AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(6): 13907–13922.
DOI: 10.3934/math.2023712
Received: 07 January 2023
Revised: 16 March 2023
Accepted: 26 March 2023
Published: 13 April 2023

Research article

Bitcoin volatility forecasting: An artificial differential equation neural network

S. Pourmohammad Azizi[1], **Chien Yi Huang**[2], **Ti An Chen**[3,*], **Shu Chuan Chen**[2] **and Amirhossein Nafei**[2,4]

[1] Department of Electrical Engineering, National Taiwan Ocean University, Keelung, Taiwan
[2] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan
[3] Department of Business Administration, JinWen University of Science Technology, Taipei, Taiwan
[4] Department of Mathematics, Semnan University, Semnan, Iran

* **Correspondence:** Email: nhchen59@just.edu.tw.

Abstract: In this article, an alternate method for estimating the volatility parameter of Bitcoin is provided. Specifically, the procedure takes into account historical data. This quality is one of the most critical factors determining the Bitcoin price. The reader will notice an emphasis on historical knowledge throughout the text, with particular attention paid to detail. Following the production of a historical data set for volatility utilizing market data, we will analyze the fundamental and computed values of Bitcoin derivatives (futures), followed by implementing an inverse problem modeling method to obtain a second-order differential equation model for volatility. Because of this, we can accomplish what we set out to do. As a direct result, we will be able to achieve our objective. Following this, the differential equation of the second order will be solved by an artificial neural network that considers the dataset. In conclusion, the results achieved through the utilization of the Python software are given and contrasted with a variety of other research approaches. In addition, this method is determined with alternative ways, and the outcomes of those comparisons are shown.

Keywords: Bitcoin; volatility; differential equation; artificial neural network; forecasting; inverse problem
Mathematics Subject Classification: 91-08, 68T01

1. Introduction

Bitcoin is a virtual currency that gained popularity and prosperity when its price reached more than 13,000 USD in early 2018. This digital currency was a combination of creativity, overcoming legal barriers, and eliminating intermediaries in various financial and banking affairs that made financial transactions possible at the international level. Therefore, in the short time of its emergence, it attracted much attention, and a bright future awaits virtual currencies. This currency has been associated with very high price fluctuations and has grown significantly in this short period . As seen in Figure 1, the tangible price growths of bitcoin occurred in 2018 and 2021 and peaked at highs of approximately 14000 USD and 64000 USD, respectively. It is expected that this currency will have more growth because the maximum supply of this currency is limited to 21,000,000 units. This restriction and varied news about cryptocurrencies has caused this currency to fluctuate more . Given the growing fluctuations of this digital currency, many studies have been done in this area. One of the topics we can use to understand its behavior better is the study of the volatility parameter. By getting an accurate estimate of Bitcoin Volatility and accurately estimating this currency's behavior in the future, we can correct the price of many of its related financial derivatives.

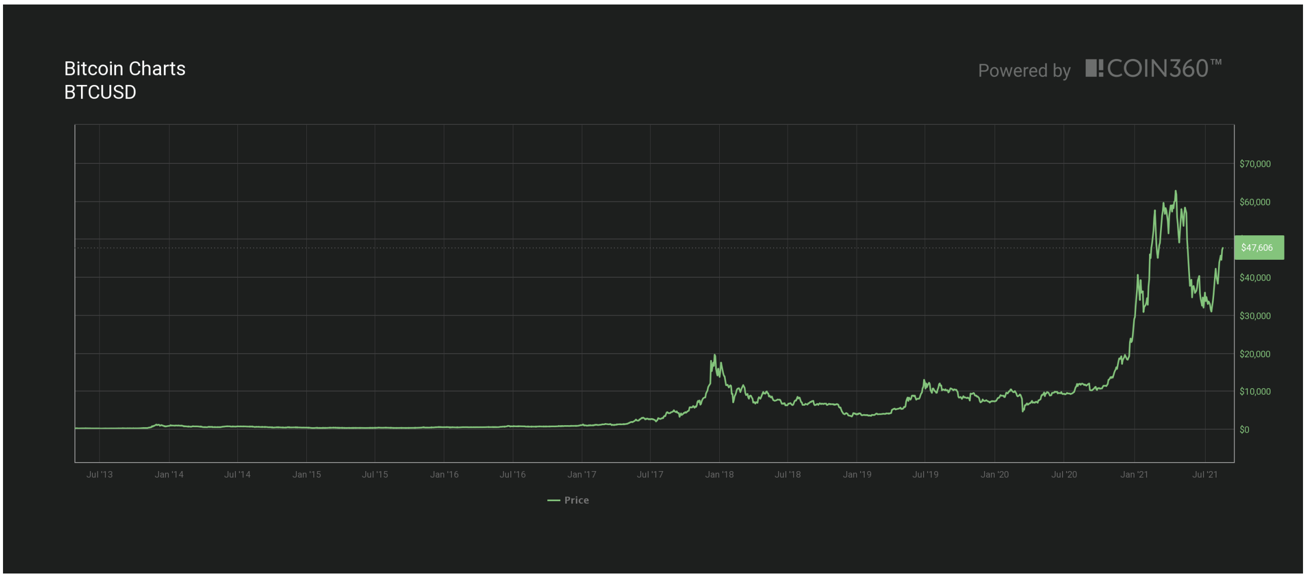

Figure 1. Bitcoin price.

