Comparison of ARIMA Time Series Model and LSTM Deep Learning Algorithm for Bitcoin Price Forecasting

Ebru Şeyma Karakoyun[1], Ali Osman Çıbıkdiken[2]
[1]Necmettin Erbakan University, Industrial Engineering Master Program, Konya, Turkey
[2]Necmettin Erbakan University, Department of Computer Engineering, Konya, Turkey
[1]*ebruseymakarakoyun@gmail.com*, [2]*aocdiken@konya.edu.tr*

Abstract

In this study, ARIMA Time Series Model and the LSTM Deep Learning Algorithm have been compared to estimate the future price of Bitcoin. The ARIMA model, which is widely used in the prediction of time series, is run in the R programming language. The LSTM model, which is the machine learning algorithm used in predicting the time series, is also built using the Keras framework in Python. The prices of the next 30 days were estimated with the obtained models. The results have been obtained approximately MAPE 11.86% with ARIMA and MAPE 1.40% with LSTM. Also according to other accuracy test results, it is observed that the LSTM Model is more successful when the results are compared.

Keywords: Bitcoin, time series, deep learning, forecasting, ARIMA
Main Conference Topic: Computer Science, Artificial Intelligence

Introduction

Bitcoin has been proposed as an electronic payment system based on mathematical proof and cryptographical method by a mysterious software developer going by the name of Satoshi Nakamoto in 2008 (Nakamoto, 2008). The system can work independently of any central authority. Money can be transferred electronically in secure, provable and immutable. Bitcoin became one of the first digital currencies to use the decentralized network to promote instant payments after Satoshi's paper. Although Bitcoin was not designed as an investment, many people purchase Bitcoin and cryptocurrencies on exchanges for investment. Today, Bitcoin can be traded on over 40 exchanges worldwide. Market size reached about 11 billion dollars in April 2018 (https://coinmarketcap.com/charts/). The price of Bitcoin is far higher than of fiat currencies. This potential requires forecasting the future price of Bitcoin. The latest price of Bitcoin (currency symbol: BTC or XBT) can be found on the exchanges. Closing prices for any period (a month, a week, a day, one hour, etc.) are used to obtain price graphics. Bitcoin prices graph of 2013-2018 years has been shown in Figure 1. The daily closing prices were obtained from bitcoin.com (Bitcoin launched in 2009. But there are no data on exchanges before 2013).

Figure 1: Bitcoin price of 2013-2018 years (bitcoin.com)

Time series concern data that has been obtained over time. Therefore, Bitcoin prices can be accepted as a time series. Time series forecasting applies in many practical fields such as business, economics, finance, and engineering, etc. Many researchers show neural networks are one of the most successfully applied technique in the financial forecasting problems (Gallo, 2005), (Bodyanskiy, & Popov, 2006), (Navon, & Keller, 2017), (Xiong et al., 2015), (Fischera, & Kraussb, 2017). Time series prediction models and machine learning models are used to estimate Bitcoin prices. Madan et al. (2014) have predicted the sign of the daily price change with an accuracy of 98.7% by using machine learning algorithms. Georgoula et al. (2015) used time-series analysis to study the relationship between Bitcoin prices, and Twitter feeds by using Support Vector Machine (SVM) Algorithm. Jang et al. (2018) revealed the effect of Bayesian neural networks (BNNs) by using the time series of Bitcoin process. The theoretical framework for time series analysis and real-time algorithms for Bitcoin price prediction have been developed by Amjad, & Shah (2016). Abu Bakar et al. (2017) implemented the forecast ARIMA (2, 1, 2) model by using monthly Bitcoin price data and obtained mean absolute percentage error for forecasting is 5.36%.

According to related works, ARIMA time series forecast model and LSTM deep learning algorithms are convenient for forecasting Bitcoin price.

Methodology

Two prediction models have been proposed using by R and Python Keras to compare time series ARIMA model and LSTM deep learning algorithm.

