NONSTATIONARY TIME SERIES MODELS ON CRYPTOCURRENCIES

Shou Hsing Shih

American University of Sharjah, Department of Mathematics and Statistics Sharjah, PO Box 26666, UAE

*Corresponding Author Email: sshih@aus.edu

This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ARTICLE DETAILS

Article History:

Received 12 March 2018

Accepted 12 April 2018

Available online 15 May 2018

ABSTRACT

Cryptocurrencies are known as unpredictable due to their highly volatility. In time series, the forecasting accuracy is strongly affected by the methodologies that are used in identifying the pattern of a nonstationary stochastic realization. The purpose of the present study is to develop an algorithm that is capable of efficiently identifying the pattern of cryptocurrencies. A brief summary of the algorithm is given. To illustrate the quality of our proposed algorithm, we study the pattern of ten different reputable cryptocurrencies and use their daily closing prices to constitute a time series. The comparison between our proposed forecasting algorithm versus the autoregressive integrated moving average (ARIMA) process will be demonstrated.

KEYWORDS

Time Series Forecasting, Cryptocurrencies, Polynomial Trend Analysis, ARIMA .

1. INTRODUCTION

In time series analysis, it is known that one cannot proceed in building a time series model without conforming certain mathematical constraints such as stationarity of a given stochastic realization. Therefore, we shall first define several important mathematical concepts that are essential in developing the analytical process. In dealing with stationary data, let $\{x_t\}$ be a stationary time series, the Mixed Autoregressive- Moving Average process is defined as follows:

$$\phi_p(B)x_t = \theta_q(B)e_t \quad (1)$$

where

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (2)$$

and

$$\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q \quad (3)$$

Therefore, the $ARMA(p, q)$ process can be expressed as

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} \quad (4)$$

By letting $e_t = 0$, we can obtain the one-day ahead forecast as

$$\hat{x}_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} \quad (5)$$

In dealing with nonstationary data, let $\{x_t\}$ be a nonstationary time series, the Autoregressive Integrated Moving Average process is defined as follows

$$\phi_p(B)(1-B)^d x_t = \theta_q(B)e_t \quad (6)$$

where

$$(1-B)^d \quad (7)$$

is the difference filter, and

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (8)$$

and

$$\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q \quad (9)$$

Similar to the $ARMA(p, q)$ process, by letting $e_t = 0$, we can obtain the one-day ahead forecast $\hat{x}_t$ of the $ARIMA(p, d, q)$ by expanding (6).

2. PROPOSED METHODOLOGY

In reality, most of the time series are nonstationary in nature, and it is known that one cannot proceed the parameters estimations without confirming the stationarity of the given time series. Therefore, choosing a proper filter to reduce the nonstationary stochastic realization into a stationary one is extremely important. In the model building process of the $ARIMA$ model, we rely on the differencing filter to reduce the nonstationary time series into a stationary series. The accuracy of the model will be highly depending on the effect of the differencing filter. In other words, the larger the daily change values, the less accuracy of the final model. We believe the key to address this inaccuracy problem caused by the differencing filter is to reduce the variation of the stationary time series, and our proposed methodology is described as follows.

In order for us to properly illustrate our proposed algorithm, we chose ten well-known cryptocurrencies as part of our analysis: Bitconnect (BCC), Bitcoin (BTC), Dash (DASH), Ethereum (ETH), Litecoin (LTC), NEO (NEO), OmiseGO (OMG), Qtum (QTUM), Monero (XMR), and Ripple (XRP). Each data set consists of one year of its daily closing prices except OMG and QTUM, because both cryptocurrencies only start about half year ago.