Black and Scholes [8] considered six assumptions for pricing financial assets. Since then, many articles have been published on pricing financial assets without considering some of these conditions. One of these conditions is that the volatility parameter is constant, which will be obtained using the data variance directly [7]. However, this parameter is generally dynamicin the real world, and Merton was one of the first to study it [44]. Another critical issue in line with these problems is the behavior of this parameter in the future. Recognizing future volatility behavior as a risk indicator in various financial areas such as risk management, portfolio optimization, asset pricing, and several financial behaviors is essential ([6, 9, 20, 46]). Many studies have been conducted on modeling and predicting volatility in the stock market ([3, 47, 48]), and an attempt has been made to provide an efficient method or model for it. However, volatility behavior in the cryptocurrency market is different. Different methods have been presented to estimate this parameter, and one of these methods is GARCH-type models. Glaser et al. 2014 [27]; Gronwald 2014 [32] using linear GARCH, Dyhrberg 2016 [1]; Bouoiyour and Selmi

2015 [11]; Bouri et al. 2017 [13] adopting the Threshold GARCH, Dyhrberg 2016 [23] applying Exponential GARCH, tried to estimate this parameter. Although they opened a new view , these researchers generally considered just one heteroskedasticity model. Katsiampa compared different mentioned GARCH models and their ability in other circumstances [37]. The ARCH and GARCH families are generally used as tools for Volatility modeling. One of the problems with these methods is that each member of this family depends on specific conditions and may lose effectiveness by changing the conditions. For example, GARCH is unsuitable in asymmetric conditions and does not have a suitable output [22, 15]. On the other hand, the approach of using past data to predict the future of this family is beneficial. One of the powerful instruments to solve this problem is differential equations, which will not change the method used as the conditions change, and the future value depends on past weights. Therefore, this paper will use differential equations to estimate future volatility value. Another problem with the GARCH family of models is that they are generally used for low-frequency data without considering the nonlinear properties of the information and are not well responsive to high frequencies [24, 41]. To solve the problem of high-frequency data, Corsi [17] proposed the heterogeneous autoregressive model of realized volatility (HAR-RV) for heterogeneous markets. Also, Andersen et al. [2] introduced the autoregressive fractionally integrated moving average (ARFIMA) prediction model. Although the ARFIMA and HAR-RV models are well adapted for high-frequency data, they still do not work well for cases where the data behavior is nonlinear [41]. Many studies were presented to solve such problems ([10, 18, 26, 28, 43]), but these studies were still based on the old econometrics methods. With the introduction of different forms of machine learning, many of these problems were solved. Among the various fields of machine learning, deep learning has solved many problems in the mentioned methods alone. For example, Karaoglu et al. [36] use the model provided by Graves [30] to predict stock prices. Bao et al. [5] considered technical indicators as instruments to forecast stock prices using deep learning. This approach was not limited to stock price predicting but was also used in volatility forecasting (e. g. [41, 56]). Lei et al. [41] used this instrument to predict volatility based on high-frequency financial data and showed that this method is much better than the ARCH and GARCH families. Deep learning features that can be a disadvantage are sufficient data and high error cases, so using dynamic systems and differential equations can help us control these cases. In this study, we want to combine machine learning and differential equations and investigate the result with direct machine learning.

In addition to the mentioned methods, Azizi and Neisy [4] modeled volatility using past data and linear regression, but obviously, linear regression will not work well. Hoang and Baur [34] tried to estimate implied volatility using AutoRegressive Moving Average or Heterogeneous AutoRegressive in Bitcoin options traded on the options exchange Deribit. Their results show it does not work well for the lower weekly timeframe. One of the essential advantages of differential equations is their time flexibility, and they can adapt to a different timetable according to the accessible data. The inverse problem is also another method used for estimating volatility. Chiarella et al. [16] developed an efficient, less complex process for evaluating by considering inverse problems and studying Lagnado and Osher's work [40]. In fact, by using inverse problems, they reached a PDE for volatility, and by solving it, they obtained a suitable model. Furthermore, Neisy and Salmani [45] and Xu and Jia [55] applied the Chiarella et al. method to the jump-diffusion processes asset model. We also use this modeling approach in part of our work. Inverse problems are compelling but have a fundamental problem. These methods are available for predicting and working with considering assumptions over a while. Hence, we use this method to arrive at a differential equation and combine it with machine learning. This paper provides a new approach to estimating Bitcoin volatility that covers the mentioned problems, such as volatility model inconsistency with issue conditions and lack of unique data. It uses available data to obtain a suitable model from an Artificial Neural Network (ANN) framework. In detail, by comparing the actual and computed value of Bitcoin Future price, we generate a volatility historical data set and, using Chiarella et al. inverse problem modeling idea, reach a second-order differential equation (ODE) for it. Finally, it will be solved by applying ANN. As mentioned, in this study, we try to predict and estimate the volatility value of Bitcoin using a historical volatility data value obtained from the market and comparing the actual value and computed value of this cryptocurrency's future price. For this purpose, the next section is dedicated to pricing the future, and section 3 is about obtaining the volatility ODE and solving it by ANN. Finally, in section 4, we investigate the performance of this method with actual data and show results by Python.

2. Future pricing

This section focused on modeling and pricing the Bitcoin Future using mathematical concepts. For this purpose, consider that the Bitcoin price changes follow a stochastic process as below:

$$dB(\tau) = \alpha B(\tau)d\tau + \sigma B(\tau)dw, \tag{2.1}$$

where α, σ and w are drift, volatility and Wiener process and also consider $F(B(\tau), \tau)$ as a Bitcoin Future price with Bitcoin price $B(\tau)$ at time τ.

According to the multiplication property of differentials as follows [39]:

$$\begin{gathered} d\tau \times d\tau = 0, \\ d\tau \times dw = 0, \\ dw \times dw = d\tau, \end{gathered}$$

and using Ito lemma, the price changes of the future is as [31], [49]:

$$dF = F_\tau d\tau + F_B dB + \frac{1}{2}F_{BB}dBdB, \tag{2.2}$$

where

$$dBdB = \sigma^2 B^2 dt. \tag{2.3}$$

With simplification, we obtain the following:

$$dF = (F_\tau + \alpha B F_B + \frac{1}{2}\sigma^2 B^2 F_{BB})d\tau + \sigma B F_B dw. \tag{2.4}$$

Applying the above equations to the Feynman-Kac formula and changing the time variable as $t = T - \tau$, we have the future price PDE as follows [35, 50]:

$$F_t = \alpha B F_B + \frac{1}{2}\sigma^2 B^2 F_{BB} - rF, \tag{2.5}$$

we consider the initial condition that shows the price at contract time as bellow [19]:

$$F(B(0),0) = B(0), \tag{2.6}$$

and boundary conditions:

$$\begin{aligned} &\lim_{B(t)\to 0} F(B(t),t) = 0,\\ &\lim_{B(t)\to\infty} F(B(t),t) = \lim_{B(t)\to\infty} B(t). \end{aligned} \tag{2.7}$$

