Journal of Applied Finance & Banking, Vol. 12, No. 5, 2022, 25-43
ISSN: 1792-6580 (print version), 1792-6599(online)
https://doi.org/10.47260/jafb/1252
Scientific Press International Limited

Predicting Bitcoin Prices via Machine Learning and Time Series Models

Yu-Min Lian[1,*], Jia-Ling Chen[2] and Hsueh-Chien Cheng[3]

Abstract

In this study, we predict Bitcoin price trends using the back propagation neural network (BPNN), autoregressive integrated moving average (ARIMA), and generalized autoregressive conditional heteroscedasticity (GARCH) models. Based on principal component analysis (PCA), we extract two new input components for BPNN from Bitcoin's three-day closing prices, MA5, MA20, daily trading volume, Ether price, and Ripple price. The training set covers the period between September 1, 2015 and March 31, 2020, and the forecasting set covers the period between April 1, 2020 and June 30, 2020. Empirical results reveal (1) the predictive ability of BPNN over that of the ARIMA models; (2) BPNN with two hidden layers is able to predict price trends more precisely than that with only one hidden layer; (3) in terms of time series models, the ARIMA-GARCH family of models demonstrates better predictive performance than ARIMA models; and (4) among the ARIMA-GARCH family of models, the ARIMA-EGARCH model is proven to produce the best predictive results on price, and the ARIMA-GARCH model predicts more accurately than the ARIMA-GJR-GARCH model. Specifically, our findings provide a reference on Bitcoin for market participants.

JEL classification numbers: C32, C45, C53, G17.
Keywords: Bitcoin, Back propagation neural network, Autoregressive integrated moving average, Generalized autoregressive conditional heteroscedasticity, Principal component analysis.

---

[1] Assistant Professor, Department of Business Administration, Fu Jen Catholic University, New Taipei City, Taiwan. [*] Corresponding author.
[2] Department of Business Administration, Fu Jen Catholic University, New Taipei City, Taiwan.
[3] Department of Business Administration, Fu Jen Catholic University, New Taipei City, Taiwan.

Article Info: *Received:* May 3, 2022. *Revised:* June 1, 2022.
Published online: June 6, 2022.

1. Introduction

Bitcoin, proposed by Satoshi Nakamoto in 2008, was the first cryptocurrency. Its peer-to-peer structure enables cryptocurrency transactions in a decentralized rather than centralized structure. However, Bitcoin transactions often require hours for confirmation, and Bitcoin is mainly acquired through costly "mining." Cryptocurrency value also fluctuates more than fiat currency. Accurate price forecasts enable investors to formulate better strategies.

Time series models have always been used for price forecasting. Ohyver and Pudjihastuti (2018) formulated autoregressive integrated moving average (ARIMA) models to forecast the price of rice from January 2015 to April 2017 and obtained the optimal ARIMA(1,1,2) model. Empirical analyses suggested that ARIMA models were better in the short term, so the data require frequent updating. The government can reference historical data price to make better decisions. Wood and Dasgupta (1996) used an ARIMA model, a neural network (NN), and a regression model to predict trends in the MSCI U.S.A. Capital Market Index; their empirical analysis indicated that the ARIMA model had the best predictive accuracy. To capture price volatility, the GARCH family of models is often employed. Qiao et al. (2020) forecast the VIX index using the GARCH, GJR-GARCH, E-GARCH, and DJI-GARCH models and found that GARCH(1,1) and GJR-GARCH are relatively better for out-of-sample testing. Lama et al. (2015) applied the ARIMA, GARCH, and EGARCH models to forecast an international cotton price series and found that the EGARCH model provided more precise predictions.

As artificial intelligence grow, machine learning methods such as NN and back propagation neural networks (BPNN) have become more popular. Liu and Ma (2012) demonstrated that BPNN was able to forecast the Shanghai stock index, and that BPNN with one hidden layer could deal with complex continuous questions. Grudnitski and Osburn (1993) used BPNN to predict monthly price changes in gold futures and the S&P 500 index; the accuracy rate for predicting gold futures was 61% and for the S&P 500 was 75%. To predict Canadian stock returns, Olson and Mossman (2003) classified 2,353 Canadian companies' 61 accounting ratios into four to six categories as input data and compared BPNN forecasts with ordinary least squares and logistic regression (logit) techniques; BPNN outperformed traditional regression-based forecasting in non-linear situations.

In recent years, Bitcoin has attracted investors' attention. Lian et al. (2019) proposed a Monte Carlo simulation and spot-futures with the cost of carry to build a dynamic price model for Bitcoin futures. According to the results, various factors had a significant impact on the Bitcoin futures price. Katsiampa (2017) applied various GARCH models to analyze Bitcoin volatility and showed that AR-CGARCH was the best model. Since Bitcoin differs from other financial assets, it is possible to create benefits for stakeholders using portfolio analysis and risk management. Chen et al. (2019) employed BPNN and ARIMA models to forecast the Bitcoin price from 2014 to 2018 and demonstrated that BPNN exhibited better predictive performance than ARIMA models. ARIMA models exhibited smaller deviations when price volatility was relatively stable. Therefore, the study suggested that investors should observe the characteristics of recent Bitcoin historical data when establishing a forecasting model. Furthermore, Corbet et al. (2020a, 2020b) provided the feasible works so that the empirical results are achieved. Lian and Chen (2021) empirically investigated the properties of cryptocurrency returns and priced the European-style cryptocurrency options.

