



Article

# GJR-GARCH Volatility Modeling under NIG and ANN for Predicting Top Cryptocurrencies

Fahad Mostafa <sup>1,\*</sup>, Pritam Saha <sup>2</sup>, Mohammad Rafiqul Islam <sup>3</sup> and Nguyet Nguyen <sup>4,\*</sup>

<sup>1</sup> Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409, USA

<sup>2</sup> Rawls College of Business, Texas Tech University, Lubbock, TX 79409, USA; pritam.saha@ttu.edu

<sup>3</sup> Department of Mathematics, Florida State University, Tallahassee, FL 32306, USA; mislam@math.fsu.edu

<sup>4</sup> Department of Mathematics and Statistics, Youngstown State University, Youngstown, OH 44555, USA

\* Correspondence: fahad.mostafa@ttu.edu (F.M.); ntnguyen01@ysu.edu (N.N.)

**Abstract:** Cryptocurrencies are currently traded worldwide, with hundreds of different currencies in existence and even more on the way. This study implements some statistical and machine learning approaches for cryptocurrency investments. First, we implement *GJR-GARCH* over the *GARCH* model to estimate the volatility of ten popular cryptocurrencies based on market capitalization: Bitcoin, Bitcoin Cash, Bitcoin SV, Chainlink, EOS, Ethereum, Litecoin, TETHER, Tezos, and XRP. Then, we use Monte Carlo simulations to generate the conditional variance of the cryptocurrencies using the *GJR-GARCH* model, and calculate the value at risk (VaR) of the simulations. We also estimate the tail-risk using VaR backtesting. Finally, we use an artificial neural network (ANN) for predicting the prices of the ten cryptocurrencies. The graphical analysis and mean square errors (MSEs) from the ANN models confirmed that the predicted prices are close to the market prices. For some cryptocurrencies, the ANN models perform better than traditional *ARIMA* models.

**Keywords:** artificial neural network; cryptocurrency; *GJR-GARCH*; *NIG*; Monte Carlo simulation; value at risk backtesting



**Citation:** Mostafa, Fahad, Pritam Saha, Mohammad Rafiqul Islam, and Nguyet Nguyen. 2021. GJR-GARCH Volatility Modeling under NIG and ANN for Predicting Top Cryptocurrencies. *Journal of Risk and Financial Management* 14: 421. <https://doi.org/10.3390/jrfm14090421>

Academic Editor: Jong-Min Kim

Received: 2 August 2021

Accepted: 25 August 2021

Published: 3 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A cryptocurrency is a digital currency and is used as a medium of exchange. This currency is a unique combination of three different attributes: anonymity, freedom from central authority, and provision of protection against double spending attacks; Lánský (2016). Cryptocurrency, also known as crypto, is one of the fastest-growing asset classes, with a total market capitalization of \$761.46 billion as of December 2020.<sup>1</sup> The underlying reason is that the market for cryptocurrency offers hedging properties when included with traditional asset classes such as stocks and bonds; Wang et al. (2019). The hedging benefit is more pronounced in developed markets; Wang et al. (2019). Moreover, some prior studies extensively studied only Bitcoin for hedging and diversification benefits, such as the fact that Bitcoin is appealing for diversification purposes Shahzad et al. (2021); Bitcoin is a strong hedge and safe-haven against some categorical economic policy uncertainty metrics, including fiscal policy, taxes, national security, and trade policy under bullish market conditions Mokni et al. (2021); Bitcoin is a good hedging instrument during high uncertainty periods Mokni et al. (2020); Bitcoin can provide diversification benefits while optimizing the risk and return with a new approach Hatemi-J et al. (2019); Bitcoin is a good hedge against both newspaper and internet-search-based measures of economic uncertainty Bouri and Gupta (2019). It is a digital or virtual currency with no regulatory council and, to date, without any government interventions, leading to a risk factor for cryptocurrency investment. Moreover, there are no indices to measure the volatility of cryptocurrency return. Therefore, it is essential to understand the risk–return properties of cryptocurrency. Besides, it is necessary to predict future prices for investors. The main objective of this study is to offer an appropriate model for estimating volatility and predicting the price

of cryptocurrency as the risk–return tradeoff of cryptocurrency is distinct. It can only be explained by cryptocurrency market-specific factors [Liu and Tsyvinski \(2018\)](#).

The prior literature has widely studied volatility and price prediction using different models; however, these studies mostly include asset classes other than cryptocurrency. [Borri \(2019\)](#) examines the volatility of four cryptocurrencies using *GARCH* models. However, one of the key caveats of estimating volatility for cryptocurrency is to account for fat-tails. As a result, we apply the *GJR-GARCH* model against the *GARCH* model in our study. Besides, there has been limited research in price prediction for cryptocurrency. Therefore, we apply artificial neural networks (*ANNs*) for predicting the price of the cryptocurrency.

In this study, we assess risk and forecast price using *ANNs* for ten cryptocurrencies selected based on market capitalization. Our sample period is from 11 November 2018 to 31 December 2020. Initially, we identify an appropriate time series model, *GJR-GARCH* [Glosten et al. \(1993\)](#), to forecast volatility and estimate VaR, which allows us to identify more accurate risk than the *GARCH* [Bollerslev \(1986\)](#) model. The existing literature provides evidence that cryptocurrencies suffer from extreme tail-risk [Borri \(2019\)](#); [Feng et al. \(2018\)](#). Using the *GJR-GARCH* model at a 1% confidence level, we confirm that cryptocurrencies suffer from tail-risk and show that Bitcoin SV and Chainlink have the highest tail-risks of  $-19.78\%$  and  $-15.57\%$ , respectively. In contrast, TETHER suffers from the least tail-risk among our selected cryptocurrencies. Value at risk (VaR) backtesting [Danielsson \(2011\)](#) is a statistical risk managing tool that helps investors to monitor and quantify the risk level associated with an investment portfolio. We perform VaR [Nieppola \(2009\)](#) backtesting to examine the accuracy of the VaR models. Then, we apply *ANNs* [McCulloch and Pitts \(1943\)](#), which are considered one of the most promising methods for time series prediction, to predict price in a complex environment. Most prior studies implement *ANNs* in predicting stock prices [Hassan et al. \(2007\)](#); [White \(1988\)](#). Some other studies implement this new technique in predicting stock indices [Yao et al. \(1999\)](#) and the credit rating process [Hájek \(2011\)](#). We show that *ANNs* can forecast next-day cryptocurrency prices with higher precision.

In this paper, we try to answer three essential questions for cryptocurrency investment: (1) Does *GJR-GARCH* perform better in estimating volatility than the *GARCH* model? (2) Does VaR Backtesting show any interesting results for top cryptocurrencies? (3) Does an *ANN* perform better than traditional *ARIMA* models for predicting prices of cryptocurrencies? This study contributes to the growing literature of cryptocurrencies as follows. First, the study contributes to tail-risk literature by providing a well-established methodology, the *GJR-GARCH* model. Second, we estimate confidence intervals for risk and returns and compare them with simulated data by Markov Chain Monte Carlo. Third, value at risk backtesting is used to measure the accuracy of the value at risk calculations, which determines how well the mentioned cryptocurrency investment strategy would perform using historical data. Finally, this study contributes to the literature of *ANNs* by showing the application in predicting prices of cryptocurrencies.

## 2. Literature Review

There has been a sufficient amount of research regarding measuring risk in finance. The *GARCH* model has been widely implemented to estimate volatility in different types of security markets. [Moschini and Myers \(2002\)](#) developed a multivariate generalized *ARCH* (*GARCH*) parameterization for the optimal futures hedge ratio using corn prices. [Alberg et al. \(2008\)](#) implemented different *GARCH* models in the Tel Aviv stock exchange to estimate conditional variance. [De Goeij and Marquering \(2004\)](#) found strong evidence of conditional heteroskedasticity in the covariance between stock and bond market returns using the *GARCH* model. However, the *GARCH* model may not be appropriate if the asset returns suffer from fat-tail risk. In such cases, the *GJR-GARCH* model is more suited as it accounts for fat-tail distribution better than the *GARCH* model. Therefore, many other studies implement the *GJR-GARCH* models. [Ma et al. \(2020\)](#) examined whether gold can be

used as a safe haven for stocks. They used the *GJR-GARCH* model to estimate the extreme tail risk. Moreover, [Maciel \(2013\)](#) applied a similar methodology in the stock market. As cryptocurrencies suffer from fat-tail risk, we prefer to implement the *GJR-GARCH* model to accurately estimate the volatility of cryptocurrencies.

However, price predictions were mainly used for the stock market. The first significant study of neural network models used IBM's daily stock returns for prediction [White \(1988\)](#). After that, significant research was performed to check neural networks' accuracy of prediction to forecast the stock market. [Hassan et al. \(2007\)](#) proposed a fusion model by combining the hidden Markov model (*HMM*), an artificial neural network (*ANN*), and genetic algorithms (*GA*) to forecast financial market behavior. [Adebiyi et al. \(2014\)](#) compared the forecasting performance by *ARIMA* and artificial neural networks for stock data. Some other studies used *ANNs* to predict the stock market index [Moghaddam et al. \(2016\)](#); [Yao et al. \(1999\)](#). Other than predicting stock prices, some studies implemented *ANNs* in different fields of finance. Previous studies have shown that *ANNs* exhibit better performance in bankruptcy prediction [Quah and Srinivasan \(1999\)](#) and the credit rating process [Hájek \(2011\)](#). Previous studies have applied different methodologies such as ordinary least squares (*OLS*) regression, support vector regression (*SVR*), and least absolute shrinkage and selection operator (*LASSO*) techniques from the field of machine learning to predict the price of cryptocurrency [Plakandaras et al. \(2021\)](#). Therefore, we present *ANNs* as an alternative mechanism for predicting cryptocurrency prices. Motivated by prior studies, we implement the *ANN* models in a relatively new financial market, which is cryptocurrency.

### 3. Methodology

In this section, we will explain data selections and methods that have been used to analyze the data. First, we describe the data source, how the cryptocurrencies were selected, and some properties of the data. Second, we describe the *ARIMA* process and its subsection describes robust model selection. Third, we present the *GJR-GARCH* and *GARCH* for volatility, and perform the back testing of the models. Finally, we describe how to use the supervised machine learning approach, the *ANN* model, for predicting cryptocurrency prices.