We consider the finite difference method using [42, 54] to solve this PDE. Hence, without losing the generality of the problems, consider the Bitcoin price changes as a limited interval in the form of $B_0 = B_{\min} \leq B_1 \leq ... \leq B_n = B_{\max}$ and by this discretization use the iterative method as

$$F^{k+1} = AF^k + Bo, \tag{2.8}$$

where the $F^{k=0}$ can come by the mentioned initial value 2.6, k is index of time discretization, A is derivative matrix and Bo is boundary condition as bellow:

$$\begin{aligned} &A = \begin{bmatrix} \gamma_1 & \delta_1 & & & 0 \\ \beta_2 & \gamma_2 & \delta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \gamma_{n-2} & \delta_{n-2} \\ 0 & & & \beta_{n-1} & \gamma_{n-1} \end{bmatrix}_{(n-1)\times(n-1)}, \\ &Bo = [0,...,0,\delta_{n-1}B_{\max}]^{Tra.}_{1\times(n-1)}, \\ &t_0 = 0 \leq t_1 \leq ... \leq t_k \leq ... \leq t_K = T, \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} &\beta_i = \left(\sigma^2 {B_i}^2 / \left(\Delta B^2\right) - \alpha B_i / (2\Delta B)\right)\Delta t, \\ &\gamma_i = 1 - \left(r + \sigma^2 {B_i}^2 / \left(\Delta B^2\right)\right)\Delta t, \\ &\delta_i = \left(\sigma^2 {B_i}^2 / \left(\Delta B^2\right) + \alpha B_i / (2\Delta B)\right)\Delta t, \\ &\Delta B = B_{i+1} - B_i, \\ &\Delta t = t_{k+1} - t_k. \end{aligned} \tag{2.10}$$

To obtain the volatility in each time step, we estimate it by equation the comparison 2.8 with real value. In other words, we suppose that the actual future value in the market at time t_{k+1} with Bitcoin price B_i is $F_i^{k+1^*}$ and by replacing $F_i^{k+1^*}$ on the PDE 2.5, the following equation will be obtained as follows:

$$\Omega(\sigma_k) = -\frac{F_i^{k+1^*} - F_i^k}{\Delta t} + \alpha B_i \frac{F_{i+1}^k - F_{i-1}^k}{2\Delta B} + 0.5\sigma_k^2 B_i^2 \frac{F_{i+1}^k - 2F_i^k F_{i-1}^k}{\Delta B^2} - rB_i, \tag{2.11}$$

Now, considering the Newton-Raphson, the estimation of σ_k value is as below:

$$\sigma_k^{j+1} = \sigma_k^j - \frac{\Omega(\sigma_k^j)}{\Omega_{\sigma_k}(\sigma_k^j)}, \tag{2.12}$$

In this section, the Bitcoin Future is modeled, and the amount of volatility is estimated using it. Although volatility data can be obtained from the market, we have tried to provide a method for estimating it if it is not available from the market, or similar studies do not encounter problems calculating volatility when they want to use this method.

3. Bitcoin volatility estimation

Bitcoin is one of the most volatile cryptocurrencies. In this section, we want to predict the daily volatility value by mathematical concepts by reducing the difference between the computed and actual value of the Bitcoin Future. For this matter, suppose σ is a function that depends on time. The real future is shown as $F^*(\sigma(t), B(t), t)$ and using this notation, we want to minimize the bellow function:

$$\hat{J}(\sigma(t)) = \int_0^\infty \int_0^\infty (F(\sigma(t); B(t), t) - F^*(\sigma(t); B(t), t))^2 dB(t)dt. \tag{3.1}$$

Theorem 1. Let X, and Y be normed spaces and $K : X \to Y$ be a linear compact operator with nullspace $\aleph(K) := \{x \in X : Kx = 0\}$ Let the dimension of $X/\aleph(K)$ be infinite. Then there exists a sequence $(x_n) \subset X$, such that $Kx_n \to 0$ but (x_n) does not converge. We can even choose (x_n) such that $\|x_n\| \to \infty$.

Proof. See [38] page 12. □

According to the nature of financial data and the above theorem 1, the expression 3.1 is generally ill-posed. In this situation, [16], we can use Tikhonov regularization and rewrite the above function as follows:

$$\bar{J}(\sigma(t)) = \int_0^\infty \int_0^\infty (F(\sigma(t); B(t), t) - F^*(\sigma(t); B(t), t))^2 dB(t)dt + \lambda|\sigma(t)|^2, \tag{3.2}$$

where λ is the regularization parameter. Without losing the generality of problems, consider the following equation:

$$G(\sigma(t); t) = \int_0^\infty (F(\sigma(t); B(t), t) - F^*(\sigma(t); B(t), t))^2 dB(t), \tag{3.3}$$

so we have:

$$\bar{J}(\sigma(t)) = \int_0^\infty G(\sigma(t); t)dt + \lambda|\sigma(t)|^2. \tag{3.4}$$

Suppose $\sigma(t) \in L^2$ or in other word consider $\sigma(t)$ is member of the twice differentiable space in interval $(0, \infty)$, so we have [16]

$$\int_0^\infty |\sigma(t)|^2 dt \leq c \int_0^\infty \left(\frac{d\sigma(t)}{dt}\right)^2 dt. \tag{3.5}$$

for some $c > 0$. Replacing the above integration, we reach

$$J(\sigma(t)) = \int_0^\infty G(\sigma(t); t) + \lambda\left(\frac{d\sigma(t)}{dt}\right)^2 dt, \tag{3.6}$$

In this regard, consider the below theorem.