Most previous studies believe that time series models have good price predictive ability considering different targets, and BPNN explains non-linear data more effectively. Therefore, due to Bitcoin's high price volatility, this study uses BPNN and time series models to determine the most precise Bitcoin price prediction trend. This study makes three main contributions. First, we depict the Bitcoin price dynamics through the BPNN and the ARIMA-GARCH family of models to understand the operation of the cryptocurrency market and the risks involved. Our findings are valuable for the investment of other virtual currency-linked products for which the process of Bitcoin prices are expected to follow the proposed models. Second, we employ the principal component analysis (PCA) to extract two new input components for BPNN and maintain the nature of statistical significance. Finally, we evaluate the predictive performance of the BPNN and the ARIMA-GARCH family of models, compared with the actual Bitcoin price data to demonstrate the best predictive model. The empirical results are significant for investors and for the organization of the Bitcoin market.

The remainder of this study is structured as follows. Section 2 introduces the prediction models and demonstrates the parameter setting process. Section 3 presents the empirical analysis. Section 4 concludes.

2. Methodology

2.1 Back propagation neural network (BPNN)

2.1.1 BPNN

BPNN, a type of NN, was first released by McClelland and Rumelhart in 1986. The BPNN training process is divided into two stages: learning and recalling. In the learning phase, BPNN is a supervised algorithm trained using known data. When the initial system output differs from the desired output, the error difference is back propagated into the network using the gradient steepest descent method. BPNN then constantly adjusts the weights to minimize the loss function. During the recalling phase, BPNN recalls an optimal pattern from the learning phase and generates output. BPNN is suitable for prediction, diagnosis, and classification, and it defined as follows:

$$Y_j = (net) = f\left(\sum_{i=1}^{n} w_{ij} X_i - \theta_j\right), \tag{1}$$

where Y_j represents the output layer neurons and net represents the summation function. X_i is the input variable, W_{ij} is the weight, and θ_j is the bias, or threshold of hidden layer neurons. f represents the transfer or activation function.

2.1.2 Transfer function

The transfer function introduces non-linear characteristics into BPNN. Thus, the sigmoid function is used to convey the output value between 0 and 1. Its equation is expressed as follows:

$$(net) = \frac{1}{1+exp^{-net}}, \tag{2}$$

2.1.3 The dimensionality reduction of BPNN

Principal component analysis (PCA), which simplifies several relative variables into a few independent, principal components by computing the weighted average of each variable, is represented by the following formula:

$$y_j = a_{j1}x_1 + a_{j2}x_2 + \cdots + a_{jp}x_p, \tag{3}$$

where y_j represents the j^{th} principal component, x_p represents the original variable, and a_j is the covariance matrix, which represents the λ_j eigenvector corresponding to the j^{th} eigenvalue.

2.2 ARIMA

2.2.1 Introduction of ARIMA

The ARIMA(*p*,*d*,*q*) model, introduced by Box and Jenkins (1976), combines the AR(autoregressive) model, I(integrated), and MA(moving average) models. The ARIMA(*p*,*d*,*q*) model can be written as Equations (4)–(6):

$$y_t = a_0 + \sum_{i=1}^{p} a_i \; y_{t-i} + \varepsilon_t, \tag{4}$$

$$\Delta X_t = X_t - X_{t-1}, \tag{5}$$

$$y_t = b_0 + \sum_{i=1}^{q} b_i \; \varepsilon_{t-i} + \varepsilon_t, \tag{6}$$

where a_0 is a constant, p is the order of lagged value, a_i are the actual values, ε_t is the random error at time *t*, $X_t$is the first-order difference, $\Delta^d X_t$ is the d-order difference, b_0 is a constant, q is the order of lagged value, b_i is the coefficient of ε_{t-i}, and ε_t is the random error at *t*.

2.2.2 Establish the ARIMA model

The first step in establishing the model is the unit root test. This study adapts the Augmented Dickey-Fuller (ADF), Phillips-Perron (PP), and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests to examine stationarity. If the sequence is unsteady, we proceed to the second step to determine I(*d*). Then, the white noise test is performed on the sequence. In this process, we use the Ljung-Box test to test the white noise, the null hypothesis is that the test sequence is purely random, and the statistic for this test is Q.

The Ljung-Box test can be represented as follows:

$$Q(m) = n(n+2)\sum_{k=1}^{m}\frac{\rho_k^2}{n-k}\sim\chi_m^2, \quad (7)$$

where ρ_k^2 is the k-order autocorrelation coefficient of the sequence, n is the number of samples, and m is the set lagging order.

The last step is to determine the AR(p) and MA(q) model orders. We use the most common tools, autocorrelation function (ACF) and partial correlation function (PACF), to identify the order of p and q.

2.2.3 Model selection

We adopt the Akaike information criterion (AIC) and Bayesian information criterion (BIC) to find the optimal model. The AIC was introduced by Akaike (1974), and the BIC was developed by Gideon Schwarz (1978). The smaller their values, the closer the models' goodness-of-fit. Their expressions are provided in Equations (8) and (9) as follows:

$$\mathrm{AIC} = \mathrm{T}\ln(\mathrm{SSE}) + 2\mathrm{k}, \quad (8)$$

$$\mathrm{BIC} = \mathrm{T}\ln(\mathrm{SSE}) + \mathrm{k}\ln(\mathrm{T}), \quad (9)$$

where T is total number of samples, $\ln(SSE)$ is the natural logarithm of the residual sum of squares, k is total number of parameters to be estimated, and $\ln(\mathrm{T})$ is the natural logarithm of the total number of samples.