#### 3.1. Data Selections

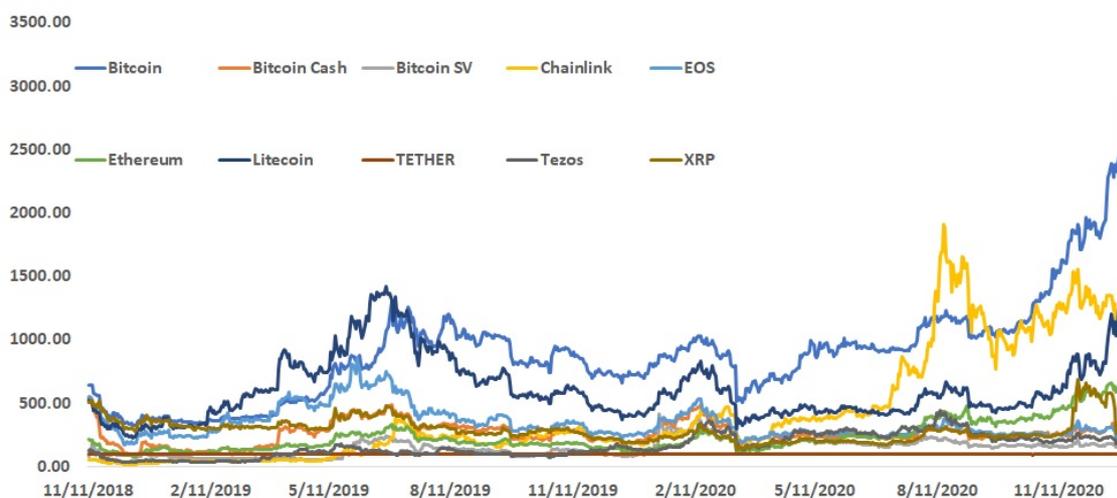
We collected daily prices, dollar volume, and market capitalization for 2500 cryptocurrencies from [coingecko.com](#) (accessed on 31 January 2021). Then, we sorted all the cryptocurrencies based on their market capitalization. Finally, we chose the top ten cryptocurrencies from the sorted list. The top ten selected cryptocurrencies were Bitcoin, Bitcoin Cash, Bitcoin SV, Chainlink, EOS, Ethereum, Litecoin, TETHER, Tezos, and XRP. The combined market capitalization of our selected cryptocurrencies covers around 80 percent of the coingecko-based market. Our sample period ran from 11 November 2018 to 31 December 2020 with a total of 781 trading days. Since the starting dates of our selected cryptocurrencies were not same, we chose a starting date of 11 November 2018, which was the first trading date of the latest cryptocurrency of our observed cryptocurrencies. Typically, we have 252 business days per year, however a cryptocurrency investor can invest and exercise every single day of the year; that is 365 days a year.

##### 3.1.1. Exploratory Data Analysis

Exploratory data analysis has been applied to summarize the main characteristics of cryptocurrencies. We plotted prices and a table of summary statistics to estimate the quick description of the data.

In [Figure 1](#), prices are scaled as follows: Bitcoin is divided by 10; XRP is multiplied by 1000; Chainlink, EOS, TETHER and Tezos are multiplied by 100; Litecoin is multiplied by 10. Data are daily prices of the cryptocurrencies from 11 November 2018 to 31 December 2020. [Figure 1](#) demonstrates a clear surge in the prices of the cryptocurrencies. In this

Figure, we see that most of the cryptocurrencies have either an upward or a downward trend except for Chainlink, TETHER, and XRP, which seem less volatile.



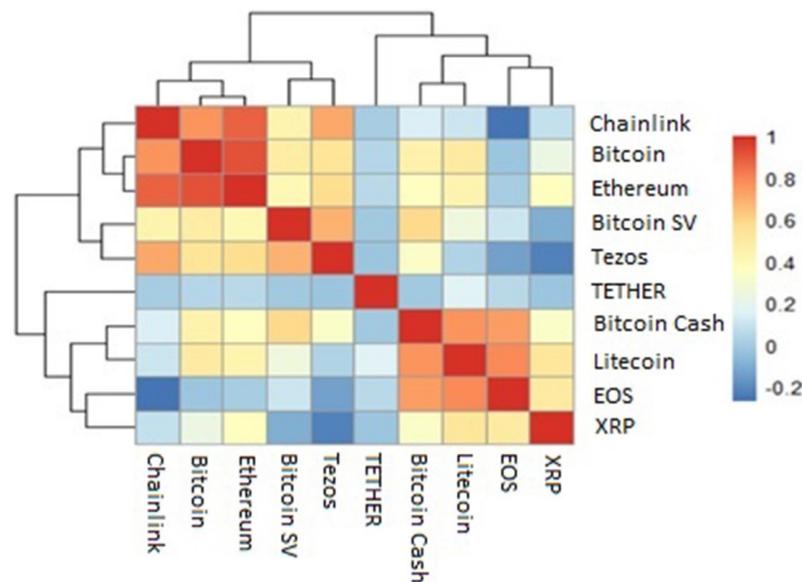
**Figure 1.** This figure plots daily prices for ten cryptocurrencies from 11 November 2018 to 31 December 2020. We scaled the prices of cryptocurrency to fit in one plot.

Table 1 presents descriptive statistics for ten cryptocurrencies where results have been calculated based on the daily log returns of the assets. The mean daily log returns of the cryptocurrencies are between  $-0.24\%$  and  $0.30\%$ . From the sample period, Chainlink observed the highest return whereas Bitcoin Cash observed the lowest return. In terms of volatility as measured by standard deviation, Bitcoin SV was the most volatile cryptocurrency whereas TETHER was the least volatile cryptocurrency.

**Table 1.** The table presents the summary statistics of the daily return of the ten cryptocurrencies. The sample period covers 11 November 2018 to 31 December 2020.

Cryptocurrency	Mean(%)	Std(%)	N	Min(%)	Max(%)
Bitcoin	0.19	3.83	781	-43.37	15.93
Bitcoin Cash	-0.06	6.10	781	-57.99	38.99
Bitcoin SV	0.09	8.53	781	-64.31	88.66
Chainlink	0.40	6.86	781	-66.08	47.61
EOS	-0.09	5.39	781	-48.87	22.90
Ethereum	0.16	4.99	781	-56.31	18.12
Litecoin	0.12	5.25	781	-47.14	26.20
TETHER	0.00	0.32	781	-1.97	2.50
Tezos	0.05	6.11	781	-62.54	27.49
XRP	-0.11	5.09	781	-54.95	34.01

We use the heatmap in Figure 2 to understand the correlation between the ten cryptocurrencies, which assists us in determining risk in further analysis. The heatmap shows that TETHER is uncorrelated with the other nine cryptocurrencies of our sample.



**Figure 2.** This figure shows the correlation among ten cryptocurrencies. Log prices of all cryptocurrencies have been used to calculate the correlation. Data are daily prices from coingecko.com. The sample period covers 11 November 2018 to 31 December 2020.

### 3.1.2. Higher Moments, Correlations, and Value at Risk

For each cryptocurrency, degrees of asymmetry in a probability distribution have been found through skewness. Because the normal distribution shows zero skewness, we use skewness to find out whether each probability distribution is normal or not. If the daily prices are skewed, a normally distributed model will always underestimate skewness risk in its predictions. The more skewed the price, the less accurate the financial model will be. Mathematically, we calculate the skewness as:

$$\tau = \frac{\mu_3}{\sigma^3} = \frac{E[(x - \mu)^3]}{E[(x - \mu)^2]^{3/2}} \tag{1}$$

Moreover, we find that kurtosis measures extreme values in either tail of a fitted distribution. This has been used to see whether the distributions with large kurtosis exhibit tail data exceeding the tails of the normal distribution. Kurtosis risks can be used to see the occasional extreme returns. Mathematically, kurtosis has been evaluated by the following formula:

$$\kappa = \frac{\mu_4}{\sigma^4} = \frac{E[(x - \mu)^4]}{E[(x - \mu)^2]^2} \tag{2}$$

where  
 $x$  = cryptocurrency returns,  
 $\mu$  = mean,  
 $\sigma$  = standard deviation.

We have found excess kurtosis because the distribution of each crypto outcome has many instances of outlier results (we also checked Q-Q plots), which causes fat tails on the bell-shaped distribution curve. Log prices of all cryptocurrencies have been used to calculate the correlation. We use the following formula:

$$R = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{3}$$

where  
 $R$  = correlation coefficient,  
 $x_i$  = values of the x-variable in a sample,

$\bar{x}$  = mean of the values of the x-variable,  
 $y_i$  = values of the y-variable in a sample,  
 $\bar{y}$  = mean of the values of the y-variable.

Value at risk, VaR, is applied on the cryptocurrency returns to see the specific losses at the left tail of the return distribution at a certain significance level. The following idea has been used to find VaR:

$$\text{VaR} = [\text{Expected Weighted Return of the Portfolio} - (\text{z-score of the confidence interval} \times \text{standard deviation of the portfolio})] \times \text{portfolio value}$$

### 3.2. ARIMA Model for Cryptocurrency Prediction

For the time series data analysis, stationarity is a very important factor. However, most of the real-world data, especially the market data (stock, cryptocurrencies, and other derivatives), are not stationary. To make the data stationary, we take the difference of the data. A non-stationary series  $X_t$  will follow an *Autoregressive Integrated Moving Average*  $ARIMA(p,d,q)$  process if  $Y_t$  follows an  $ARMA(p,q)$  process where  $Y_t$  is the  $d$ th difference of  $X_t$ ; that is,  $Y_t = \nabla^d X_t$ . A generalized  $ARIMA(p,d,q)$  model can be written as

$$(1 - \phi_p B^p - \dots - \phi_2 B^2 - \phi_1 B)(1 - B)^d Y_t = (\theta_q B^q + \dots + \theta_2 B^2 + \theta_1 B + 1)e_t \quad (4)$$

where  $B$  is the backshift operator that transforms a variable into its lagged version. For example,  $BY_t = Y_{t-1}$ ,  $B^2 Y_t = Y_{t-2}$ ,  $\dots$ ,  $B^q Y_t = Y_{t-q}$ . where  $p, q$  are the autoregression and moving average orders, respectively, which are determined from the analysis of the autocorrelation of the series.