Theorem 2. (The Euler-Lagrange Equation) If $I(Y)$ is an extremum of the functional

$$I(Y) = \int_a^b Lg(x, y, y')dx,$$

defined on all twice differentiable functions space such that $y(a) = \varphi$, $y(b) = \psi$, then $Y(x)$ satisfies the second order ordinary differential equation

$$\frac{d}{dx}\left(\frac{\partial Lg}{\partial y'}\right) - \frac{\partial Lg}{\partial y} = 0. \tag{3.7}$$

Proof. See [51], and [53] page 20. □

Now suppose

$$Lg = G(\sigma(t); t) + \lambda\left(\frac{d\sigma(t)}{dt}\right)^2. \tag{3.8}$$

Using the Euler-Lagrange equation, we have

$$\frac{d}{dt}\frac{\partial Lg}{\partial \sigma_t} - \frac{\partial Lg}{\partial \sigma} = 0, \tag{3.9}$$

where,

$$\begin{aligned} \frac{d}{dt}\frac{\partial Lg}{\partial \sigma_t} &= 2\frac{\partial^2 \sigma}{\partial t^2}, \\ \frac{\partial Lg}{\partial \sigma} &= \frac{\partial G(\sigma(t); t)}{\partial \sigma}, \end{aligned} \tag{3.10}$$

so we obtain:

$$\frac{\partial^2 \sigma}{\partial t^2} = \theta \frac{\partial G(\sigma(t); t)}{\partial \sigma}. \tag{3.11}$$

Since the value of $F^*(\sigma(t), B(t), t)$ is not accessible for all of t and $B(t)$, obtaining the model of $\theta\frac{\partial G(\sigma(t);t)}{\partial \sigma}$ is not possible easily. So, we try to estimate it with accessible, accurate data on Bitcoin volatility that can be obtained from the market or calculated by comparing the actual future value and computed future price received from the numerical method in the previous section. For this matter, we can consider the following equation:

$$g(\sigma(t), t) = \theta \frac{\partial G(\sigma(t); t)}{\partial \sigma}, \tag{3.12}$$

and rewrite the above second-order differential equation to a set of first-order differential equations as

$$\begin{cases} \frac{\partial \sigma}{\partial t} = y, \\ \frac{\partial y}{\partial t} = g(\sigma(t), t). \end{cases} \tag{3.13}$$

To solve the set 3.13, at first, the $g(\sigma(t), t)$ value must be obtained. For this purpose, we consider a neural network to estimate the $g(\sigma(t), t)$ as follows, Figure 2:

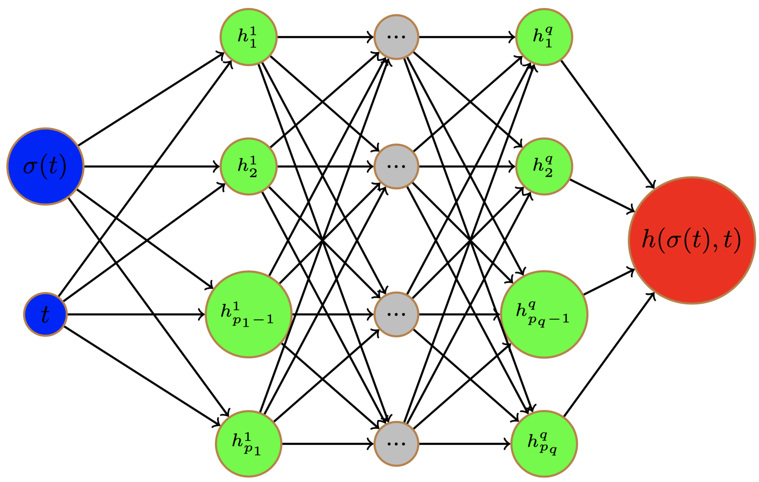

Figure 2. Neural Network graph.

According to the neural network graph, $\sigma(t)$ and t are the input value of the neural network, and $h(\sigma(t), t)$ is an estimation of $g(\sigma(t), t)$. The question is, "What is the output of this supervised learning?" for starting training. As mentioned, we have the volatility value at each time step, so consider the data set as follows:

$$D = \{(t_0, \sigma(t_0)), ..., (t_m, \sigma(t_m))\}, \Delta t = t_{i+1} - t_i, \tag{3.14}$$

and using Wang et al. study [52], the approximated $g(\sigma(t), t)$ value in these points is

$$\frac{dy(t_i)}{dt} = g(\sigma(t_i), t_i) \approx \begin{cases} \frac{-\sigma(t_{i+2})+4\sigma(t_{i+1})-3\sigma(t_i)}{2\Delta t} & i = 1, \\ \frac{\sigma(t_{i+1})-\sigma(t_{i-1})}{2\Delta t} & i = 2, ..., n-1, \\ \frac{3\sigma(t_i)-4\sigma(t_{i-1})+\sigma(t_{i-2})}{2\Delta t} & i = n. \end{cases} \tag{3.15}$$

Using $\sigma(t)$ and t as inputs and considering the above approximation as the output, the training data set is complete, and we can start training. Mean Squared Error is utilized as the cost function to train this network, and also this network is constructed using Tensorflow and trained with Adam optimizer.

Algorithm

Step 1. Data collection: Using historical data (previous volatility and future values), make the dataset as bellow:

$$D = \{(t_0, \sigma(t_0)), (t_1, \sigma(t_1)), ..., (t_m, \sigma(t_m))\},$$

Step 2. Data preparation: prepare the dataset to adapt differential equations and training problems:

$$\bar{D} = \left\{\left(X_0, \frac{dy(t_0)}{dt}\right), \left(X_1, \frac{dy(t_1)}{dt}\right), ..., \left(X_m, \frac{dy(t_m)}{dt}\right)\right\},$$

where $X_i = (t_i, \sigma_i)$ and $dy(t_i)$ obtained from Eq 3.15;

Step 3. Model training: try to find a suitable approximation of $g(\sigma(t), t)$ using $\bar{D}$ dataset and training and show it by $h(\sigma(t), t)$;

Step 4. Forecasting: predict future value using Eq 3.13 as bellow:

$$\begin{cases} \sigma_{k+1} = \sigma_k^* + y_k dt, \\ y_k = y_{k-1} + h(\sigma_k^*, t_k) dt, \end{cases}$$

where σ_{k+1} and σ_k^* are predicted volatility value for time t_{k+1} and true value of volatility that happened at time t_k.