For a brief description of Bitcoin, Dash, Ethereum, Litecoin, Monero, Ripple, see reference [1]. A brief description of Bitconnect, NEO, OmiseGo and Qtum are as follows. Bitconnect (BCC) claims to be an "open source all in one bitcoin and crypto community platform designed to provide multiple investment opportunities with cryptocurrency education and it has been accused running a Ponzi scheme [2]. Based on a study, NEO (NEO), also known as the Chinese Ethereum, is a blockchain project that utilizes blockchain technology and digital identity to digitize assets, to automate the management of digital assets using smart contracts, and to

realize a "smart economy" with a distributed network [3]. OmiseGO (OMG) is a public Ethereum-based financial technology for use in mainstream digital wallets, that enables real-time, peer-to-peer value exchange and payment services [4]. Qtum (QTUM) is "built upon Bitcoin's well - established UTXO transaction model and employs a Proof-of-Stake consensus mechanism that is more practical for business adoption [5]. Our proposed method begins with studying the cross correlations between each one of these cryptocurrencies and try to identify groups that is suitable for our study. The correlation between any two quantitative random variables can be found using

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{n \sum x^2 - (\sum x)^2} \sqrt{n \sum y^2 - (\sum y)^2}} \quad (10)$$

The reason why that cryptocurrencies are unpredictable is because they are highly volatile. It is quite common for them to go up or down 30% or more in any given day. To address that issue, we identify those cryptocurrencies that are highly correlated with each other, and we group them together. Instead of forecasting the price of one particular cryptocurrency, we will forecast the quotient of the two highly correlated cryptocurrencies. The usefulness of this idea will really benefit people who are interested in coin to coin trading.

Let $\{y_t\}$ and $\{z_t\}$ be two highly correlated cryptocurrencies. Let $\{x_t\}$ be the quotient of those two time series, and $x = 1, 2, 3, \dots, n$, where n is the sample size. We pair the data points of $\{x_t\}$ and x by letting $\{x_t\}$ be the y-axis and x be the x-axis. From the inspiration of Calculus, we know that a polynomial of degree n can have at most $n - 1$ relative extrema. Consider the following polynomial

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{15}x^{15} \quad (11)$$

By using the least square method, we can obtain a polynomial that best fits the original time series. Let

Table 1: Cross Correlation between Ten Well Known Cryptocurrencies

	BCC	BTC	DASH	ETH	LTC	NEO	OMG	QTUM	XMR	XRP
BCC	1	0.9286502	0.8719982	0.7614801	0.8088511	0.7008293	0.5570016	0.5958152	0.8463914	0.6069012
BTC	0.9286502	1	0.9580976	0.8701001	0.9387128	0.8112679	0.716489	0.7585888	0.9573027	0.7171312
DASH	0.8719982	0.9580976	1	0.9082952	0.9596602	0.8458554	0.8160922	0.8874854	0.9845499	0.7961055
ETH	0.7614801	0.8701001	0.9082952	1	0.8952923	0.9348826	0.8982666	0.8752877	0.9290543	0.8467616
LTC	0.8088511	0.9387128	0.9596602	0.8952923	1	0.8171361	0.7812025	0.8730573	0.9606459	0.7925894
NEO	0.7008293	0.8112679	0.8458554	0.9348826	0.8171361	1	0.873486	0.7990643	0.8840205	0.7997126
OMG	0.5570016	0.716489	0.8160922	0.8982666	0.7812025	0.873486	1	0.8440525	0.86531	0.8113256
QTUM	0.5958152	0.7585888	0.8874854	0.8752877	0.8730573	0.7990643	0.8440525	1	0.8948322	0.9105413
XMR	0.8463914	0.9573027	0.9845499	0.9290543	0.9606459	0.8840205	0.86531	0.8948322	1	0.8292812
XRP	0.6069012	0.7171312	0.7961055	0.8467616	0.7925894	0.7997126	0.8113256	0.9105413	0.8292812	1

The highlighted ones show strong correlations among those cryptocurrencies.

In order for us to properly illustrate our proposed algorithm, we chose five pairs of highly correlated cryptocurrencies from the above table, namely the ETH & NEO, BTC & DASH, DASH & XMR, QTUM & XRP, and XMR & LTC. Each data set consists of one year of the quotient of their daily closing prices, except the pair QTUM & XRP because the availability of historical prices of QTUM is only about half year. We shall model each of this nonstationary series using the classical *ARIMA* process and our proposed algorithm, and compare the results of each case.