$\phi$  and  $\theta$  are the autoregression and moving average parameters, respectively, which we estimate from the model based on the values of  $p, d$ , and  $q$ .

### Robust Model Selection

To identify the appropriate model for this study, we initially look at the AIC [Akaike \(1974\)](#) to choose which  $ARIMA(p,d,q)$  is the best fit for our prediction. This strategy has been followed for all ten cryptocurrencies. The Box–Jenkins [Box et al. \(2015\)](#) method has been applied for cryptocurrency returns to fit the ARIMA model, and we test normality by  $Q - Q$  plots and find non-normal behavior and use NIG distribution instead (see [Figure 3](#)). To consider the leverage effect, we use  $GJR-GARCH$  under NIG distribution instead of general  $GARCH$  models.

In [Figure 3](#), we only show why hyperbolic distribution and asymmetric- $GARCH$  modelling are necessary. We present results for Bitcoin only in [Figure 3](#) to avoid redundancy. For all other nine cryptocurrencies, same results have been found. The Empirical and theoretical densities have been shown where heavy tail elements have been found. Moreover, empirical and theoretical CDFs also show similar results. We also show Q-Q and P-P plots, which confirm our assumption of there being non-normality.

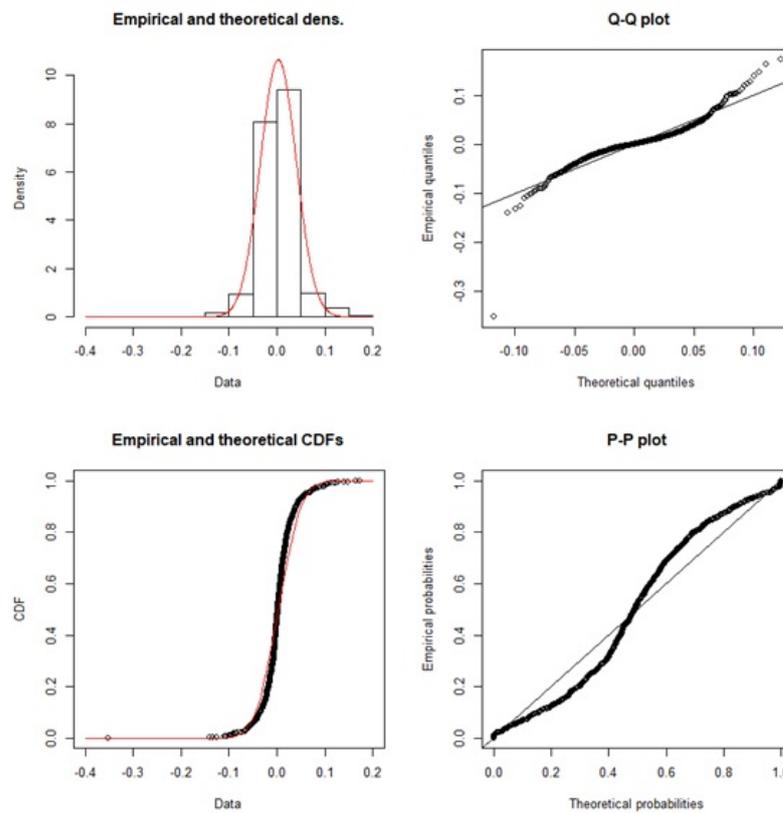


Figure 3. Comparison of Empirical and Theoretical densities.

We fit some  $ARIMA(p, d, q)$  models. Then, we compared the performance of each model in forecasting the future values. To fit these above models, we needed to make sure that the return was a stationary series. We used the Augmented Dickey Fuller (ADF) [Enders \(2008\)](#) test to check the stationarity of the models. The *null hypothesis* was “the series is not stationary”.

From [Table 2](#) we see that the p-value from the ADF test of the original series is greater than 0.05 except for TETHER and XRP, which are 0.01 and 0.013, respectively. So, at the 5% level of significance we do not have sufficient evidence against the null hypothesis that the original series is non-stationary. The null hypothesis for the KPSS test is that the data are stationary. For this test, we do not want to reject the null hypothesis. Except TETHER, all other cryptocurrencies show p-values less than or equal to 0.01 in KPSS tests.

Table 2. Tests of stationarity characteristics of each cryptocurrency.

Cryptocurrencies	ADF Test Statistics	ADF <i>p</i> -Values	Stationary Test	KPSS Level
Bitcoin	2.215	0.990	Not stationary	6.0433
Bitcoin Cash	−2.647	0.305	Not stationary	0.7578
Bitcoin SV	−2.827	0.228	Not stationary	4.4799
Chainlink	−2.325	0.441	Not stationary	8.2319
EOS	−2.297	0.453	Not stationary	2.7370
Ethereum	0.820	0.990	Not stationary	6.0329
Litecoin	−0.769	0.964	Not stationary	0.9214
TETHER	−5.514	0.010	Stationary	0.1507
Tezos	−3.054	0.132	Not stationary	8.1873
XRP	−3.910	0.013	Stationary	2.0961

As a result, we took the first and second difference depending on the data and ran different models. The best  $ARIMA(p, d, q)$  model was chosen based on criteria that the

lower AIC (Akaike’s Information Criteria) the better model as shown in the Table 3 below. However, some values of the parameters  $p, d, q$  give smaller AIC values but those are found overfitting. For example,  $ARIMA(1, 1, 2)$  has the most negative AIC value for Bitcoin but it was found that  $ARIMA(0, 2, 1)$  makes the best prediction. Similarly, all overfitted cases were excluded and the best models are shown in bold fonts (see Table 3).

**Table 3.** Choosing the best ARIMA model based on AIC values (robust model).

ARIMA(p,d,q)	ARIMA(0,2,1)	ARIMA(1,1,2)	ARIMA(0,1,1)	ARIMA(0,2,4)
Bitcoin	−2021.25	−2032.82	−2031.95	−2020.21
Bitcoin Cash	−1602.01	−1623.92	−1624.76	−1609.34
Bitcoin SV	−1300.44	−1323.13	−1318.15	−1304.74
Chainlink	−1440.84	−1454.68	−1454.35	−1442.16
EOS	−1685.25	−1707.25	−1705.49	−1692.90
Ethereum	−1723.45	−1743.80	−1738.80	−1728.94
Litecoin	−1697.86	−1711.93	−1712.14	−1700.21
TETHER	−4881.19	−5248.66	−5250.03	−5227.21
Tezos	−1512.13	−1539.72	−1534.68	−1524.31
XRP	−1628.77	−1644.95	−1641.18	−1629.69

Next, we wanted to check the ARCH effect in our data. As a result, we ran an ARCH-LM test. In Table 4 we show the results of the ARCH-LM test. The results confirm the existence of heteroskedasticity in our data. The null hypothesis in this test was that there is no ARCH effect, while an alternative effect is that there is an ARCH effect. Although we show both chi-squared values and  $p$ -values, we made the decision to reject the null hypothesis based on  $p$ -values. Therefore, ARCH effects were found in our data. The ARCH model defines the variance of the current error term or innovation as a function of the actual sizes of the error terms from prior time periods.

**Table 4.** ARCH-LM test.

Cryptocurrency	$\chi^2$	$p$ -Value
Bitcoin	770.82	$2.2 \times 10^{-16}$
Bitcoin Cash	716.47	$2.2 \times 10^{-16}$
Bitcoin SV	511.07	$2.2 \times 10^{-16}$
Chainlink	739.29	$2.2 \times 10^{-16}$
EOS	712.23	$2.2 \times 10^{-16}$
Ethereum	761.36	$2.2 \times 10^{-16}$
Litecoin	741.24	$2.2 \times 10^{-16}$
TETHER	137.97	$2.2 \times 10^{-16}$
Tezos	724.35	$2.2 \times 10^{-16}$
XRP	681.12	$2.2 \times 10^{-16}$

### 3.3. Volatility Modeling with GJR-GARCH

Let  $r_t$  denote the log return of a portfolio between periods  $t - 1$  and  $t$ , and  $F_t$  denote the information filtration generated by these terms. We have

$$\begin{aligned}
 r_t &= \mu_t + e_t \text{ where, } e_t \sim \sigma_t \epsilon_t \\
 \mathbb{E}[F_{t-1}] &= \mu_t \\
 \text{Var}(F_{t-1}) &= \sigma_t^2 \\
 \sigma_t^2 &= \alpha_0 + \sum_{m=1}^k \alpha_m e_{t-m}^2 + \sum_{n=1}^l \beta_n \sigma_{t-n}^2
 \end{aligned} \tag{5}$$

where  $\epsilon_t$  are independent and identically distributed random variables with NIG distribution. Parameter constraints are

$$\begin{aligned} &\alpha_0 > 0, \alpha_m \geq 0, \beta_n \geq 0 \\ &\text{and,} \\ &\sum_{m=1}^k \alpha_m^2 + \sum_{n=1}^l \beta_n^2 < 1 \end{aligned} \tag{6}$$

The unforeseen series can be estimated by conditional variance models. Engle (1982) discovered the Autoregressive Conditional Heteroskedasticity (ARCH) model to forecast the variance of a time series. The ARCH model assumes that, the same as the error terms in a regular AR process, the variance of the error term is dependent on previous error term variances.

$$\sigma_t^2 = \alpha_0 + \sum_{m=1}^k \alpha_m e_{t-m}^2$$

Bollerslev (1986) extended the ARCH model and developed the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model.  $GARCH(p, q)$  with three parameters can describe complex volatility structures and it is sufficient for most applications. We can forecast future volatility by using the  $GARCH(p, q)$  model below:

$$\hat{\sigma}_{t+\tau}^2 = \sigma^2 + (\alpha_1 + \beta_1)^\tau (\sigma_t^2 + \sigma^2) \tag{7}$$

where  $\sigma^2 = \frac{\alpha_0}{1-\alpha_1-\alpha_2}$  is the unconditional variance of innovations  $e_t$ .