2.3 GARCH family models

2.3.1 GARCH model

To explain the volatility clustering of financial data, we use the GARCH model to augment the lagging periods of the residual sum of squares and conditional variance in the ARIMA model. According to Bollerslev, Chou, and Kroner (1992), the GARCH(1,1) model is sufficient for most economic data. Therefore, we chose GARCH(1,1) as the parameter value in our study. The GARCH model is defined as follows:

$$y_t = x_t + \varepsilon_t, \quad (10)$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q}\alpha_i\varepsilon_{t-i}^2 + \sum_{j=1}^{p}\beta_j\sigma_{t-j}^2, \quad (11)$$

$$y_t|\Omega_t \sim N(x_t\alpha, \sigma_t), \quad (12)$$

where x_t is the variable vector, α is the coefficient vector, ε_t is the residual value of t period, σ_t^2 is the residual variance, q is the residual lagging period, p is the residual variance lagging period, and if p = 0, the model is the same as ARCH(q).

2.3.2 GJR-GARCH model

The GJR-GARCH, or TGARCH, model is used to explain the effect of leverage in financial markets and the asymmetric phenomenon of conditional variance fluctuations (Glosten, Jaganathan, and Runkle, 1993). If it represents the rate of return of a financial asset at time *t*, it can be expressed as Equation (13):

$$y_t = a_0 + \varepsilon_t, \tag{13}$$

where ε_t is the residual, indicating that when $\varepsilon_{t-1} < 0$, the previous rate of return is lower than a_0, $\varepsilon_{t-1} = y_{t-1} - a_0 < 0$, so it is bad news for financial assets. Relatively, if the previous residual $\varepsilon_{t-1} \geq 0$, at least it is not bad news for financial assets. The GJR-GARCH model represents the previous ε_{t-1} of good and bad news, and the threshold is set to zero. The equations are as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 {\varepsilon_{t-1}}^2 + \gamma {\varepsilon_{t-1}}^2 D_{t-1} + \beta_1 {\sigma_{t-1}}^2, \tag{14}$$

$$D_{t-1} = \begin{cases} 1 \; if \; \varepsilon_{t-1} < 0 \\ 0 \; if \; \varepsilon_{t-1} \geq 0 \end{cases}, \tag{15}$$

If the estimated result > 0, the leverage effect exists. It also indicates that bad news in the previous period will make the conditional variation value of the current period greater than good news, which is consistent with the leverage effect.

2.3.3 EGARCH (exponential GARCH) model

The EGARCH model is similar to GJR-GARCH, but EGARCH takes the logarithm of the conditional variance and lagged terms in the model. It uses the absolute value of the standardized residual to maintain the conditional variance in the condition of positive variance (Nelson, 1991). The EGARCH(1,1,1) variance equation is defined as follows:

$$\ln(\sigma_t^2) = \alpha_0 + \alpha_1 \left|\frac{\varepsilon_{t-1}}{\sigma_{t-1}}\right| + \gamma \frac{\varepsilon_{t-1}}{\sigma_{t-1}} + \beta_1 \ln(\sigma_{t-1}^2), \tag{16}$$

When $\varepsilon_{t-1} < 0$, $\left|\frac{\varepsilon_{t-1}}{\sigma_{t-1}}\right|$ and $\frac{\varepsilon_{t-1}}{\sigma_{t-1}}$ are equal, the leverage effect exists, and $\gamma < 0$. In the equation, bad news increases the value of conditional variation in the current period.

2.4 Performance evaluation

We evaluate predictive ability using MSE, MAE, and MAPE. The lower the estimators of MSE, MAE, MAPE are, the better the model's accuracy.

The formulas are given by the following:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_t - \hat{y}_t)^2, \tag{17}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_t - \hat{y}_t|, \tag{18}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|, \tag{19}$$

where y_t represents actual value, $\hat{y}_t$ is predictive value, and n is the number of samples.

3. Empirical analytics

We use BPNN and time series models to predict the Bitcoin price trend and evaluate predictive performance using MSE, MAE, and MAPE. The training set covers September 1, 2015–March 31, 2020, and the forecasting set covers April 1, 2020–June 30, 2020. The data for BPNN and the ARIMA-GARCH family of models are from Yahoo! Finance.com.

3.1 BPNN

3.1.1 Variable setting

Inputting variables correlated to the desired output trains BPNN to learn autonomously. Table 1 presents and describes the eight variables selected for BPNN.

Table 1: Variable setting for BPNN

Variable (X_k)		**Description**
X_1~X_3	$P_{t-1}, P_{t-2}, P_{t-3}$	Past three-day closing prices
X_4~X_5	ETH, XRP	Ether and Ripple prices
X_6~X_7	MA_5, MA_{20}	5 and 20 days moving averages
X_8	Vol	Daily trading volume

3.1.2 PCA

According to Table 2, the first and second principal components explain 94.906% of the original variables. The empirical results indicate that first to second principal components should be selected and their eigenvalues should be greater than 1. Thus, we adopt the first and second principal components as new variables of BPNN.