Data Set

The data set consists of Bitcoin price sampled at daily between April 28, 2013 and October 29, 2017 (source: www.quandl.com). 1646 daily Bitcoin prices have been used for training. The obtained models predict Bitcoin prices for next 30 days.

Used Technology

R has been created as a language and platform for statistical computation and graphics. R is an integrated suite of software tools for data manipulation, calculation, and graphical display. R allows performing time series analysis using interactive mode or programming (Coghlan, 2017). "forecast", "ggplot2", and "tseries" R libraries have been used for creating

ARIMA model in this study. Keras deep-learning library has been built on Python. It has minimalist API to build Deep Learning models easily (Chollet, 2015).

ARIMA Model

ARIMA model is a type of statistical models for forecasting time series data (Box, & Jenkins, 1970). Non-stationary time series is made stationary by using finite differencing of the data in ARIMA models. ARIMA is an acronym that represents AutoRegressive Integrated Moving Average.

- **AR** (Autoregression): The dependent relationship between observation and some number of lagged observations.
- **I** (Integrated): The use of differencing of raw observations to obtain the time series stationary.
- **MA** (Moving Average): The dependency between an observation and a residual error from a moving average model used to lagged observations.

ARIMA(p,d,q) is a standard notation that depicts the integer value parameters to indicate the ARIMA model. The parameters of the ARIMA model are represented as follows:

- p: The number of lag observations.
- d: The number of times that the raw observations are differenced.
- q: The size of the moving average window.

The full mathematical model can be written as:

$$y'_t=c+\prod_1 y'_{t-1}+...+\prod_p y'_{t-p}+\theta_1\varepsilon_{t-1}+...+\theta_q\varepsilon_{t-q}+\varepsilon_t \qquad (1)$$

where y'_t is the differenced series (Hyndman, & Athanasopoulos, 2012). It is much easier to work with the backshift notation to combine components. (1) can be written in backshift notation as:

$$(1-\Phi_1 B-...-\Phi_p B^p)\ (1-B)^d y_t=c+(1+\theta_1 B+...+\Phi_q B^q)\varepsilon_t \qquad (2)$$

It can be difficult to choose appropriate values for p, d, and q. However, the auto.arima() function in R will do it automatically.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) unit root test has been used to check stationary. The null hypothesis is that the data are stationary. In this case, small p-values (e.g., less than 0.05) suggest that difference is required (Hyndman, & Athanasopoulos, 2012).

Following steps has been introduced to create ARIMA model.

Proposed Process ARIMA Model:
Step 0. Load data set and libraries
Step 1. Create time series data
Step 2. Check stationary
Step 3. Check seasonality
Step 4. Obtain ARIMA model

Step 5. Create forecast model
Step 6. Create train data and test data
Step 7. Test accuracy results

Proposed ARIMA Model has been run using by R. R sources codes and outputs have been shown below:

Step 0. Load data set and libraries
```
> library(forecast)
> library(ggplot2)
> library(tseries)
> library(readr)
> btc_usd_daily <- read_csv("btc_usd_daily.csv")
```

Step 1. Create time series data
```
> BTC <- ts(btc_usd_daily,frequency=365)
```

Step 2. Check stationary
```
> kpss.test(BTC)
> autoplot(BTC)
```

KPSS test computed p-value = 0.01. This result shows that data set is not stationary (Figure 2).

