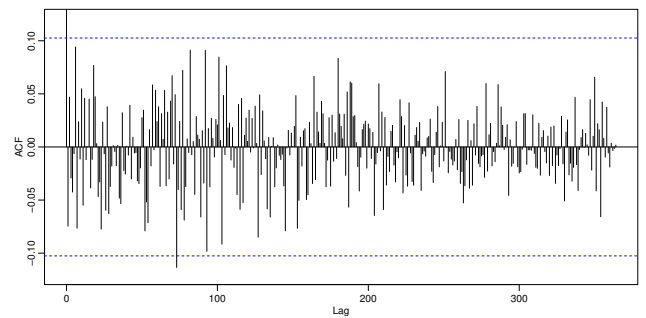


Figure 5: ACF - First Differencing

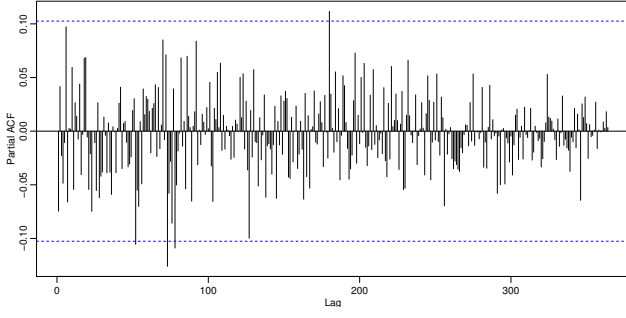


Figure 6: PACF - First Differencing

data, the 'sarima' method in 'astsa' package gives 4 diagnostic plots (Figure 7) of standardized residuals

$$e_t = \frac{x_t - \hat{x}_t^{t-1}}{\sqrt{\hat{P}_t^{t-1}}}$$

where \hat{x}_t^{t-1} is the one-step-ahead prediction of x_t , and \hat{P}_t^{t-1} is the estimated one-step-ahead error variance.

(a) Ideally, the residuals should have iid distribution with $\mu = 0$ and $\sigma^2 = 1$. The time plot shows no obvious patterns and the mean is around 0. But a few values exceed 3 standard deviations in magnitude, which could be outliers.

(b) To check the assumption of randomness, we inspect the sample ACF. There are no significantly large values, indicating no apparent departure from this assumption.

(c) The normal Q-Q plot is heavy tailed. This means the assumption of normality may not strictly hold.

(d) The p values for Ljung-Box Q-statistic are large (greater than 0.2), thus we do not reject the null hypothesis of model adequacy.

Thus, we conclude that this model is suitable for our dataset in general.

4.4.3 Model Performance

Since 1 data point is lost due to previous differencing, we have 365 data in total. First, after train-test split, we use first 328 for train and the last 37 for test. Second, we fit the model with training data. The train MSE is 0.0022. Third, one-step-ahead forecast gives MSE of 0.0063 and MAPE of 43.7934. In m-step-ahead forecast with $m = \text{test size} = 37$, MSE is 0.0064 and MAPE is 23.4065. The forecast results for log first differenced price are shown in Figure 8.

4.5 SVM Experiment

4.5.1 Model Building

When applying SVM to time series data, we need to integrate autoregression of order p . We choose $p=1$ to comply with ARIMA experiment. In this experiment, we adopt SVM with Radial Basis Function (RBF) kernel, using parameters $C=100$, $\gamma=0.02$, and $\epsilon=0.01$.