4. Numerical result

In this section, we tend to investigate the performance of our method. Hence, we want to predict the value of Bitcoin volatility using real data value. For this matter, consider the actual value of Bitcoin Future from 1-Jan-2021 to 29-Jul-2021, whose maturity is 21-Aug-2021, as shown in Figure 3.

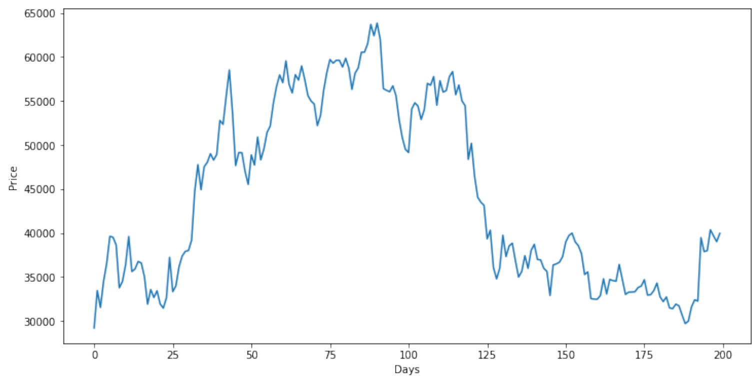

Figure 3. Bitcoin Future price.

Using the previous section's numerical method to price future and volatility value in the market, we form the volatility data set with its value at each time step and try to train the model.

Using $\sigma(t)$, t, and a neural network with four layers such as (2,30,20,10,1), the y value is obtained as shown in Figure 4.

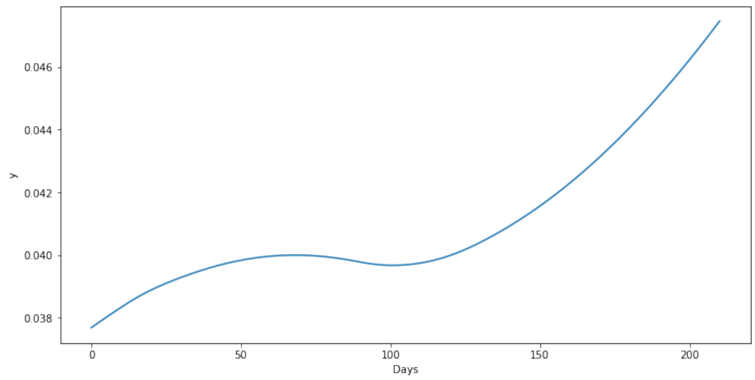

Figure 4. 'y' value.

In the following, considering the 150 days volatility value as training data and the next 50 days as testing data, the resulta are shown in Figure 5.

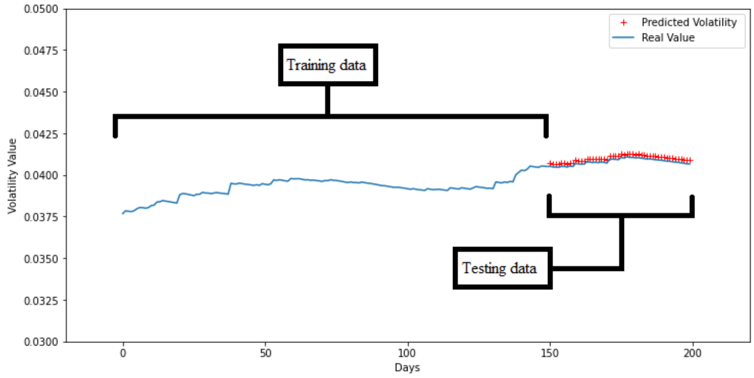

Figure 5. Volatility prediction.

Where its relative error is as Figure 6 shows the high performance of our methods with 10^{-2} accuracy in relative error.

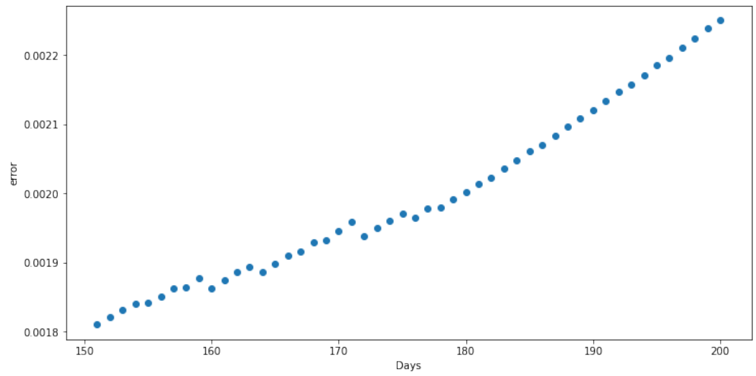

Figure 6. Relative error.

In the following, the presented method is compared to ARCH, GARCH, and Deep Learning. For comparing the approaches, we firstconsider the dataset 3.14 and present the model. The first model is ARCH as follows (for more details, see [29]):

$$\begin{gathered} \sigma_t = v_t \varepsilon_t \\ v_t = \sqrt{a_0 + a_1 \sigma_{t-1} + a_2 \sigma_{t-2} + \ldots + a_p \sigma_{t-p}}\ , \\ \varepsilon_t \in N(0,1) \end{gathered} \tag{4.1}$$

Where $a_i > 0$ is estimated using historical data. Also, the GARCH is as (for more details, see [25]):

$$\begin{gathered} \sigma_t = v_t \varepsilon_t \\ v_t = \sqrt{a_0 + a_1 \sigma_{t-1} + a_2 \sigma_{t-2} + \ldots + a_p \sigma_{t-p} + b_1 v_{t-1} + b_2 v_{t-2} + \ldots + b_q v_{t-q}}\ , \\ \varepsilon_t \in N(0,1) \end{gathered} \tag{4.2}$$

where $a_i > 0$ and $b_j > 0$. For Deep Learning, considering dataset 3.14, try to train the model using a neural network like mentioned in section 3 with t input and $\sigma(t)$ output [41]. Quantifying $p = 15$ and $q = 15$, the results is as Figure 7.