From Figure 1 to 6, we have only illustrated the (historical, quotient, classical approach, trend function and our proposed method) line graphs for one (ETH & NEO) of the five pairs we discussed above. As all the other four pairs illustrates similar results, we decided it was not needed to reproduce all the line graphs for those pairs too.

Figure 1 and Figure 2 provides the line graph of the historical prices of Ethereum and NEO, and it is obvious to see that both of them follows the same pattern except at the earlier stage.

$$y_t = f(t) \quad (12)$$

and

$$z_t = x_t - y_t \quad (13)$$

After removing the trend from the original nonstationary time series, the series $\{z_t\}$ is now stationary and has a smaller variance than $(1 - B)x_t$. We then use the *ARMA* process to find the best model that fits the series $\{z_t\}$. Let $\hat{z}_t$ be the one-day ahead forecast of z_t . The one-day ahead forecast of x_t can be expressed by using

$$\hat{x}_t = f(t) + \hat{z}_t \quad (14)$$

To illustrate the usefulness and accuracy of our proposed algorithm, the comparison between our proposed methodology and the classical *ARIMA* approach will be demonstrated with real data in the later section.

2.1 Model Selection and Parameters Estimations

In both *ARMA* and *ARIMA* processes that we discussed in this paper, we selected the final model according to Akaike's information criterion (AIC) [6]. To be specific, after reducing the original time series into a stationary series, let $m \leq 5$ be the order of the autoregressive operator, and $n \leq 5$ be the order of the moving average operator. We have total of 35 models in each case, and we shall select the model with the smallest AIC. For the parameters estimations, it is referenced in [7-12].

2.2 Comparison

The following table shows the cross correlations between each cryptocurrency and other cryptocurrencies.

Figure 1: Historical Price of Ethereum

Figure 2: Historical Price of NEO

Let $\{h_t\}$ and $\{k_t\}$ be the historical prices of Ethereum and NEO, respectively. Let $x_t = \frac{h_t}{k_t}$.

Figure 3 represents the line graph on the series of $\{x_t\}$. It can be observed that the quotient series has stabilized somewhere before the middle. It indicates that if we only consider the stable section, the correlation coefficient between the two series will improve significantly.

Figure 3: The Quotient between ETH & NEO

Figure 4: The Original VS Forecasts on the Quotient between ETH & NEO using Classical Approach

Figure 4 provides the comparison between the original and forecasts using the classical ARIMA approach. The forecasts are pretty close to the actual quotient series, but the lag problem has always been the issue for the classical ARIMA approach.

For our proposed method, we begin with estimating the parameters of the trend function for the quotient series using the least square method that we discussed earlier, and the trend function is shown as Figure 5. It can be observed that our proposed trend model did a nice job on picking up the trend of the quotient series.

Table 2: Efficiency Comparison for ETH & NEO

Original	Classical	Proposed	Res01	Res02	
11.2183	10.43361	11.26263	0.784694	-0.044326	*
10.23901	12.98767	11.01917	-2.74866	-0.780155	*
10.21075	10.7773	10.19257	-0.566549	0.018175	*
10.66417	9.438509	10.20539	1.225665	0.458783	*
10.22897	9.694029	10.59582	0.534942	-0.366853	*
9.930451	9.605235	9.963851	0.325216	-0.0334	*
8.020009	12.15868	9.831355	-4.138671	-1.811346	*
6.893549	7.847537	7.970326	-0.953988	-1.076777	
7.251824	7.522777	7.435661	-0.270953	-0.183837	*
7.174436	4.826113	7.684035	2.348323	-0.509599	*
7.263984	7.638128	7.365049	-0.374144	-0.101065	*
7.448746	7.772584	7.178478	-0.323838	0.270268	*

Figure 5: The Trend Function of the Quotient between ETH & NEO

Figure 6: The Original VS Forecasts on the Quotient between ETH & NEO Using Our Proposed Method

The above Figure 6 demonstrates that our proposed algorithm not only address the lag problem, but also outperform the classical ARIMA approach on the forecasting the quotient series.