Table 2: Description of variance ratios

	Eigenvalue	**Percentage of variance**	**Cumulative percentage of variance**
Principal component 1	6.521	81.512	81.512%
Principal component 2	1.072	13.394	94.906%
Principal component 3	0.264	3.302	98.208%
Principal component 4	0.116	1.446	99.654%
Principal component 5	0.020	0.246	99.900%
Principal component 6	0.006	0.070	99.970%
Principal component 7	0.002	0.026	99.996%
Principal component 8	0.000	0.004	100.000%

Table 3: The PCA model

	Mathematical equation
The first principal component	$y_1 = 0.165x_1 + 0.164x_2 + 0.162x_3 - 0.255x_4 - 0.264x_5 + 0.164x_6 + 0.147x_7 + 0.524x_8$
The second principal component	$y_2 = 0.048x_1 + 0.05x_2 + 0.051x_3 + 0.434x_4 + 0.436x_5 + 0.05x_6 + 0.067x_7 - 0.394x_8$

3.1.3 The structure of BPNN

The optimal parameters and output values for BPNN are conducted by 2,000 times of training; the five learning rates are 0.1, 0.3, 0.5, 0.7, and 0.9; and the momentum terms are 0.5, 0.7, and 0.9. The sigmoid function is employed as the transfer function in this process. According to the rule of thumb, one to two hidden layers have the best convergence properties. Therefore, we set the number of hidden layers to one and two with the trial-and-error method, with 1 to 14 neurons in one hidden layer. Regarding the two hidden layers, 1 to 14 neurons are set in the first layer, and 2 to 7 neurons are set in the second layer. Table 4 demonstrates the number of hidden layers and neurons used in this study.

Table 4: Neuron setting for BPNN

	One hidden layer	**Two hidden layers**
The number of *input layer neurons*	2	2
The number of 1^{st} hidden layer neurons	1~14	1~14
The number of 2^{nd} hidden layer neurons	-	2~7
The number of *output layer neurons*	1	1

3.1.4 Evaluation of prediction ability

MSE, MAE, and MAPE are used to assess prediction error between the actual Bitcoin price and the price forecast by the models. Tables 5–7 show the top three BPNNs with one hidden layer, based on the three performance assessments; the optimal BPNN structures occur with the learning rate set to 0.9 and with 4 to 5 hidden layer neurons. Tables 8–10 show the top three BPNNs with two hidden layers. Based on the three evaluation indicators, the optimal network structures are those with, respectively, 4 and 3 neurons at the first and the second hidden layer, a 0.1 learning rate, and a 0.7 momentum term. BPNN with two hidden layers has a slightly better predictive performance. Regardless of the number of hidden layers, 4 neurons tend to be selected for the first hidden layer.

Table 5: The top three BPNNs with one hidden layer based on MSE

MSE ranking	**The number of hidden layer**	**The structure of BPNN**	**Learning rate**	**Momentum term**	**MSE**
1	1	02-04-01	0.9	0.9	443493.359
2	1	02-04-01	0.7	0.9	445035.193
3	1	02-04-01	0.3	0.9	447239.754

Table 6: The top three BPNNs with one hidden layer based on MAE

MAE ranking	**The number of hidden layer**	**The structure of BPNN**	**Learning rate**	**Momentum term**	**MAE**
1	1	02-05-01	0.3	0.9	513.648
2	1	02-04-01	0.3	0.7	515.066
3	1	02-03-01	0.5	0.5	515.439

Table 7: The top three BPNNs with one hidden layer based on MAPE

MAPE ranking	The number of hidden layer	The structure of BPNN	Learning rate	Momentum term	MAPE
1	1	02-04-01	0.9	0.9	593.672%
2	1	02-04-01	0.7	0.9	594.759%
3	1	02-04-01	0.3	0.9	596.138%

Table 8: The top three BPNNs with two hidden layers based on MSE

MSE ranking	The number of hidden layer	The structure of BPNN	Learning rate	Momentum term	MSE
1	2	02-04-03-01	0.1	0.7	431041.920
2	2	02-04-02-01	0.5	0.9	438732.042
3	2	02-04-02-01	0.3	0.9	446393.075

Table 9: The top three BPNNs with two hidden layers based on MAE

MAE ranking	The number of hidden layer	The structure of BPNN	Learning rate	Momentum term	MAE
1	2	02-04-03-01	0.1	0.7	511.581
2	2	02-04-02-01	0.5	0.9	520.859
3	2	02-02-04-01	0.1	0.5	521.119

Table 10: The top three BPNNs with two hidden layers based on MAPE

MAPE ranking	The number of hidden layer	The structure of BPNN	Learning rate	Momentum term	MAPE
1	2	02-04-03-01	0.1	0.7	589.767%
2	2	02-04-02-01	0.3	0.9	597.737%
3	2	02-04-02-01	0.5	0.9	598.780%

Figures 1–2 illustrate the prediction trends of BPNN with one and two hidden layers during the training period. We demonstrate the best prediction model for each hidden layer. The optimal network structure of BPNN with one hidden layer is (02-04-01) with a 0.9 learning rate and a 0.9 momentum term. The optimal network structure of BPNN with two hidden layers is (02-04-03-01) with a 0.1 learning rate and a 0.7 momentum term. In conclusion, the models' forecasting price fits well with the actual Bitcoin price with both one and two hidden layers.