Note that  $\alpha_1 + \beta_1 < 1$  as  $\tau \rightarrow \infty$ ; we also achieve  $\hat{\sigma}_{t+\tau}^2 \rightarrow \sigma^2$ . So, prediction of volatility goes with time asymptotically to the unconditional variance. Returns of an asset have positive excess kurtosis. Thus, we consider a fat tail,

$$\sigma_t^2 = \alpha_0 + \sum_{m=1}^k \alpha_m e_{t-m}^2 + \sum_{n=1}^l \beta_n \sigma_{t-n}^2 + \sum_{p=1}^h \gamma_p e_{t-k}^2 I_{t-k}$$

where

$$I_{t-k} = \begin{cases} 1 & \text{if } e_{t-k}^2 > 0 \\ 0 & \text{otherwise} \end{cases}$$

$\gamma_p > 0$  produces an extra volatility for negative innovation that adjusts the asymmetric impact on volatility.

The GJR-GARCH model is slightly different than the original ARCH or GARCH model. We consider GJR –  $GARCH(p, q)$  with

$$\begin{aligned} &\sum_{m=1}^k \alpha_m^2 + \sum_{n=1}^l \beta_n^2 + \frac{1}{2} \sum_{p=1}^h \gamma_p^2 < 1 \\ &\text{with,} \\ &\alpha_0 > 0, \alpha_m \geq 0, \beta_n \geq 0, \alpha_m + \gamma_p \geq 0 \\ &\hat{\sigma}_{t+\tau}^2 = \alpha_0 + \frac{\alpha_1 + \gamma_1}{2} + \beta_1 \sigma_t^2 (\tau - 1) \end{aligned} \tag{8}$$

where the leverage order is automatically considered equal to order 1, and all parameter constraints are quite similar to the  $GARCH(p, q)$  model. Barndorff-Nielsen (1977) proposed NIG distribution, a hyperbolic distribution, which captures more data than

a Gaussian distribution. The probability density function of random variable  $G$  can be defined as

$$f_{NIG}(G; \alpha, \kappa, \mu, \delta) = \frac{\alpha\delta}{\pi} e^{\delta\sqrt{\alpha^2 - \beta^2} + \beta(x - \mu)} C_1[\alpha q(x)]$$

where

$$q(x) = \left( (x - \mu)^2 + \delta^2 \right)$$

$\alpha > 0$  evaluates the shape,  $\delta > 0$  is the scaling parameter,  $\mu \in (-\infty, \infty)$  and  $0 \leq |\kappa| \leq \alpha$  evaluates the skewness.  $C_1(x)$  is the modified Bessel function of the 2nd kind with index one. We consider all innovation under *NIG*.

### 3.3.1. Confidence Bound by Markov Chain Monte Carlo Simulation

Furthermore, we consider an independent random drawn from a specific probabilistic model with realization of the entire sample path of specific length  $K, y_1, y_2, \dots, y_n$ . For a large number of simulated draws, we have  $P$  sample paths with each having length  $K$ . We use Monte Carlo simulation Hertz (1964) to generate the next conditional variance by applying *GJR-GARCH*. As the accuracy of the Monte Carlo method depends on a finite number of simulations, thus it has some errors, which can be controlled by increasing the number of paths  $M$ . We estimate the probability of the future event using the sample proportion of the event occurrence across  $M$  simulations. Here,

$$\hat{p} = \frac{C}{M} \tag{9}$$

where  $C$  is the number of time events occurring in  $M$  simulated sample paths. Finally, we estimate Monte Carlo simulated path error.

$$SE = \sqrt{\frac{\hat{p}(1 - \hat{p})}{M}}$$

We plotted the average and the 97.5% and 2.5% percentiles of the simulated paths and compared the simulation statistics to the original data.

### 3.3.2. Calculation of Sharpe Ratio

We use the Sharpe ratio for calculating the risk-adjusted returns. It is used to evaluate a portfolio's past performance (ex-post) where actual returns are used in the formula. Sharpe ratio measures the performance of an investment, which is excess return per unit of risk (measured by standard deviation) of that investment. Higher Sharpe ratios suggest better performance. The Sharpe ratio,  $S_p$ , is calculated as follows:

$$S_p = \frac{r_c - r_f}{\sigma_c} \tag{10}$$

where  $r_c$  and  $\sigma_c$  represent the return and standard deviation of cryptocurrencies and  $r_f$  presents the risk-free rates. It explains whether a cryptocurrency portfolio's excess returns are due to some smart investment decisions or a result of too much risk. Note that a negative Sharpe ratio does not present any useful meaning. For daily cryptocurrency investors, Sharpe ratio is a nice tool with which to see future returns, but it is not the only one.

### 3.3.3. Value at Risk Backtesting

Next, we examine how our ex-ante risk forecast performs in the ex-post basis. To monitor and quantify various level of risks associated with cryptocurrency investment, the VaR statistical risk management technique has been applied where backtesting measures the accuracy of the value at risk calculations. Using backtesting we determine how well a

crypto investment strategy would perform using historical data for each cryptocurrency. We obtain VaR at a 1% level, which is  $VaR_{0.01}(t + 1)$ . If  $X$  is the random variable indicating loss under given parameters  $0 < \alpha < 1$  then we can write that the  $\alpha - VaR$  of  $X$  is

$$VaR_{\alpha}(X) := -\{c : \mathbb{P}(X \leq c) \geq \alpha, \} \tag{11}$$

where  $VaR_{\alpha}(X)$  is the minimum loss that will not be exceeded with probability  $\alpha$ . The smallest loss is in the  $(1 - \alpha) \times 100\%$  worst cases. Thus, our backtesting is based on our assumed distribution at a  $(1 - \alpha)$  confidence level in VaR. We also performed Kupiec’s tests and Christoffersen’s tests for all ten cryptocurrencies, however we have not included these in our paper.

### 3.4. Supervised Machine Learning Approach: Artificial Neural Network

In ANN, some inputs are provided to an artificial neuron, and weight is associated with each input. Weight increases the steepness of the activation function. This means that weight decides how fast the activation function will trigger whereas bias  $\beta$  is used to delay the triggering of the activation function. Fundamentally, the mechanism has three layers: input, hidden, and output layers. Each layer may contain many nodes or neurons and the hidden part may also contain several layers. However, for the time series analysis and forecasting, the single-layer feed-forward network is the most widely used model structure Zhang et al. (1998). For a typical neuron, if the inputs are  $x_1, x_2, \dots, x_n$  then the synaptic weights to be applied to them are denoted as  $w_{pi}$  for  $i = 1, 2, \dots, n$ . The following equation represents the idea:

$$Y_t = W_0 + \sum_{j=1}^q W_j \cdot g \left( W_{0,j} + \sum_{i=1}^p W_{i,j} \cdot X_{t-i} \right) + \epsilon_t \tag{12}$$

where  $W_{i,j}$  and  $W_j$  for  $i = 1, 2, \dots, p, j = 1, 2, \dots, q$  are known as connection weights and  $W_0$  is the initial bias. Here  $Y_t$  is the return  $r_t$ . The weight shows the effectiveness of a particular input. The higher the weight of an input, the more impact there is on the network.

Network has been created and trained in open loop form as shown in Figure 4. In fact, open loop (single-step) is more efficient than closed loop (multi-step) training data sets. Open loop allows us to supply the network with correct past outputs as we train it to produce the correct current outputs.

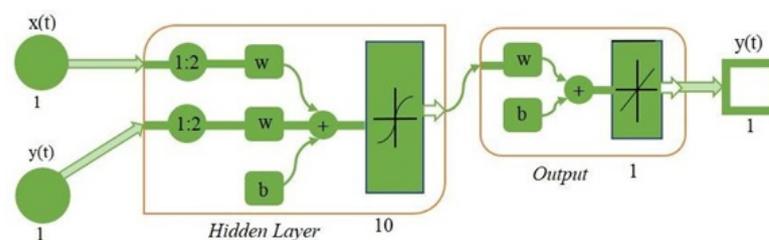


Figure 4. Architecture of artificial neural network.

The performance of the ANN Crick (1989) was evaluated using the determination of coefficient  $R^2$  and the mean square error ( $MSE$ ) of the output of the model.  $MSE$  indicates the average squared difference between the predicted values estimated from a model and the actual values. The  $MSE$  is calculated as follows based on cryptocurrency return.

$$MSE = \frac{1}{N} \sum_{t=1}^N (r_t - \hat{r}_t)^2 \tag{13}$$

where  $r_t$  and  $\hat{r}_t$  are actual and predicted prices of the cryptocurrencies. We chose the Bayesian regularization training algorithm as used in [MacKay and Mac Kay \(2003\)](#) and [Selvamuthu et al. \(2019\)](#). Training stops according to adaptive weight minimization (regularization). To examine whether our model was performing well or not, we checked mean squared error (MSE).

#### 4. Results and Discussion

In this section, we present the results of the *GJR-GARCH* and *GARCH* models for estimating the volatilities of the ten selected cryptocurrencies, simulated cryptocurrency volatility, and predicted cryptocurrency prices.