2.3 Evaluation of the Proposed Models

To further evaluate the efficiency of our proposed models, we decided to hide the last 12 observations for these five applications and re-predict the last 12 forecasts only using the previous information. We defined the residuals as follows

$$residuals = actual - forecasts \quad (15)$$

where the Res01 represents the residuals of the classical approach, and Res02 represents the residuals of our proposed models. The sign “*” is an indication that our proposed model produced a smaller absolute residual on that particular forecast.

The following Tables gives the efficiency comparison of the five pairs of cryptocurrencies in this study between the classical approach and our proposed model. From Table 2, it shows that 11 out of 12 of our proposed forecasts are closer to the actual quotient series than the classical ARIMA approach.

Table 3 illustrates the efficiency comparison between DASH and Monero, and our proposed model perform better than the classical ARIMA model 9 out of 12 times.

Table 3: Efficiency Comparison for DASH & XMR

Original	Classical	Proposed	Res01	Res02	
2.685338	2.803399	2.816687	-0.118061	-0.131349	
2.690006	2.686215	2.704192	0.003791	-0.014186	
2.713396	2.68656	2.695044	0.026836	0.018352	*
2.834388	2.707108	2.684293	0.12728	0.150095	
2.672045	2.812011	2.754345	-0.139966	-0.0823	*
2.619355	2.698034	2.648103	-0.078679	-0.028748	*
2.538221	2.655617	2.617726	-0.117396	-0.079505	*
2.265456	2.548096	2.532719	-0.28264	-0.267263	*
2.373516	2.293427	2.345406	0.080089	0.02811	*
2.486199	2.357279	2.397359	0.12892	0.08884	*
2.640029	2.438665	2.44279	0.201364	0.197239	*
2.320663	2.605608	2.577387	-0.284945	-0.256724	*

Table 4 illustrates the efficiency comparison between Monero and Litecoin, and our proposed model perform better than the classical ARIMA model 6 out of 12 times.

Table 4: Efficiency Comparison for XMR & LTC

Original	Classical	Proposed	Res01	Res02	
1.583789	1.587138	1.566111	-0.003349	0.017678	
1.67313	1.595694	1.566952	0.077436	0.106178	
1.627575	1.662436	1.650292	-0.034861	-0.022717	*
1.55029	1.621631	1.626223	-0.071341	-0.075933	
1.666723	1.572708	1.593693	0.094015	0.07303	*
1.620626	1.652644	1.680533	-0.032018	-0.059907	
1.673873	1.606232	1.645483	0.067641	0.02839	*
1.781028	1.69326	1.706284	0.087768	0.074744	*
1.699358	1.769312	1.77661	-0.069954	-0.077252	
1.711291	1.690022	1.719013	0.021269	-0.007722	*
1.641101	1.73095	1.726902	-0.089849	-0.085801	*
1.889614	1.642027	1.63889	0.247587	0.250724	

Table 5 illustrates the efficiency comparison between DASH and Bitcoin, and our proposed model perform better than the classical ARIMA model 6 out of 12 times.

Table 5: Efficiency Comparison for DASH & BTC

Original	Classical	Proposed	Res01	Res02	
0.071704	0.078581	0.077564	-0.00687697	-0.00585996	*
0.076179	0.071584	0.072654	0.00459475	0.00352454	*
0.073889	0.076552	0.078295	-0.00266288	-0.00440565	
0.075182	0.07402	0.075318	0.00116254	-0.00013532	*
0.075452	0.075376	0.075671	7.614E-05	-0.00021824	
0.076619	0.075139	0.076361	0.00148018	0.00025731	*
0.073633	0.07671	0.076989	-0.00307746	-0.00335594	
0.06796	0.073139	0.075812	-0.00517946	-0.00785254	
0.066109	0.067828	0.068628	-0.00171964	-0.00251909	
0.070927	0.066064	0.067178	0.00486286	0.00374825	*
0.07281	0.070862	0.071336	0.00194821	0.00147386	*
0.072898	0.07378	0.071626	-0.00088127	0.00127238	

Table 6 illustrates the efficiency comparison between QTUM and Ripple, and our proposed model perform better than the classical ARIMA model 8 out of 12 times. It evidences that our proposed model is capable of forecasting the data in the future.