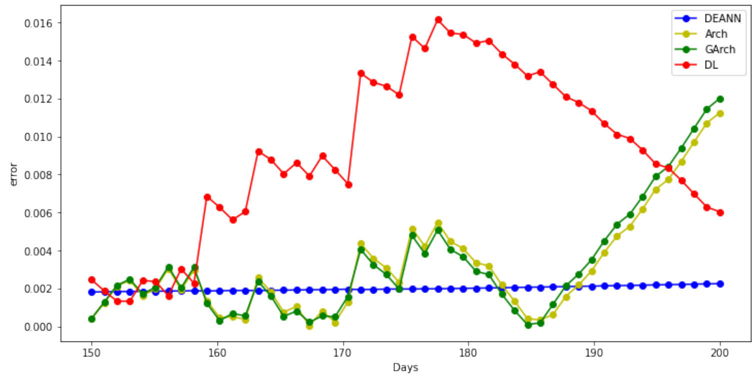

Figure 7. Relative error comparison.

The line chart 7 illustrates the change in prediction error value in each subsequent 50 days for five forecasting methods. Overall, as can be seen, all forms fluctuated in this period except this research approach (DEANN). Although Lei et al. [41] study result, the Deep learning (DL) method did not work well for this example compared to ARCH and GARCH. However, their error values converge through time. Since Lei et al. [41] used the LSTM DL technique, which employs a highly complex network, gradient vanishing may occur. Thus, this flaw may cause the example's poor performance. Although DL did not meet our expectations, the ARCH and GARCH performed suitably in the fifty percent of the route. However, in the following, with short volatility, the amount of error has grown tangible quickly. With all of the oscillations in different approaches, the DEANN method has the slightest fluctuation. The combination of differential equations and neural networks leads us to the outputs whose value change moves parallel to the actual value and is so close to them.

In addition to assessing Bitcoin Volatility, this strategy's effectiveness was evaluated using two estimations of the volatility of Google and Amazon shares, as shown in Figure 8.

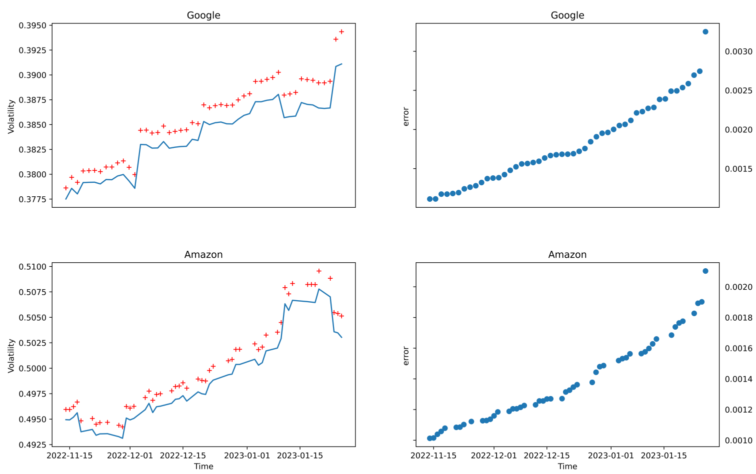

Figure 8. Google and Amazon volatility estimation.

Our technique predicted the volatility of Google and Amazon shares with a tiny margin of error, as shown in Figure 8. In the following, to investigate this method's prediction error, the histogram and density diagram for these section numerical results are prepared in Figure 9.

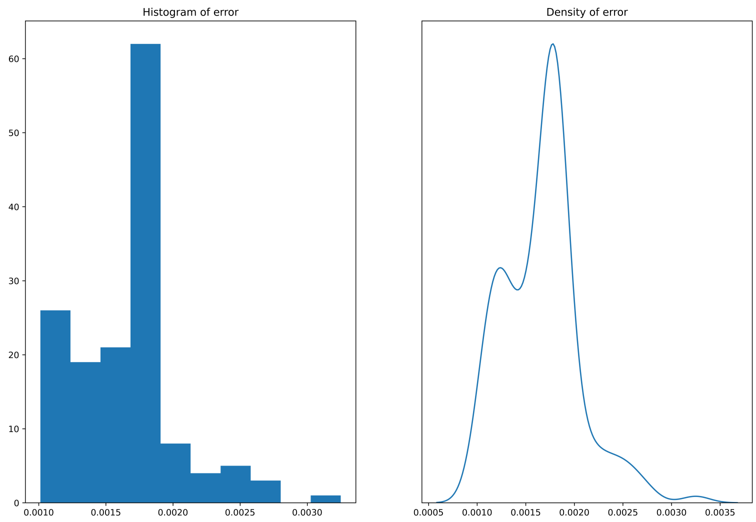

Figure 9. Histogram and density of error.

As can be seen, the error almost follows the log-normal distribution, although the range of this distribution is minimal. Although the error follows a log-normal distribution, the error domain is concise. From the error diagram of numerical results that are shown in Figure 6, Figure 8, and Figure 9, one may conclude that the error is almost constant. The stability of the error in this process may be attributed to the machine learning modeling technique based on step 4 of the algorithm described in the preceding section, which uses differential equations. From a different perspective, the constant value of the error value might suggest that this algorithm cannot determine the proper bias parameter, which can be seen as a drawback of this approach.

5. Conclusions

In this paper, we present a new method for predicting bitcoin volatility. Predictive models in this field (such as the GARCH method) may not work well in various situations. Also, techniques such as inverse problems require large amounts of data that may not be easily possible, and inverse problems generally are not used for forecasting. According to this paper's approach, the presented method's flexibility is such that it does not encounter these problems. In the first step of this method, a historical volatility dataset was generated using the market data and the actual value of future prices in the market. In the next step, we obtained an ODE for volatility considering inverse problem modeling, and in the end, since the ODE had an unknown part, we solved it by ANN using the considered dataset. Finally, we showed the results using Python. As can be seen in the numerical results, 200 days of data have been used, and the next 50 days are predicted. Since the volatility value is less than zero, the relative error is used, presenting that the accuracy of test data is 10^{-2}. Comparing this method with other methods, the changes in the output value of our model are parallel to the actual value, and the error value is almost constant. By considering this feature, we can combine this method with reinforcement learning for further study, improve the algorithm, and obtain better results.