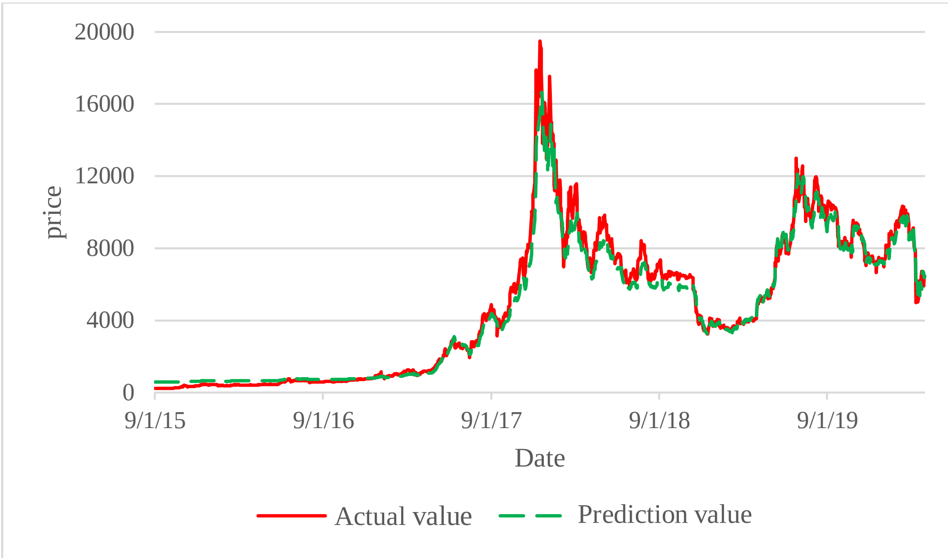


Figure 1: Prediction trend of the best BPNN with one hidden layer (2-4-1) under training

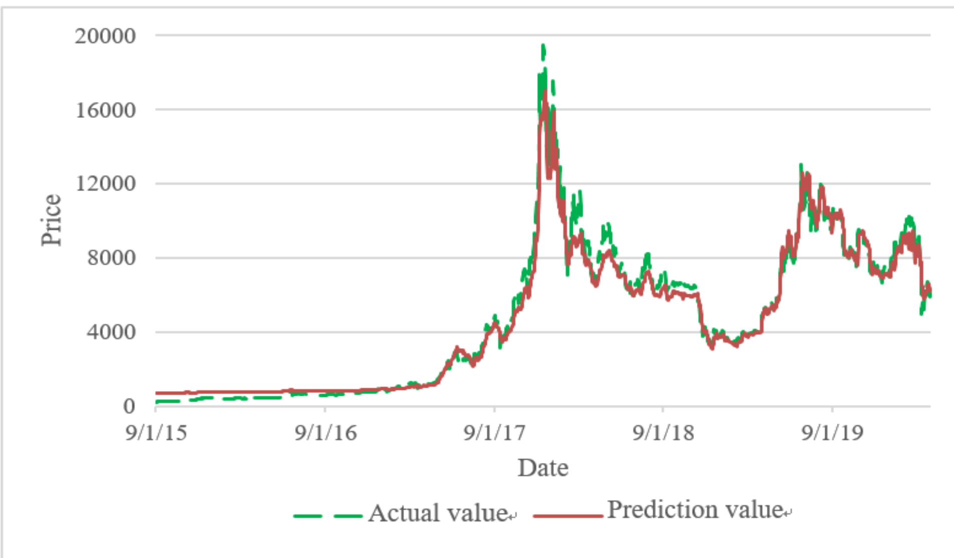


Figure 2: Prediction trend of the best BPNN with two hidden layers (2-4-3-1) under training

3.2 ARIMA model

3.2.1 Unit root test and white noise

We use ADF, PP, and KPSS to test whether the sequence is steady. Table 11 shows that the original Bitcoin sequence is unsteady. After the first-order difference, the sequence is steady as an I(1) sequence. Table 12 illustrates that the I(1) sequence is tested by white noise, and its *p*-value is less than 5%. Therefore, the null hypothesis is rejected, indicating that the sequence exhibits autocorrelation.

Table 11: Unit root test

	ADF	**PP**	**KPSS**
Level	-1.94294(10)	-1.72797(10)	107.4131(0)*
1^{st} difference	-11.5604(9)*	-41.1503(0)*	0.0411 (4)

Notes: * represents statistical significance at the 0.05 level.

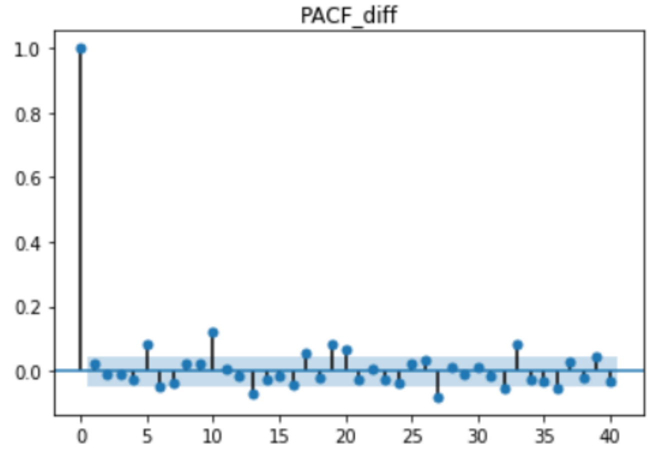
Table 12: White noise test

	***p*-value**	**Result**
1^{st} difference of series	2.5183374624102813e-14	Reject the null hypothesis

Notes: The null hypothesis $\boldsymbol{H_0}$ indicates series are purely random processes.