##### 4.1. GJR-GARCH and GARCH Models for Cryptocurrency Volatilities

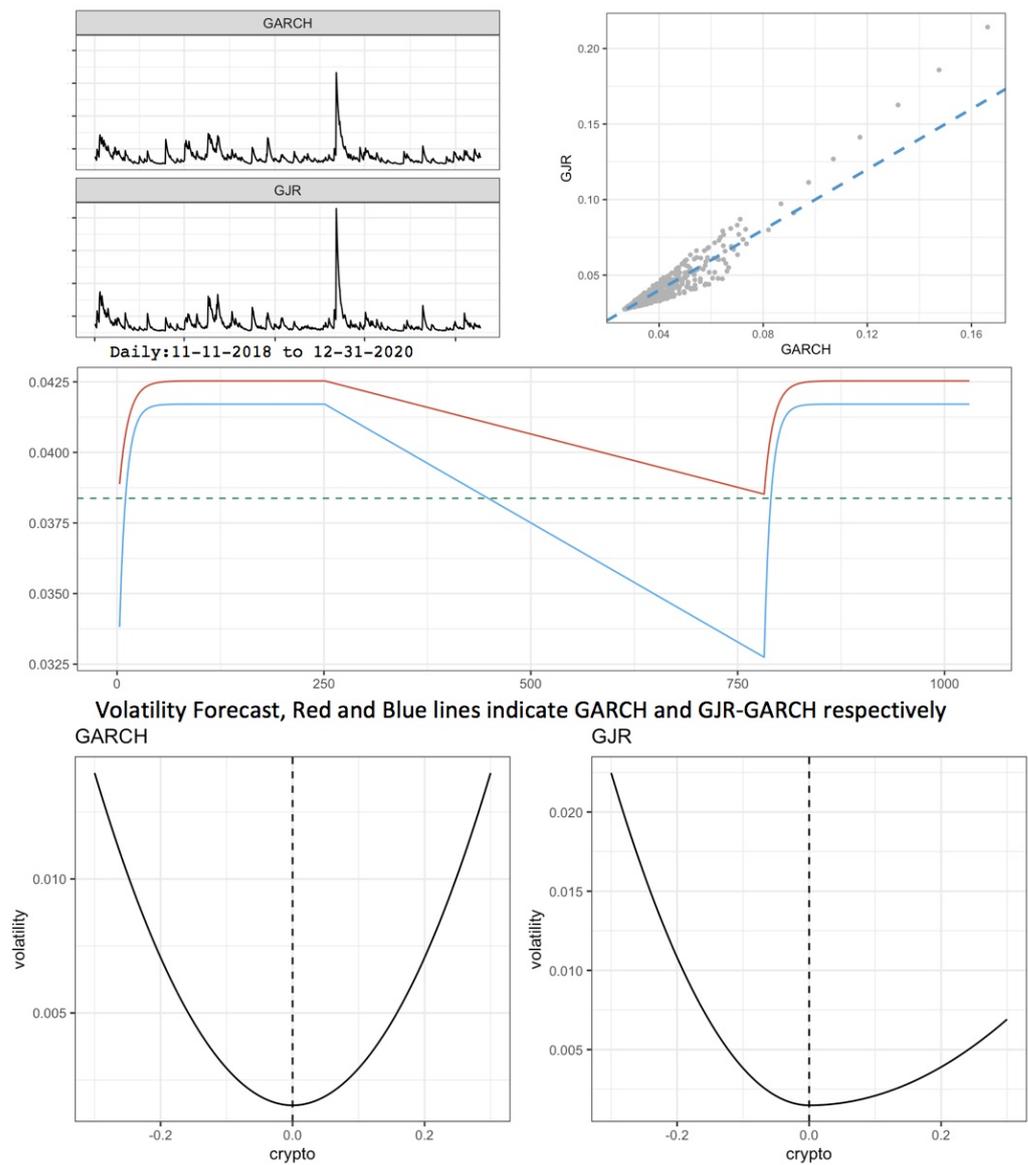
First, we present descriptive statistics of the ten cryptocurrencies in [Table 5](#). Sharpe ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk. Intuitively, it measures the excess return earned per unit of risk. In our sample, Chainlink has the highest Sharpe ratio of 0.046, whereas Bitcoin has the second-highest Sharpe ratio of 0.029 in our sample. Some cryptocurrencies have negative Sharpe ratios, which signify that their average return is less than the risk-free rate in our sample.

[Table 5](#) shows that cryptocurrency returns exhibit fat-tail distributions. Generally, prior literature has used the *GARCH* model over the autoregressive conditional heteroscedastic (*ARCH*) model. However, to control for asymmetric responses of volatility to innovation fluctuations, we used the *GJR-GARCH* model. From [Table 5](#), we observe all positive values of excess kurtosis, which means all of the returns have leptokurtic distributions. This refers to a heavy degree of risks with extreme return values. Moreover, except Bitcoin SV and TETHER, all the other cryptocurrencies show skewness with negative values, which are compensated with high future returns for higher volatility. These two higher moments play with investors' sentiments.

**Table 5.** Higher moments and value at risk ( $q = 0.01$ ).

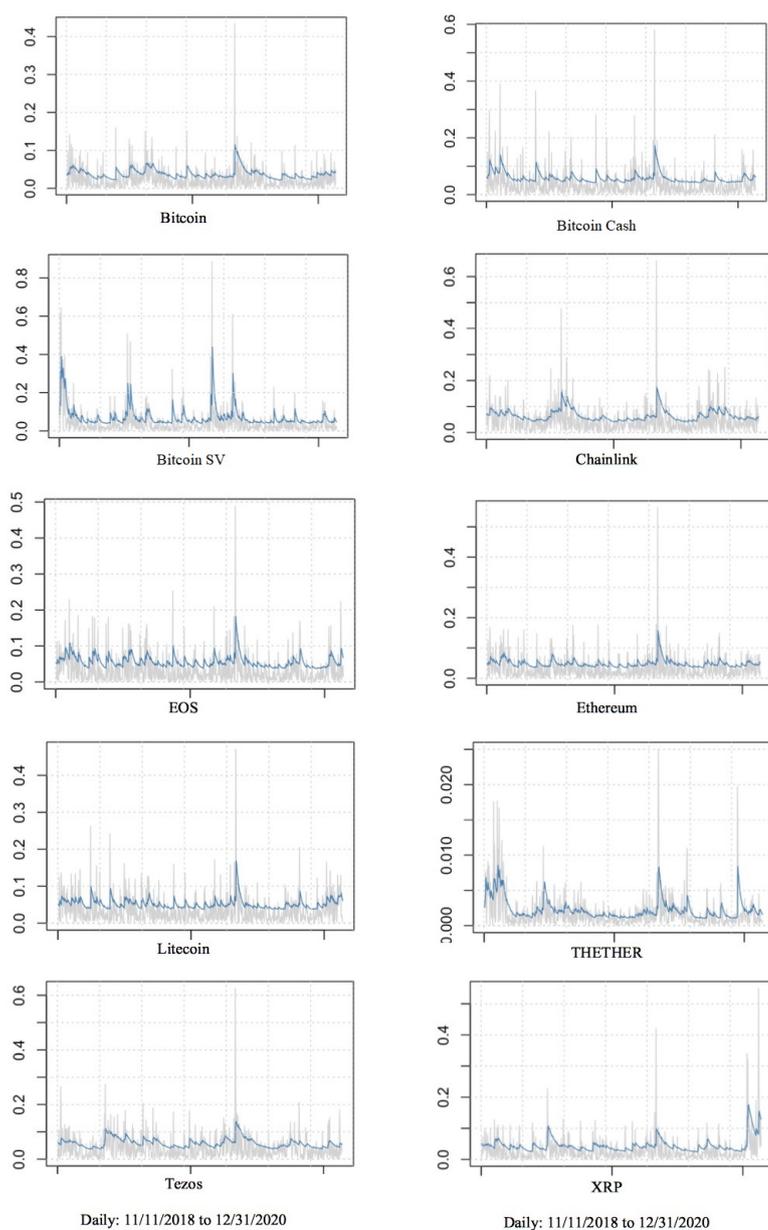
Cryptocurrency	Skewness	Kurtosis	VaR <sub>q</sub>	Sharpe Ratio
Bitcoin	−1.78	22.87	−0.087	0.029
Bitcoin Cash	−0.68	16.53	−0.143	−0.022
Bitcoin SV	1.43	29.69	−0.198	0.001
Chainlink	−0.43	14.87	−0.156	0.046
EOS	−1.00	11.02	−0.126	−0.032
Ethereum	−1.93	22.06	−0.114	0.016
Litecoin	−0.62	10.43	−0.121	0.007
TETHER	0.59	13.71	−0.007	−0.252
Tezos	−1.16	15.49	−0.142	−0.004
XRP	−1.31	28.89	−0.119	−0.038

[Figure 5](#) depicts the comparison between the *GARCH(p,q)* and *GJR-GARCH(p,q)* models to estimate volatility using daily data. The top left panel of this Figure shows the time-series and the right panel shows the scatter plot of the volatility estimate. The middle panel shows a comparison in volatility forecasting. The bottom panel shows the news impact curve. We used the daily data for Bitcoin and we found similar results for all others. Therefore, to avoid redundancy and to show volatility model selection, we only show results for a single cryptocurrency. We also use the lowest AIC value for determining the best  $p$  and  $q$  for a robust volatility *GJR-GARCH(p,q)* model for each cryptocurrency. Furthermore, there is a valid reason for choosing *GJR-GARCH* over the *GARCH* model: it is empirically found that negative cryptocurrency returns at time  $t - 1$  have a stronger impact on the volatility at time  $t$  than positive cryptocurrency returns. This asymmetric process is known as the leverage effect. The increment of the risk is realized to come from the increased leverage induced by the negative returns of cryptos [Glosten et al. \(1993\)](#).



**Figure 5.** Comparison between  $GARCH(p,q)$  and  $GJR-GARCH(p,q)$  models.

In Figure 6, we forecast the volatility for the ten cryptocurrencies. It shows daily data where the blue line indicates predicted volatility. If we examine the historical return series, it does not show conditional mean offset and thus exhibits volatility clustering.



**Figure 6.** Volatility prediction by conditional variance (vs. absolute value of returns) using GJR-GARCH model.

However, we initially forecast the volatility of Bitcoin using both *GARCH* and *GJR-GARCH* in Figure 5 and found that *GJR-GARCH* provides the better estimate. That is why we implemented the *GJR-GARCH*( $p,q$ ) model to forecast the return and volatility for our selected cryptocurrencies. In Figure 6, we forecast the conditional variance (vs. absolute value of returns) for all ten cryptocurrencies.

#### 4.2. Monte Carlo Simulations of the Cryptocurrencies' Volatility Using GJR-GARCH Model

Next, we simulated the conditional variance of the cryptocurrency's returns from a fully specified *GJR-GARCH* model based on historical data. In Figure 7, we plotted the average and the 97.5% and 2.5% percentiles of the simulated paths and compared the simulation statistics to the original data for each cryptocurrency. The red band indicates the confidence bound, and all of our selected cryptocurrencies are embedded in the simulated Monte Carlo paths. We followed the *GJR-GARCH*( $p,q$ ) model based on a 1% confidence level and daily frequency to estimate VaR. We have presented our results both in Table 5

and Figure 7. Our results show that Bitcoin, Bitcoin Cash, Ethereum, and Litecoin suffer the most from tail-risk whereas TETHER has the least tail-risk.