Conflict of interest

The author declares no conflict of interest.

References

1. A. H. Dyhrberg, Hedging capabilities of bitcoin, Is it the virtual gold? *Financ. Res. Lett.*, **16** (2016), 139–144. https://doi.org/10.1016/j.frl.2015.10.025
2. T. G. Andersen, T. G. Bollerslev, F. X. Diebold, P. Labys, Modeling and forecasting realized volatility, *Econometrica*, **71** (2003), 579–625. https://doi.org/10.1111/1468-0262.00418
3. T. G. Andersen, T. Bollerslev, N. Meddahi, Correcting the errors: Volatility forecast evaluation using high-frequency data and realized volatilities, *Econometrica*, **73** (2005), 279–296. https://doi.org/10.1111/j.1468-0262.2005.00572.x
4. S. P. Azizi, A. Neisy, A new approach in geometric brownian motion model, In: Cao, BY. Eds, Fuzzy Information and Engineering and Decision, IWDS 2016, *Adv. Intell. Syst. Comput.*, **646** (2018), 336–342. https://doi.org/10.1007/978-3-319-66514-6_34
5. W. Bao, J. Yue, Y. Rao, A deep learning framework for financial time series using stacked autoencoders and long-short term memory, *PloS One*, **12** (2017), e0180944. https://doi.org/10.1371/journal.pone.0180944
6. R. Becker, A. E. Clements, M. B. Doolan, A. S. Hurn, Selecting volatility forecasting models for portfolio allocation purposes, *Int. J. Forecasting*, **31** (2015), 849–861. https://doi.org/10.1016/j.ijforecast.2013.11.007
7. T. Björk, *Arbitrage theory in continuous time*, Oxford university press, 2009. https://doi.org/10.1093/0199271267.001.0001
8. F. Black, M. Scholes, The pricing of options and corporate liabilities, *J. Polit. Econ.*, **81** (1973), 637–654. https://www.jstor.org/stable/1831029
9. T. Bollerslev, B. Hood, J. Huss, L. H. Pedersen, Risk everywhere: Modeling and managing volatility, *Rev. Financ. Stud.*, **31** (2018), 2729–2773. http://dx.doi.org/10.2139/ssrn.2722591
10. T. Bollerslev, A. J. Patton, R. Quaedvlieg, Exploiting the errors: A simple approach for improved volatility forecasting, *J. Econ.*, **192** (2016), 1–18. https://doi.org/10.1016/j.jeconom.2015.10.007
11. J. Bouoiyour, R. Selmi, Bitcoin price: Is it really that new round of volatility can be on way?, *University Library of Munich, Germany*, (2015). Available from: https://mpra.ub.uni-muenchen.de/65580/
12. J. Bouoiyour, R. Selmi, Bitcoin: A beginning of a new phase, *Econ. B.*, **36** (2016), 1430–1440. Available from: https://econpapers.repec.org/article/eblecbull/eb-16-00372.htm
13. E. Bouri, G. Azzi, A. H. Dyhrberg, On the return-volatility relationship in the Bitcoin market around the price crash of 2013, *Economics*, **11** (2017). https://doi.org/10.5018/economics-ejournal.ja.2017-2
14. Á. Cebrián-Hernández, E. Jiménez-Rodríguez, Modeling of the bitcoin volatility through key financial environment variables: An application of conditional correlation MGARCH models, *Mathematics*, **9** (2021), 267. https://doi.org/10.3390/math9030267

15. A. Charles, O. Darné, Volatility estimation for Bitcoin: Replication and robustness, *Int. Econ.*, **157** (2019), 23–32. https://doi.org/10.1016/j.inteco.2018.06.004

16. C. Chiarella, M. Craddock, N. El-Hassan, The calibration of stock option pricing models using inverse problem methodology, *QFRQ Research Papers, UTS Sydney*, **39** (2000). Available from: https://ideas.repec.org/p/uts/rpaper/39.html

17. F. Corsi, A simple approximate long-memory model of realized volatility, *J. Financ. Econ.*, **7** (2009), 174–196. https://doi.org/10.1093/jjfinec/nbp001

18. F. Corsi, D. Pirino, R. Reno, Threshold bipower variation and the impact of jumps on volatility forecasting, *J. Econ.*, **159** (2010), 276–288. https://doi.org/10.1016/j.jeconom.2010.07.008

19. G. Cortazar, E. S. Schwartz, Implementing a stochastic model for oil futures prices, *Energy Econ.*, **25** (2003), 215–238. https://doi.org/10.1016/S0140-9883(02)00096-8

20. P. Della Corte, T. Ramadorai, L. Sarno, Volatility risk premia and exchange rate predictability, *J. Financ. Econ.*, **120** (2016), 21–40. https://doi.org/10.1016/j.jfineco.2016.02.015

21. C. Dritsaki, An empirical evaluation in GARCH volatility modeling: Evidence from the Stockholm stock exchange, *J. Math. Financ.*, **7** (2017), 366–390. https://doi.org/10.4236/jmf.2017.72020

22. D. J. Duffy, *Finite Difference methods in financial engineering: A Partial Differential Equation approach*, John Wiley & Sons, 2013. https://doi.org/10.1002/9781118673447

23. A. H. Dyhrberg, Bitcoin, gold and the dollar–A GARCH volatility analysis, *Financ. Res. Lett.*, **16** (2016), 85–92. https://doi.org/10.1016/j.frl.2015.10.008

24. R. F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation, *Econometrica: J. Econ. Soc.*, **50** (1982), 987–1007. https://doi.org/10.2307/1912773

25. C. Francq, J. M. Zakoian, *GARCH models: Structure, statistical inference and financial applications*, John Wiley & Sons, 2019. ISBN: 978-1-119-31348-9.