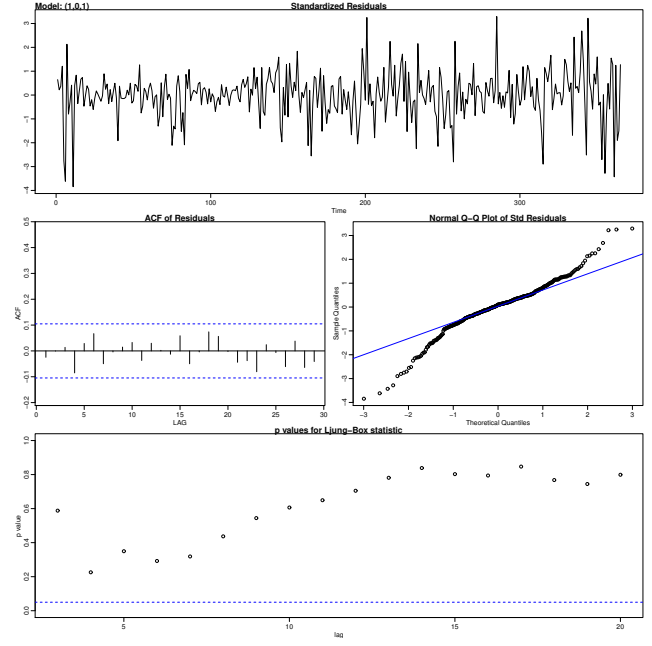
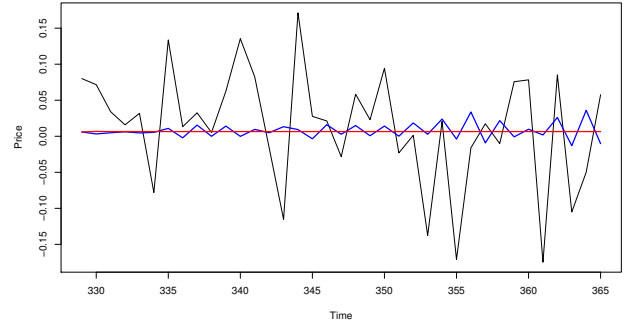
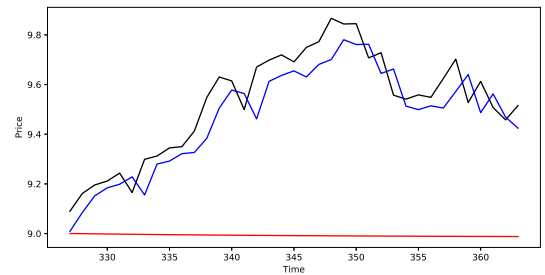


Figure 7: Residual Diagnostics



Black: true; Blue: one-step-ahead; Red: m-step-ahead.

Figure 8: ARIMA Forecast for Log First Differenced Price



Black: true; Blue: one-step-ahead; Red: m-step-ahead.

Figure 9: SVM Forecast for Log Price

4.5.2 Model Performance

Since 1 data point is lost in autoregression of order 1, we have 365 data in total. First, after train-test split, we use first 328 for train and the last 37 for test. Second, we fit the model with training data. The train error is measured by $MSE = 0.0022$. Third, one-step-ahead forecast gives MSE of 0.0086 and $MAPE$ of 0.8527. In m -step-ahead forecast with $m = \text{test size} = 37$, MSE is 0.3308 and $MAPE$ is 5.5688. The forecast results for log price are shown in Figure 9.

5 Conclusions and Future Work

For both ARIMA and SVM, the training errors are small. However, test errors are relatively large. One-step-ahead forecast is generally more accurate than m -step-ahead forecast. In fact, m -step-ahead prediction produces a sequence with very small changes, which is ineffective. In the future, we would like to try different strategies for long range forecast other than recursive one-step-ahead approach. Second, to make a reasonable comparison between ARIMA and SVM, it is necessary to transform the results to a consistent differencing order in post-processing. Third, we want to explore other hyper-parameters to find optimal settings for the models, which may lead to higher prediction accuracy.

References

- Kaggle. Bitcoin historical data at 1-min intervals. <http://www.kaggle.com/mczielinski/bitcoin-historical-data>. Accessed: 2018-02-05.
- Leung, C. K. S.; MacKinnon, R. K.; and Wang, Y. 2014. A Machine Learning Approach for Stock Price Prediction. In *Proceedings of the 18th International Database Engineering*, 274–277. New York, NY, USA: Association for Computing Machinery.
- Sheehan, D. 2017. Predicting cryptocurrency prices with deep learning.
- Taylor, S. J., and Letham, B. 2017. Forecasting at scale. *PeerJ PrePrints* 5:e3190v2.
- Tsochantaridis, I.; Hofmann, T.; Joachims, T.; and Altun, Y. 2004. Support vector machine learning for interdependent and structured output spaces. In *Twenty-First International Conference on Machine Learning, ICML 2004*.
- Ved, P. U.; Subhash, P.; Ramchander, M.; and Ravindra, P. 2016. Forecasting stock market movements using various kernel functions in support vector machine. In *Proceedings of the International Conference on Advances in Information Communication Technology Computing, AICTC 2016*. New York, NY, USA: Association for Computing Machinery.
- Vinay, G.; Gagan, G.; Ajay, G.; and Rajeev, A. 2013. Assessment of arima-based prediction techniques for road-traffic volume. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems, MEDES 2013*, 246–251. New York, NY, USA: Association for Computing Machinery.
- Zhang, G. P. 2003. Times series forecasting using a hybrid arima and neural network model. *Neurocomputing* 50:159–75.