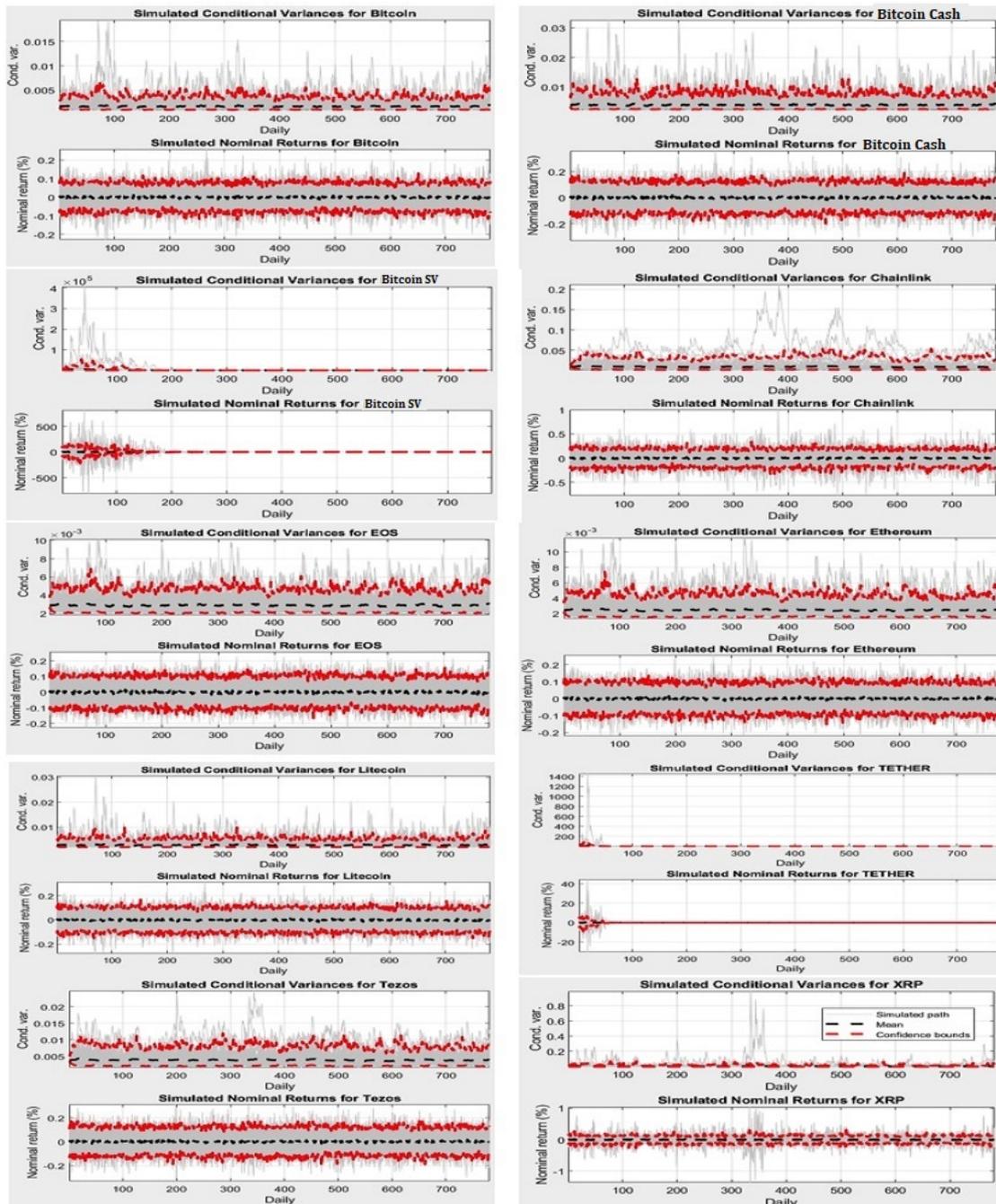
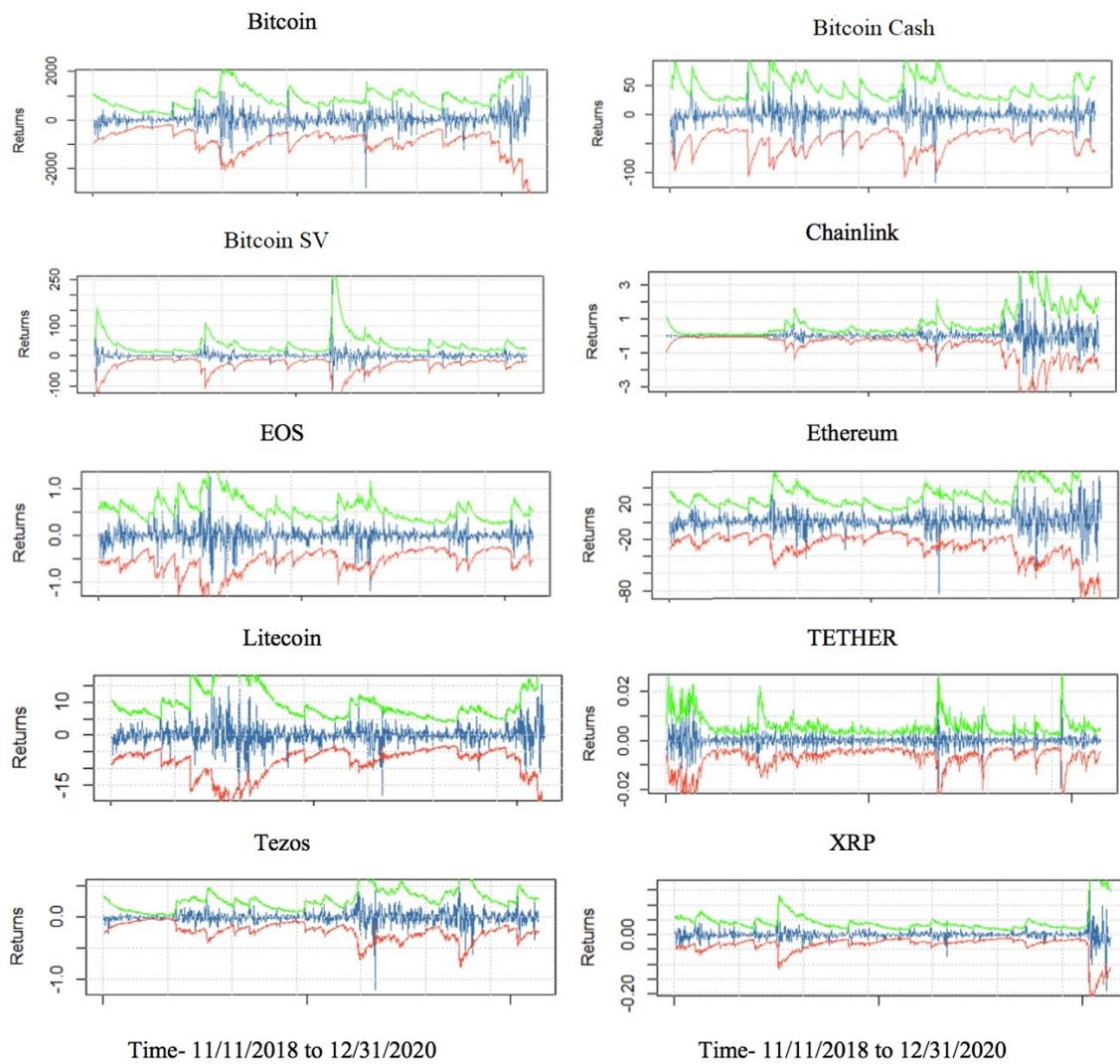


Figure 7. The average and the 97.5% and 2.5% percentiles of the simulated paths for 10 cryptocurrencies.

In Figure 8, the jagged red line running across the bottom of the plot indicates the portfolio’s (negative) one-day 99% value-at-risk. For any instance of a cryptocurrency falling below that line, there will occur an exceedance risk. This would predict a 99% value-at-risk measure to experience approximately very few exceedances in under our horizon. We observe that Bitcoin, Bitcoin Cash, EOS, Ethereum, and Litecoin are highly risky assets, because these have many exceedance of jagged lines. Referring to Figure 8, Bitcoin has at least four exceedances of jagged lines. Thus, it supports the hypothesis that Bitcoin is one of the highly volatile cryptocurrencies.



**Figure 8.** Value at risk backtesting for daily portfolio of top ten cryptocurrencies.

#### 4.3. ARIMA and ANN for Forecasting Cryptocurrencies' Prices

Figure 9 presents a window plot of the predicted price of each cryptocurrency with their actual prices. From close observation we see that ARIMA models also give higher accuracy for the next-cryptocurrency price prediction.