26. G. M. Gallo, E. Otranto, Forecasting realized volatility with changing average levels, *Int. J. Forecasting*, **31** (2015), 620–634. https://doi.org/10.1016/j.ijforecast.2014.09.005

27. F. Glaser, K. Zimmermann, M. Haferkorn, M. C. Weber, M. Siering, Bitcoin-asset or currency? revealing users' hidden intentions, *Revealing Users' Hidden Intentions*, (2014) ECIS. Available from: https://ssrn.com/abstract=2425247

28. X. Gong, B. Lin, Structural breaks and volatility forecasting in the copper futures market, *J. Futures Markets*, **38** (2018), 290–339. https://doi.org/10.1002/fut.21867

29. C. Gouriéroux, *ARCH models and financial applications*, Springer Science & Business Media (2012). https://doi.org/10.1007/978-1-4612-1860-9

30. A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24797-2

31. M. Grigoriu, *Stochastic calculus: applications in science and engineering*, Birkhäuser Boston, MA, Springer Science & Business Media (2013). https://doi.org/10.1007/978-0-8176-8228-6

32. M. Gronwald, The Economics of Bitcoins – Market Characteristics and Price Jumps, *CESifo Working Paper Series*, **5121** (2014). http://dx.doi.org/10.2139/ssrn.2548999

33. T. Guo, A. Bifet, N. Antulov-Fantulin, Bitcoin Volatility Forecasting with a Glimpse into Buy and Sell Orders, *2018 IEEE International Conference on Data Mining (ICDM)*, Singapore, (2018), 989–994, http://dx.doi: 10.1109/ICDM.2018.00123

34. L. T. Hoang, D. G. Baur, Forecasting bitcoin volatility: Evidence from the options market, *J. Futures Markets*, **40** (2020), 1584–1602. https://doi.org/10.1002/fut.22144

35. J. N. Hugonnier, The Feynman–Kac formula and pricing occupation time derivatives, *Int. J. Theor. Appl. Fin.*, **2** (1999), 153–178. https://doi.org/10.1142/S021902499900011X

36. S. Karaoglu, U. Arpaci, S. Ayvaz, A deep learning approach for optimization of systematic signal detection in financial trading systems with big data, *Int. J. Intell. Syst. Appl. Eng.*, **7** (20119), 31–36. https://doi.org/10.18201/ijisae.2017SpecialIssue31421

37. P. Katsiampa, Volatility estimation for Bitcoin: A comparison of GARCH models, *Econ. Lett.*, **158** (2017), 3–6. https://doi.org/10.1016/j.econlet.2017.06.023

38. A. Kirsch, *An introduction to the mathematical theory of inverse problems*, Springer New York, NY (2011). https://doi.org/10.1007/978-1-4419-8474-6

39. F. C. Klebaner, *Introduction to stochastic calculus with applications*, World Scientific Publishing Company, (2012). https://doi.org/10.1142/p821

40. R. Lagnado, S. Osher, A technique for calibrating derivative security pricing models: numerical solution of an inverse problem, *J. Comput. Finance*, **1** (1997), 13–26. https://doi.org/10.1088/0266-5611/15/3/201

41. B. Lei, Z. Liu, Y. Song, On Stock Volatility Forecasting Based on Text Mining and Deep Learning Under High Frequency Data, *J. Forecasting*, **40** (2021), 1596–1610. https://doi.org/10.1002/for.2794

42. G. Levy, *Computational finance: numerical methods for pricing financial instruments*, Butterworth-Heinemann, (2004). ISBN: 9780080472270.

43. F. Ma, Y. Wei, L. Liu, D. Huang, Forecasting realized volatility of oil futures market: A new insight, *J. Forecasting*, **37** (2018), 419–436. https://doi.org/10.1002/for.2511

44. R. C. Merton, An intertemporal capital asset pricing model, *Econometrica: J. Econ. Soc.*, **41** (1973), 867–887. https://doi.org/10.2307/1913811

45. A. Neisy, K. Salmani, An inverse finance problem for estimation of the volatility, *Comp. Math. Math. Phys.*, **53** (2013), 63–77. https://doi.org/10.1134/S0965542513010090

46. R. Paschke, M. Prokopczuk Commodity derivatives valuation with autoregressive and moving average components in the price dynamics, *J. Banking Financ.*, **34** (2010), 2742–2752. https://doi.org/10.1016/j.jbankfin.2010.05.010

47. A. J. Patton, Volatility forecast comparison using imperfect volatility proxies, *J. Econometrics*, **160** (2011), 246–256. https://doi.org/10.1016/j.jeconom.2010.03.034

48. S. H. Poon, C. W. Granger, Forecasting volatility in financial markets: A review, *J. Econ. Lit.*, **41** (2003), 478–539. https://doi.org/10.1257/002205103765762743

49. S. Shreve, *Stochastic calculus for finance I: the binomial asset pricing model*, Springer Science Business Media, (2005). https://doi.org/10.1007/978-0-387-22527-2

50. S. E. Shreve, *Stochastic calculus for finance II: Continuous-time models*, Springer Science & Business Media, (2004). Available from: https://link.springer.com/book/9780387401010

51. The Euler-Lagrange Equation, https://www.ucl.ac.uk .

52. Y. Wang, H. Wang, S. Zhang, Prediction of daily PM2. 5 concentration in China using data-driven ordinary differential equations, *Appl. Math. Comput.*, **375** (2020), 125088. https://doi.org/10.1016/j.amc.2020.125088

53. R. Weinstock, *Calculus of variations: with applications to physics and engineering. Courier Corporation*, McGraw-Hill, (1974). ISBN: 978-0486630694.

54. J. Wloka, *Partial differential equations*, Cambridge University Press, Cambridge, UK (1987). https://doi.org/10.1017/CBO9781139171755

55. Z. Xu, X. Jia, The calibration of volatility for option pricing models with jump diffusion processes, *Appl. Anal.*, **98** (2017), 810–827. https://doi.org/10.1080/00036811.2017.1403588

56. Y. Yu, W. Duan, Q. Cao, The impact of social and conventional media on firm equity value: A sentiment analysis approach, *Decis. Support Syst.*, **55** (2013), 919–926. https://doi.org/10.1016/j.dss.2012.12.028