Table 6: Efficiency Comparison for QTUM & XRP

Original	Classical	Proposed	Res01	Res02	
0.043944	0.038244	0.043492	0.00570043	0.00045204	*
0.039466	0.04407	0.043239	-0.00460339	-0.00377235	*
0.036428	0.039483	0.041052	-0.00305553	-0.00462486	
0.042623	0.036635	0.040449	0.00598844	0.00217393	*
0.040468	0.042312	0.042219	-0.00184402	-0.00175121	*
0.036929	0.040423	0.039599	-0.00349448	-0.0026698	*
0.034393	0.037215	0.036075	-0.00282131	-0.00168128	*
0.031951	0.034836	0.0349	-0.00288484	-0.00294848	
0.03275	0.032502	0.03496	0.00024896	-0.00220927	
0.03592	0.03316	0.033612	0.00276005	0.0023082	*
0.042815	0.035997	0.035303	0.00681848	0.00751164	
0.039551	0.042226	0.039388	-0.00267538	0.00016316	*

3. CONCLUSION

In the present study, we have introduced the idea of forecasting the quotient series of two highly correlated cryptocurrencies; it addresses the highly volatile issue of cryptocurrencies and make them possible to be predicted. THE Algorithm that we developed is capable of forecasting the quotient series of all five pairs of highly correlated cryptocurrencies in the study. The methods of the parameters estimations and model selection are discussed in the study. The detailed comparisons and evaluations have shown that our proposed methodology is more efficient, stable and accurate than the classical approach.

REFERENCES

- [1] Chan, Stephen, Chu, J., Nadarajah, S., Osterrieder, J. 2017. A statistical analysis of cryptocurrencies. *Journal of Risk Financial Management* 10: 12. doi:10.3390/jrfm10020012.
- [2] Bitconnect. 2018. Bitconnect main website. Available online: <https://bitconnect.co/> (Accessed on 20 February 2018).
- [3] NEO 2018. NEO main website. Available online: <https://neo.org/> (Accessed on 20 February 2018).
- [4] OmiseGo 2018. OmiseGo main website. Available online: <https://omisego.network/> (Accessed on 20 February 2018).
- [5] Qtum 2018. Qtum main website. Available online: <https://qtum.org/en/> (Accessed on 20 February 2018).
- [6] Akaike, H. 1974. A New Look at the Statistical Model Identification, *IEEE Transactions on Automatic Control*, AC-19, 716-723.
- [7] Brockwell P.J., Davis, R.A. 1996. *Introduction to Time Series and Forecasting*. Springer, New York. Sections 3.3 and 8.3.
- [8] Durbin, J., Koopman, S.J. 2001. *Time Series Analysis by State Space Methods*. Oxford University Press.
- [9] Gardner, G., Harvey, A.C., Phillips, G.D.A. 1980. Algorithm AS154. An algorithm for exact maximum likelihood estimation of autoregressive-moving average models by means of Kalman filtering. *Applied Statistics*, 29, 311-322.
- [10] Harvey, C. 1993. *Time Series Models*, 2nd Edition, Harvester Wheatsheaf, sections 3.3 and 4.4.
- [11] Jones, R.H. 1980. Maximum likelihood fitting of ARMA models to time series with missing observations. *Technometrics*, 20, 389-395.
- [12] Ripley, B.D. 2002. Time series in R 1.5.0. *R News*, 2/2, 2-7. http://www.r-project.org/doc/Rnews/Rnews_2002-2.pdf