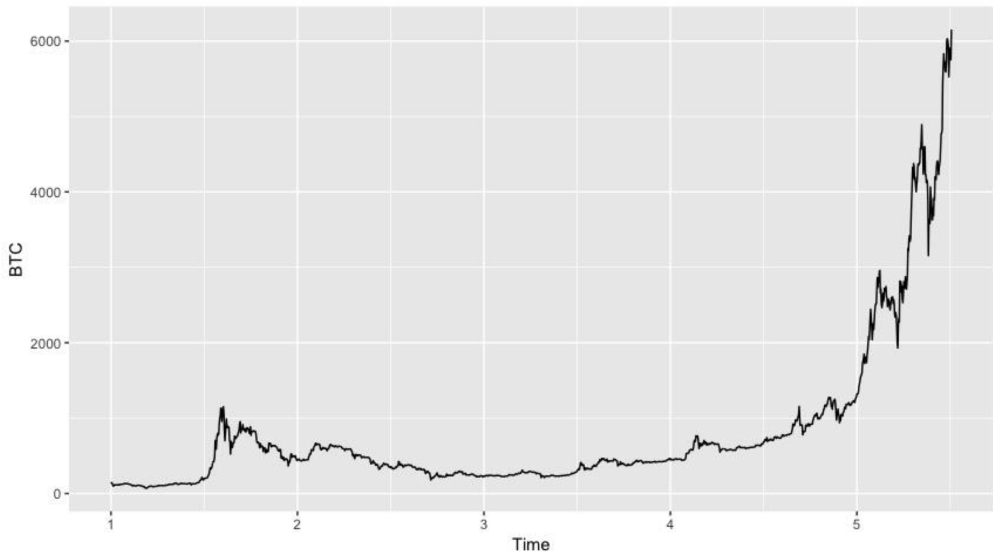

Figure 2: BTC price graph

Step 3. Check seasonality
```
> Test_Seasonal <- tbats(BTC)
> Seasonal <- !is.null(Test_Seasonal$seasonal)
> Seasonal
[1] FALSE
```

Step 4. Obtain ARIMA model
```
> Fit<-auto.arima(BTC,seasonal=FALSE,
stepwise=FALSE,approximation=FALSE,trace=TRUE)
> Fit
```

```
Series: BTC
ARIMA(4,2,1)
Coefficients:
           ar1      ar2     ar3      ar4      ma1
       -0.0391  -0.0138  0.0034  -0.0745  -0.9899
s.e.    0.0251   0.0251  0.0252   0.0251   0.0036
sigma^2 estimated as 3465:  log likelihood=-9032
AIC=18076.01   AICc=18076.06   BIC=18108.44
> tsdisplay(residuals(Fit),lag.max=32, main="(4,2,1) Model Residuals")
```

ARIMA(4,2,1) is obtained model by computing auto.arima() function. This model will be used as a Fit. Obtained ARIMA(4,2,1) Model result has been shown in Figure 3.

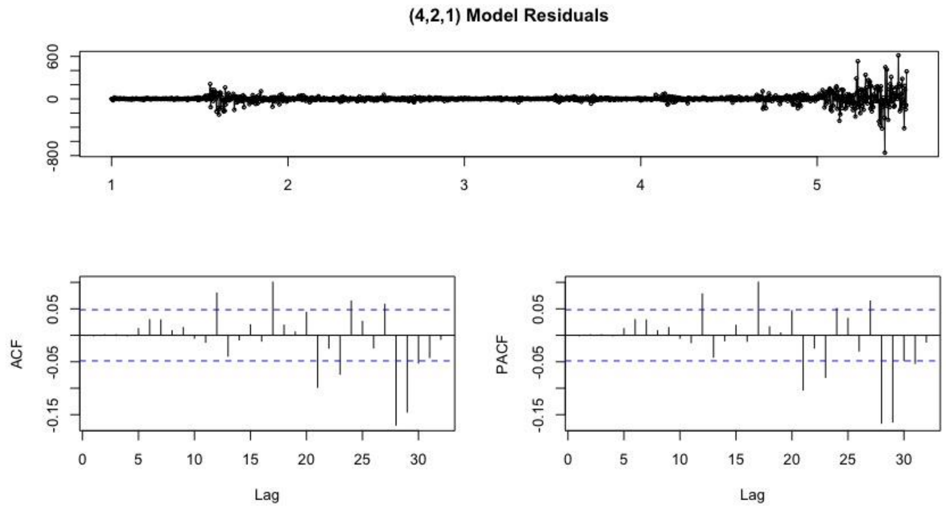

Figure 3: ARIMA(4,2,1) Model residuals, ACF and PACF graphics

Step 5. Create forecast model

```
> Forecast_30 <- forecast(Fit,h=30)
```

Forecast_30 includes BTC price for next 30 days (between October 30, 2017 and November 30, 2017). The minimum price is $6156.963 and maximum price is $6926.424. Prediction prices for next 30 days have been shown in Figure 4.