3.2.2 PACF and ACF

Using the unit root test, ARIMA(0,1,0) is proven to be stationary. Figure 3 illustrates that the highest protruding part of PACF(p) is close to 10, and the highest protruding part of ACF(q) is also close to 10. As a consequence, this study evaluates the models of ARIMA(0,1,1), ARIMA(0,1,2),…, up to ARIMA(10,1,10), and finds out the top three of them.



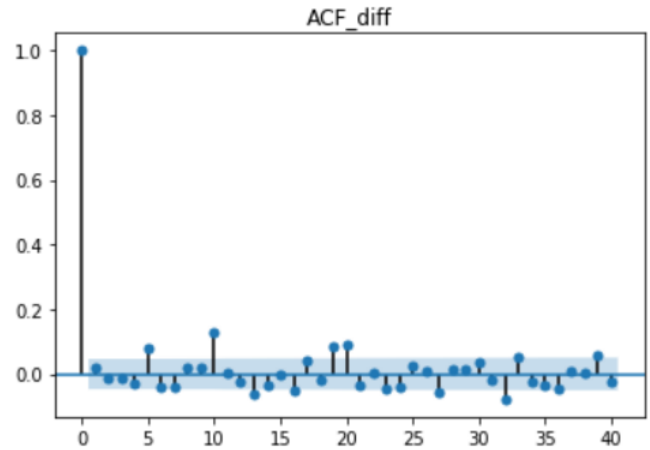


Figure 3: PACF (top panel) and ACF (bottom panel) after the 1^{st} difference in Bitcoin prices.

3.2.3 AIC and BIC

Table 13 demonstrates the evaluation results for the top three ARIMA models with the best goodness-of-fit of AIC and BIC. AIC selects ARIMA(10,1,10), ARIMA(9,1,8), ARIMA(8,1,9), and BIC selects ARIMA(0,1,0), ARIMA(0,1,1), ARIMA(1,1,0).

Table 13: Goodness-of-fit test for ARIMA (p,d,q)

Ranking	ARIMA	AIC	Ranking	ARIMA	BIC
1	ARIMA(10,1,10)	25202.50	1	ARIMA(0,1,0)	25289.98
2	ARIMA(9,1,8)	25207.96	2	ARIMA(0,1,1)	25296.76
3	ARIMA(8,1,9)	25212.74	3	ARIMA(1,1,0)	25296.78

3.2.4 Performance evaluation

After the AIC and BIC tests, the prediction errors of the ARIMA models are compared using MSE, MAE, and MAPE. As shown in Tables 14–16, ARIMA (10,1,10) is the best model based on its MSE performance evaluation; as evaluated by MAE and MAPE, ARIMA (9,1,8) has the best predictive ability. Based on the white noise test, the residual of the ARIMA (10,1,10) model's *p*-value is 0.24032265 > 0.05, and the null hypothesis is not rejected, indicating that the Bitcoin price prediction was completely expressed by the ARIMA (10,1,10) model. However, the *p*-value of ARIMA (9,1,8) is 9.90614734E–04 < 0.05, which means that the null hypothesis could be rejected. This result indicates that the residuals of ARIMA (9,1,8) exhibit autocorrelation, so we should add a dynamic model to the analysis.

Table 14: Evaluation of top three ARIMA models using MSE

MSE	Model	MSE
1	ARIMA(10,1,10)	4522056.233
2	ARIMA(9,1,8)	4522625.575
3	ARIMA(8,1,9)	4543247.409

Table 15: Evaluation of top three ARIMA models using MAE

MAE	Model	MAE
1	ARIMA(9,1,8)	1873.794
2	ARIMA(10,1,10)	1874.558
3	ARIMA(8,1,9)	1879.068

Table 16: Evaluation of top three ARIMA models using MAPE

MAPE	Model	MAPE
1	ARIMA(9,1,8)	1853.963%
2	ARIMA(10,1,10)	1855.142%
3	ARIMA(8,1,9)	1859.533%

3.3 ARIMA-GARCH family of models

3.3.1 Determination of GARCH models

As previously established, the ARIMA(9,1,8) residuals exhibit autocorrelation. Thus, the GARCH family of models are used to extend the ARIMA (9,1,8) model. We chose GARCH, EGARCH, and GJR-GARCH. Based on the literature review, GARCH(1,1), is the optimal predictive model.
Therefore ARIMA(9,1,8)-GARCH(1,1), ARIMA(9,1,8)-EGARCH(1,1,1), and ARIMA(9,1,8)-GJR-GARCH(1,1,1) are used.