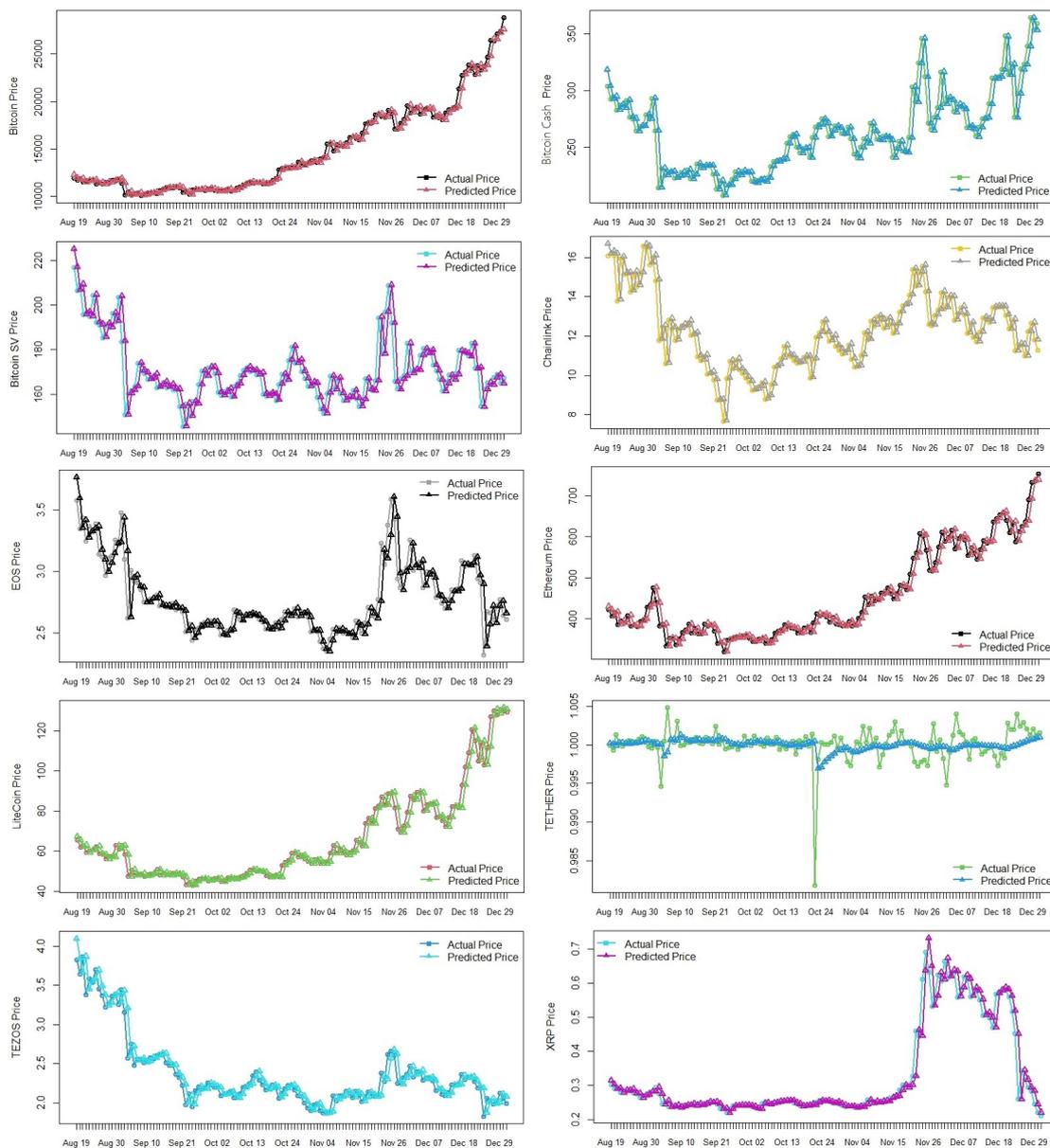


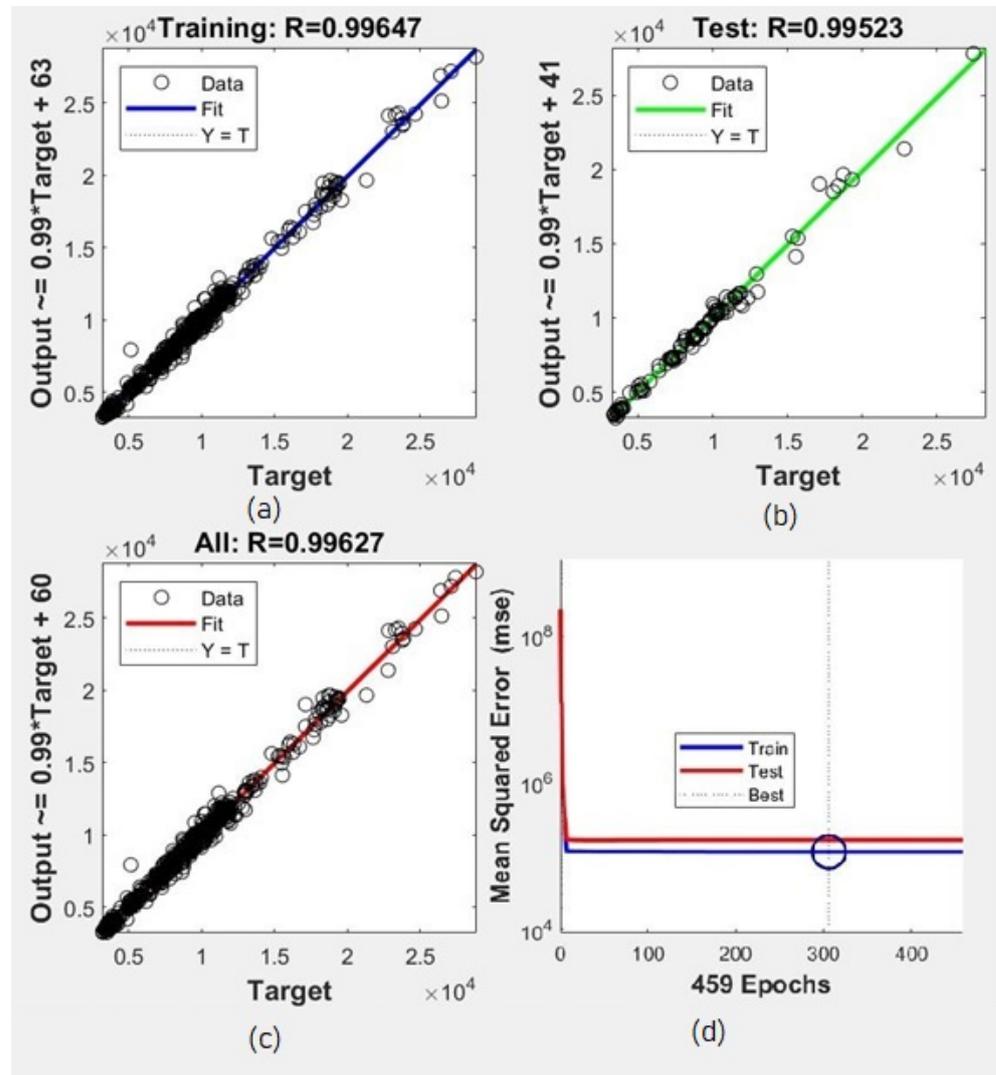
Figure 9. Predicted prices of the cryptocurrencies by the ARIMA model.

We use ANN models for forecasting the future price of ten cryptocurrencies. However, we choose to show the complete process of forecasting future prices using ANN for only Chainlink. After training, the network may be converted to closed loop form. We choose daily price data of Chainlink randomly and divide into three different category named training set (70%), testing set (15%) and Validation set (15%).

In Figure 10, we find that we don't have any overfit of the model. In Figure 10a, we evaluate regression fit by the value of regression coefficient. Figure 10b allows us to check on the training progress of our Neural Network. The Figure 10b shows MSE and epochs of the Neural Network for training and test set. To observe whether our ANN is training well, we look for training set's Loss and Accuracy and whether they converge as the number of epochs increases. As our Loss and Accuracy from both sets are diverging from each other, there is no sign of overfitting from our ANN model. It shows a good fit for Chainlink. We also observe similar output for all other nine cryptocurrencies.

In our case, we show it for 9 epochs in Figure 10c. All types of data sets are approximately overlapping at 3 epochs. Therefore, we do not have to change our ANN structure. Next, we tested our model against actual data to see how well it performed. We found that

for all of the ten cryptocurrencies, our ANN model could predict the price, which means it is performing quite well. Here Figure 10,  $R$  is actually  $R^2$ ; the coefficient of the model.



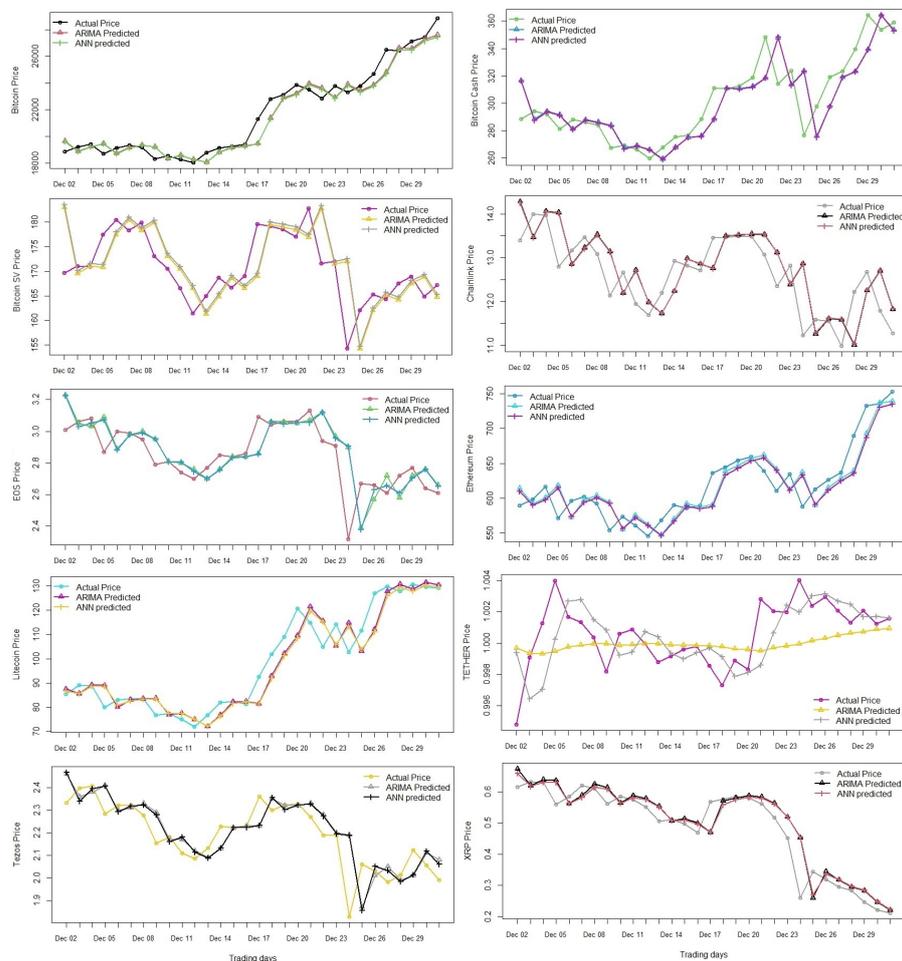
**Figure 10.** ANN model diagnostics. Validation Performance plot are displayed by the regression plots. (a) Training data set, (b) Testing data set, (c) Overall data set, and (d) Epochs vs. MSE.