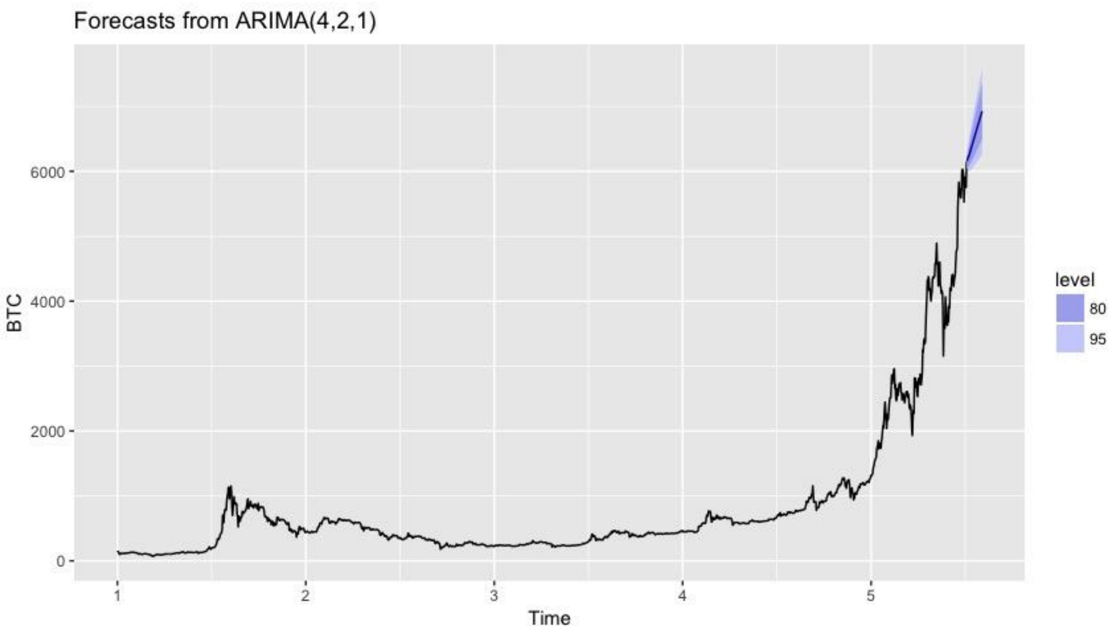

Figure 4: Prediction prices of next 30 days

Step 6. Create train data and test data

```
> Traindata <- data.frame(Forecast_30)
> Traindata <- Traindata$Point.Forecast
> Testdata <- c(6158.76, 6115.15, 6411.84, 6694.38, 7069.03, 7199.62, 7369.08, 7400.39, 7025.14, 7167.19, 7447.27, 7154.71, 6622.42, 6368.32, 5852.81, 6527.20, 6714.17, 7280.20, 7814.49, 7738.42, 7833.45, 8012.64, 8225.41, 8132.54, 8260.71, 8054.45, 8239.31, 8706.60, 9207.99, 9713.31)
```

Testdata includes real Bitcoin prices on between October 30, 2017 and November 30, 2017.

Step 7. Test accuracy results

```
> accuracy(Traindata,Testdata)
                ME     RMSE      MAE      MPE     MAPE
Test set 880.0015 1146.067 939.5819 10.86483 11.86484
```

R function accuracy shows some accuracy tests. It means ARIMA(4,2,1) model can predict prices by approximately 11.84% error.

LSTM Deep Learning Algorithm

Deep Learning (DL) is a part of machine learning (also known as deep structured learning or hierarchical learning). DL is based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers composed of multiple non-linear transformations. LSTM Deep Learning algorithm, developed by Hochreiter and Schmidhuber (1997), allows the preservation of the weights that are forward and back-propagated through layers. The network can continue to learn over many time steps by maintaining a more constant error. Thus, the network can be used to learn long-term dependencies. An LSTM cell contains the forget and remember gates which allow the cell to determine what information to prevent or pass based on its strength and importance. Adam optimizer is an excellent general-purpose optimizer that performs our gradient descent via backpropagation through time (Kingma, & Ba, 2014). The Rectified Linear Unit (ReLU) has become very popular in deep learning process. It computes the function *f(x)=max(0,x)* (Nair, & Hinton, 2010).