3.3.2 Performance evaluation for ARIMA-GARCH family of models

We implement MSE, MAE, and MAPE to assess the predictive accuracy of the ARIMA-GARCH family of models. Table 17 illustrates that ARIMA(9,1,8)-EGARCH(1,1,1) is the optimal model as evaluated by the three performance indicators. In Table 19, the white noise test is performed on the residual sum of squares of the ARIMA(9,1,8)-GARCH(1,1), ARIMA(9,1,8)-EGARCH(1,1,1), and ARIMA(9,1,8)-GJR-GARCH(1,1,1) models. The results show that all of their *p*-values are greater than 0.05, and the null hypothesis is not rejected. The residual sum of squares of the GARCH models are random sequences and completely explain Bitcoin price volatility.

Table 17: The top four econometric models evaluated by MSE, MAE, and MAPE

ARIMA	MSE ranking		MAE ranking		MAPE ranking	
ARIMA(9,1,8)-EGARCH(1,1,1)	1	1125136.78	1	984.9482	1	1017.696%
ARIMA(9,1,8)-GARCH(1,1)	2	2833556.82	2	1461.839	2	1443.157%
ARIMA(9,1,8)-GJR-GARCH(1,1,1)	3	3041903.84	3	1508.479	3	1485.892%
ARIMA (10,1,10)	4	4522056.23	5	1874.558	5	1855.142%
ARIMA (9,1,8)	5	4522625.57	4	1873.794	4	1853.963%

Table 18: The residual sum of squares of GARCH models

Model	***p*-value**	**Result**
ARIMA(9,1,8)-GARCH(1,1)	0.91745257	The residual sum of squares is a white noise series
ARIMA(9,1,8)-EGARCH(1,1,1)	0.99954981	The residual sum of squares is a white noise series
ARIMA(9,1,8)-GJR-GARCH(1,1,1)	0.99741978	The residual sum of squares is a white noise series

Notes: The null hypothesis H_0 indicates that the series is random.

3.4 Comparison of BPNN and time series models

We compare the prediction errors of the best BPNN with one and two hidden layers and the best time series model. Table 19 demonstrates that the BPNN models have smaller prediction errors and predict more accurately than ARIMA(9,1,8)-EGARCH(1,1,1). Therefore, BPNN forecasts more precisely than time series models in terms of Bitcoin price prediction.

Table 19: Evaluation of model performance

MSE ranking	**1**	**2**	**3**
Model	BPNN(2-4-3-1)	BPNN(2-4-1)	ARIMA(9,1,8)-EGARCH(1,1,1)
MSE	431,041.920	443,493.359	1,125,136.78
MAE ranking	**1**	**2**	**3**
Model	BPNN(2-4-3-1)	BPNN(2-4-1)	ARIMA(9,1,8)-EGARCH(1,1,1)
MAE	511.581	513.648	984.948
MAPE ranking	**1**	**2**	**3**
Model	BPNN(2-4-3-1)	BPNN(2-4-1)	ARIMA(9,1,8)-EGARCH(1,1,1)
MAPE	589.767%	593.672%	1,017.696%

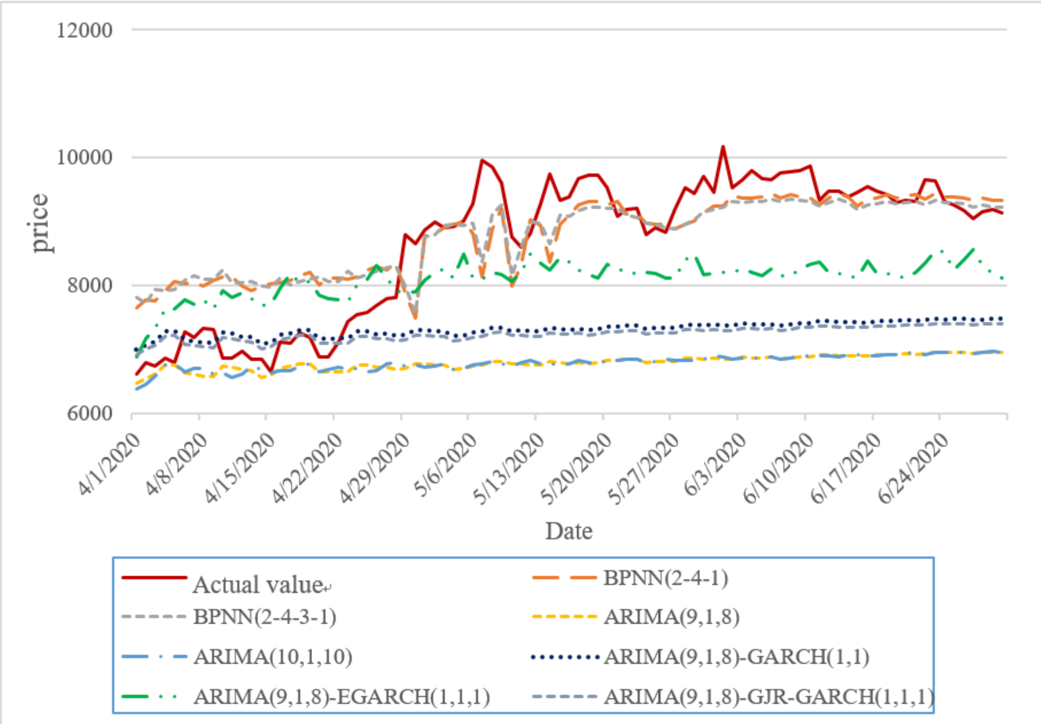


Figure 4: Comparison of the actual value, the prediction values of BPNNs, and the prediction values of time series models

4. Conclusions

In this study, predicting the Bitcoin price trend is performed by adopting BPNN and time series models. Two new input variables for BPNN were extracted using PCA from Bitcoin's past three-day closing prices, MA5, MA20, daily trading volume, Ether price, and Ripple price. We use python to program BPNN and time series models, and then apply the sigmoid function as the transfer function. The BPNN structure with the smallest predictive error has a 0.1 learning rate, a 0.7 momentum term, and 4 and 3 neurons in the first and the second hidden layers, respectively. We utilize AIC and BIC to select the best-fitting ARIMA model and compare the prediction errors using MSE, MAE, and MAPE. We also perform the white noise test on the residuals. The results indicated that ARIMA(10,1,10) residuals fully expressed the sequence information; however, ARIMA(9,1,8) residuals exhibited autocorrelation. Therefore, we established the GARCH, EGARCH, and GJR-GARCH models to explain the volatility.

According to the results, the predicted price trends using ARIMA models are far from the actual trends. Nevertheless, the predictive ability improved significantly after augmenting the ARIMA models with GARCH to express volatility. The ARIMA-EGARCH model is the most precise time series model due to its consideration of the news influence. BPNN with one and two hidden layers provided more accurate predictions than the time series models and BPNN with two hidden layers outperformed any other model. To summarize, investors can refer to the impact of news about Bitcoin and price volatility to predict Bitcoin prices using BPNN with two hidden layers.

Acknowledgments

Yu-Min Lian is grateful to the Ministry of Science and Technology (MOST) for support through Project No.: MOST109-2410-H-030-019-MY2.

References

[1] Akaike, H. (1974). A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19, 6, pp. 716 - 723.
[2] Bollerslev, T., Chou, R.Y. and Kroner, K.F. (1992). ARCH modeling in finance: A review of the theory and empirical evidence. Journal of Econometrics, 52, 1-2, pp. 5-59.
[3] Box, G. E. P. and Jenkins, G. M. (1976). Time series analysis forecasting and control, 2nd ed. San Francisco: Holden-Day.
[4] Chen, C. C., Chang, J. H., Lin, F. C., Hung, J. C., Lin, C. S. and Wang, Y. H. (2019). Comparison of forecasting ability between backpropagation network and ARIMA in the prediction of Bitcoin price. 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS).
[5] Corbet, S., Larkin, C. and Lucey, B. (2020b). The contagion effects of the COVID-19 pandemic: Evidence from gold and cryptocurrencies. Finance Research Letters, 35, 101554.
[6] Corbet, S., Larkin, C., Lucey, B., Meegan, A. and Yarovaya, L. (2020a). The impact of macroeconomic news on Bitcoin returns. European Journal of Finance, 26, 14, pp. 1396-1416.
[7] Glosten, L.R., Jagannathan, R. and Runkle, D.E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. The Journal of Finance, 48, pp. 1779-1801.
[8] Grudnitski, G. and Osburn, L. (1993). Forecasting S&P and gold futures prices: An application of neural networks. Journal of Futures Markerts. 13, 6, pp. 631-643.
[9] Katsiampa, P. (2017). Volatility estimation for Bitcoin: A comparison of GARCH models. Economics Letters, 158, pp. 3-6.
[10] Lama, A., Jha, G. K., Paul, R. K. and Gurung, B. (2015). Modelling and forecasting of price volatility: An application of GARCH and EGARCH models. Agricultural Economics Research Review, 28, pp. 73-82.
[11] Lian, Y. M. and Chen, J. H. (2021). Pricing virtual currency-linked derivatives with time-inhomogeneity. International Review of Economics and Finance, 71, pp. 424-439.

[12] Lian, Y. M., Cheng, C. H., Lin, S. H. and Lin, J. H. (2019). A cost of carry-based framework for the Bitcoin futures price modeling. Journal of Mathematical Finance, 9, pp. 42-53.
[13] Liu, X. and Ma, X. (2012). Based on BP neural network stock prediction. Journal of Curriculum and Teaching, 1, 1, pp. 45-50.
[14] McClelland J. L. and Rumelhart D. E. (1986). Parallel distributed processing: Explorations in microstructure of cognition. MA: MITPress, 1.
[15] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.
[16] Nelson, D.B. (1991). Conditional heteroskedasticity in asset returns: A new approach. Econometrica, 59, 2, pp. 347-370.
[17] Ohyver, M. and Pudjihastuti, H. (2018). ARIMA model for forecasting the price of medium quality rice to anticipate price fluctuations. Procedia Computer Science, 135, pp. 707-711.
[18] Olson, D. and Mossman, C. (2003). Neural network forecasts of Canadian stock returns using accounting ratios. International Journal of Forecasting, 19, 3, pp. 453-465.
[19] Qiao, G., Yang, J. and Li, W. (2020). VIX forecasting based on GARCH-type model with observable dynamic jumps: A new perspective. North American Journal of Economics and Finance, 53, 101186.
[20] Schwarz, G. (1978). Estimating the dimension of a model. Annals of Statistics, 6, 2, pp. 461-464
[21] Wood, D. and Dasgupta, B. (1996). Classifying trend movements in the MCSI U.S.A. capital index–A comparison of regression, ARIMA and neural network method. Computer and Operations Research, 23, 6, pp. 611-622.