Daily prices have been used as input values in LSTM deep learning algorithm. ReLU as an activation function and Adam as an optimizer function have been chosen for the model. Obtained Process LSTM Model has been shown Figure 5.

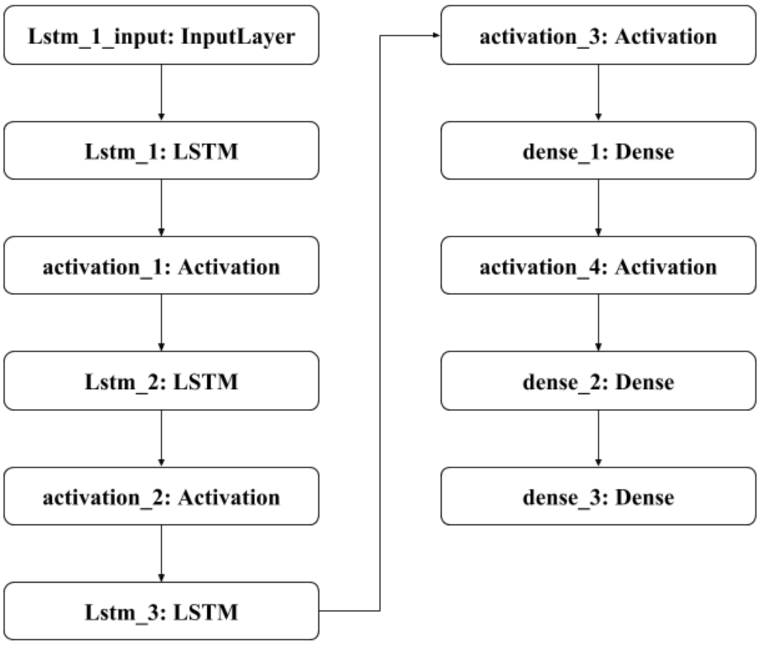

Figure 5: LSTM deep learning output model for prediction Bitcoin price

LSTMs are sensitive to the scale of the input data. The data has been rescaled to the range of 0 to 1, also called normalizing. The model has one output to predict price. 30 days data have been used to predict next days prices.

The model has 3 LSTM and 2 hidden layers between input and output layers. First LSTM layer has 64 neurons which are fully connected with input layers. Second and third LSTM layers have 128 neurons. ReLU activation function has been applied to all LSTM layers. A hidden layer which has 128 layers follow the third LSTM layer. Between the third LSTM layer and hidden layer, ReLU activation function has been applied. The second hidden layer has 64 neurons. Between the first and second MLP layers, ReLU activation function has been applied. Output layer follows the second MLP layer. There is no activation function between second MLP and output layers. Adam optimization function has been used in this study. Learning rate has been determined as “0.001”. All learning processes have been run in 1000 epochs. Obtained model has predicted Bitcoin price in next 30 days by 1.4% error.

Results

A few essential performance measures which are frequently used by researchers have been described below (Adhikari, & Agrawal, 2013).

MAE (The Mean Absolute Error) = $\frac{1}{n}\sum_{t=1}^{n}|e_t|$

MAPE (The Mean Absolute Percentage Error) = $\frac{1}{n}\sum_{t=1}^{n}|\frac{e_t}{y_t}|100$

MPE (The Mean Percentage Error) = $\frac{1}{n}\sum_{t=1}^{n}(\frac{e_t}{y_t})100$

RMSE (The Root Mean Squared) = $\sqrt{\frac{1}{n}\sum_{t=1}^{n}e_t^2}$

These measures have been used to compare the results for ARIMA Model and LSTM Model. Table 1 depicts all results for models. LSTM Model was successful for all result of the tests.

Table 1: Comparison of accuracy tests for obtained ARIMA model and LSTM deep learning model.

Model	RMSE	MAE	MPE	MAPE
ARIMA(4,2,1) Model	1146.07	939.58	10.86	11.86
LSTM Deep Learning Model	93.27	81.56	1.40	1.40

Conclusion

Bitcoin prices data is a time series. Forecasting the future price of Bitcoin is very important for investors. ARIMA model as a statistical model and LSTM deep learning algorithm as a machine learning have been used for forecasting Bitcoin prices data. The results have been obtained using proposed ARIMA model process with R and using proposed LSTM deep learning model process with Python. LSTM Model computed predict price by 1.4% error. The results show machine learning techniques are convenient for future prediction price of Bitcoin.

References

1. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
2. Gallo, C., (2005). Artificial Neural Networks in Finance Modelling, *EconWPA*.
3. Bodyanskiy, Y., Popov, S. (2006). Neural network approach to forecasting of quasiperiodic financial time series, *European Journal of Operational Research 175 (2006)*, pp.1357–1366.
4. Navon, A., Keller, Y. (2017). Financial Time Series Prediction using Deep Learning, *Electrical Engineering and Systems Science*, arXiv:1711.04174.
5. Xiong, R., Nichols, E. P., Shen, Y. (2015). Deep learning stock volatility with Google domestic trends, arXiv:1512.04916.
6. Fischera, T., Kraussb, C. (2017). Deep learning with long short-term memory networks for financial market predictions, *FAU Discussion Papers in Economics*, No. 11/2017.

7. Madan, I., Saluja, S., Zhao, A. (2014). Automated Bitcoin Trading via Machine Learning Algorithms, *Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep*.
8. Georgoula, I., Pournarakis, D., Bilanakos, C., Sotiropoulos, D. N., Giaglis, G. M. (2015). Using Time-Series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices, *MCIS 2015 Proceedings*.
9. Jang, H., Lee, J. (2018). An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian *Neural Networks Based on Blockchain Information*, in IEEE Access, vol. 6, pp. 5427-5437.
10. Amjad, M. J., Shah, D. (2016). Trading Bitcoin and Online Time Series Prediction, *Proceedings of the Time Series Workshop at NIPS 2016*, PMLR 55:1-15.
11. Abu Bakar N., Rosbi, S. (2017). 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 (IJAERS)*, Vol:4, Issue:11.
12. Coghlan, A. (2017). A Little Book of R For Time Series, *Parasite Genomics Group*, Wellcome Trust Sanger Institute, Cambridge, U.K.
13. Chollet, F. (2015). Keras, *https://github.com/fchollet/keras*.
14. Box, G. E. P., Jenkins, G. (1970). Time Series Analysis, Forecasting and Control, *Holden-Day*, San Francisco, CA.
15. Hyndman, R. J., Athanasopoulos, G. (2012). Forecasting: principles and practice. *Otexts, http://otexts.com/fpp*.
16. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory, *Neural Comput.* Nov 15; 9(8):1735-80.
17. Kingma, D. P., Ba, J. (2014). Adam: A Method for Stochastic Optimization, arXiv:1412.6980.
18. Nair, V., Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines, *In Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML 10)*, Johannes Fürnkranz and Thorsten Joachims (Eds.). Omnipress, USA, 807-814.
19. Adhikari, R., Agrawal, R. K. (2013). An Introductory Study on Time Series Modeling and Forecasting, *LAP Lambert Academic Publishing*.

Brief biographies of the authors

Ebru Şeyma Karakoyun

She is a master student at Industrial Engineering Programme in Necmettin Erbakan University. She graduated Department of Mathematics in Mugla Sitki Kocman University in 2015. She works about machine learning, financial models.

Ali Osman Cibikdiken

He works at Department of Computer Engineering, Necmettin Erbakan University as an assistant professor. His interests are Machine Learning, Deep Learning, Cloud Computing, Scientific Computing and Blockchain.