Table 6 has been given for the  $MSE$  of each model. In the studied supervised machine learning technique, we evaluated the model by using  $MSE$ . For some mentioned cryptocurrencies, we obtained less  $MSE$  for ANN predictions, and for other cryptocurrencies, we see  $ARIMA$  perform better. As we know, minimal or least  $MSE$  is considered the best fit. Therefore, from this study, it is not possible to say that the ANN is the best model but one can implement the ANN as an alternative procedure. For Chainlink and all other nine cryptocurrencies, K-fold validations are compared using the mean square error ( $MSE$ ) metric. We observe that the  $MSE$  of the fitted model is less than cross-validation  $MSE$  for all ten cryptocurrencies. For our example, the cross-validation  $MSE$  of Chainlink is 0.53 approximately, which is obviously less than the  $MSE$  of  $ARIMA$  and ANN prediction.

**Table 6.** Mean square error.

Cryptocurrencies	ARIMA Prediction	ANN Prediction
Bitcoin	174,304.1	183,090.2
Bitcoin Cash	142.6201	143.4331
Bitcoin SV	60.34515	60.29052
Chainlink	0.5256038	0.5184568
EOS	0.01581617	0.01552488
Ethereum	279.0574	288.4151
LiteCoin	11.12067	11.33181
TETHER	$4.18 \times 10^{-6}$	$5.09 \times 10^{-6}$
Tezos	0.01715067	0.01733191
XRP	0.000741516	0.00071525

The window plot in Figure 11 presents only a one-month prediction output. All of the models are showing good output. In most of the cases, the ANN is performing better than ARIMA.



**Figure 11.** Comparison between market prices versus ARIMA and ANN predicted prices.

Overall, in this paper we have contributed two separate studies for cryptocurrencies. One is for volatility/risk and another is for return. An investor may think about these two properties of the financial market. Cheikh et al. (2020) investigated the presence of asymmetric volatility dynamics in Bitcoin, Ethereum, Ripple, and Litecoin using threshold GARCH models. We have extended this group of literature. We compared GARCH and GJR-GARCH volatility estimates for estimating volatility considering the normal inverse

Gaussian (*NIG*) distribution. Then, we showed that to estimate volatility for cryptocurrencies, the *GJR-GARCH* model under *NIG* is better. Similarly, for predicting prices of cryptocurrencies, we applied an *ANN* and showed the comparison with *ARIMA* instead of ordinary least squares (*OLS*) regression, support vector regression (*SVR*), and the least absolute shrinkage and selection operator (*LASSO*) of Bouri et al. (2021). Our findings have implications for money managers. Money managers can implement an *ANN* for predicting the prices of cryptocurrencies.

## 5. Conclusions

We conducted a standard learning approach for cryptocurrency investment by fitting two different time-series models, the *GARCH*( $p,q$ ) and the *GJR-GARCH*( $p,q$ ) models, in predicting cryptocurrency risks. We concluded that *GJR-GARCH*( $p,q$ ) provides a better estimate for volatility of cryptocurrency returns than *GARCH*( $p,q$ ). We also estimated the tail-risk of cryptocurrencies using VaR, and backtesting provides a comparison of the Monte Carlo simulated VaR measure to the actual volatility of cryptocurrency returns. We showed that cryptocurrencies suffer from excessive tail-risk. Finally, we presented a comparison of *ARIMA* and *ANN* for forecasting cryptocurrency prices with market values. Based on the results, the *ANN* can be considered an alternative method to the *ARIMA* model for predicting cryptocurrency prices. From our analysis we observe that Bitcoin, Bitcoin Cash, EOS, Ethereum, and Litecoin are highly risky cryptocurrencies. Although the paper focused on comparing some common financial models in evaluating cryptocurrency risk and predicting prices, it is among the few papers in the literature that have studied the cryptocurrency market. Therefore, the paper would make a good contribution to cryptocurrency risk management. This paper mainly analyzes some top cryptocurrencies for estimating risk and predicting prices by comparing two models for each objective. *GJR-GARCH* under *NIG* and *ANN* for cryptocurrency price prediction can be alternative powerful techniques. Future research may extend the analysis by fitting more models for several cryptocurrencies to estimate risk and predict prices. More specifically, we want to introduce more statistical machine learning models in cryptocurrency price prediction. We did not use any machine learning techniques for volatility prediction or portfolio optimization, such as CVaR, mean variance, or Black Litterman portfolio optimization in the case of cryptocurrency, to optimize asset allocation. In future we will contribute in this sector. Since machine learning techniques are data driven, thus they will help portfolio managers to understand the future risk of investors' assets more profoundly.

**Author Contributions:** Data Collection: P.S., and F.M., Methodology Development: F.M., P.S., and M.R.I., Writing: all authors contributed equally, Funding and overall supervision: N.N., All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** In this study, publicly available data sets were used. These data can be found here: [coinmarketcap.com, accessed on 31 January 2021]

**Acknowledgments:** The authors are grateful to Svetlozar Rachev of the Department of Mathematics and Statistics, Texas Tech University, for teaching Advanced Statistical Methods in Finance at the Summer Graduate Course. The authors are also thankful to the unknown reviewers for wonderful suggestions which make this paper more sound. Codes were written in MATLAB-R2021a under a Texas Tech University authorized licence. Some analyses were done in R, an open statistical software package.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Note

<sup>1</sup> coinmarketcap.com accessed on 31 January 2021.

## References

- Adebiyi, Ayodele Ariyo, Aderemi Oluyinka Adewumi, and Charles Korede Ayo. 2014. Comparison of arima and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics* 2014: 614342. [CrossRef]
- Akaike, Hirotugu. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19: 716–23. [CrossRef]
- Alberg, Dima, Haim Shalit, and Rami Yosef. 2008. Estimating stock market volatility using asymmetric garch models. *Applied Financial Economics* 18: 1201–8. [CrossRef]
- Barndorff-Nielsen, Ole. 1977. Exponentially decreasing distributions for the logarithm of particle size. *Proceedings of the Royal Society of London A Mathematical and Physical Sciences* 353: 401–19.
- Bollerslev, Tim. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31: 307–27. [CrossRef]
- Borri, Nicola. 2019. Conditional tail-risk in cryptocurrency markets. *Journal of Empirical Finance* 50: 1–19. [CrossRef]
- Bouri, Elie, Konstantinos Gkillas, Rangan Gupta, and Christian Pierdzioch. 2021. Forecasting realized volatility of bitcoin: The role of the trade war. *Computational Economics* 57: 29–53. [CrossRef]
- Bouri, Elie, and Rangan Gupta. 2019. Predicting bitcoin returns: Comparing the roles of newspaper-and internet search-based measures of uncertainty. *Finance Research Letters* 38: 101398. [CrossRef]
- Box, George E. P., Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. 2015. *Time Series Analysis: Forecasting and Control*. Hoboken: John Wiley & Sons.
- Cheikh, Nidhaleddine Ben, Younes Ben Zaid, and Julien Chevallier. 2020. Asymmetric volatility in cryptocurrency markets: New evidence from smooth transition garch models. *Finance Research Letters* 35: 101293. [CrossRef]
- Crick, Francis. 1989. The recent excitement about neural networks. *Nature* 337: 129–32. [CrossRef]
- Danielsson, Jon. 2011. *Financial Risk Forecasting: The Theory and Practice of Forecasting Market Risk with Implementation in R and Matlab*. New York: John Wiley & Sons, vol. 588.
- De Goeij, Peter, and Wessel Marquering. 2004. Modeling the conditional covariance between stock and bond returns: A multivariate garch approach. *Journal of Financial Econometrics* 2: 531–64. [CrossRef]
- Enders, Walter. 2008. *Applied Econometric Time Series*. New York: John Wiley & Sons.
- Engle, Robert F. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society* 50: 987–1007. [CrossRef]
- Feng, Wenjun, Yiming Wang, and Zhengjun Zhang. 2018. Can cryptocurrencies be a safe haven: A tail risk perspective analysis. *Applied Economics* 50: 4745–62. [CrossRef]
- Glosten, Lawrence R., Ravi Jagannathan, and David E. Runkle. 1993. On the relation between the expected value and the volatility of the nominal excess return on stocks. *The Journal of Finance* 48: 1779–801. [CrossRef]
- Hájek, Petr. 2011. Municipal credit rating modelling by neural networks. *Decision Support Systems* 51: 108–18. [CrossRef]
- Hassan, Md Rafiul, Baikunth Nath, and Michael Kirley. 2007. A fusion model of hmm, ann and ga for stock market forecasting. *Expert Systems with Applications* 33: 171–80. [CrossRef]
- Hatemi-J, Abdunasser, Mohamed A. Hajji, Elie Bouri, and Rangan Gupta. 2019. *The Benefits of Diversification between Bitcoin, Bonds, Equities and the Us Dollar: A Matter of Portfolio Construction*. Technical Report. Pretoria: Department of Economics, University of Pretoria.
- Hertz, David B. 1964. Risk Analysis in Capital Investment: Harvard Business Review. Available online: <https://hbr.org/1979/09/risk-analysis-in-capital-investment> (accessed on 30 March 2021).
- Lánský, Jan. 2016. Analysis of cryptocurrencies price development. *Acta Informatica Pragensia* 5: 118–37. [CrossRef]
- Liu, Yukun, and Aleh Tsyvinski. 2018. Risks and returns of cryptocurrency. *The Review of Financial Studies* 34: 2689–727. [CrossRef]
- Ma, Xiaomeng, Ruixian Yang, Dong Zou, and Rui Liu. 2020. Measuring extreme risk of sustainable financial system using gjr-garch model trading data-based. *International Journal of Information Management* 50: 526–37. [CrossRef]
- Maciel, Leandro. 2013. A hybrid fuzzy gjr-garch modeling approach for stock market volatility forecasting. In *Advances in Financial Risk Management*. Berlin: Springer, pp. 253–83.
- MacKay, David J. C., and David J. C. Mac Kay. 2003. *Information Theory, Inference and Learning Algorithms*. Cambridge: Cambridge University Press.
- McCulloch, Warren S., and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* 5: 115–33. [CrossRef]
- Moghaddam, Amin Hedayati, Moein Hedayati Moghaddam, and Morteza Esfandyari. 2016. Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science* 21: 89–93. [CrossRef]
- Mokni, Khaled, Ahdi Noomen Ajmi, Elie Bouri, and Xuan Vinh Vo. 2020. Economic policy uncertainty and the bitcoin-us stock nexus. *Journal of Multinational Financial Management* 57: 100656. [CrossRef]
- Mokni, Khaled, Elie Bouri, Ahdi Noomen Ajmi, and Xuan Vinh Vo. 2021. Does bitcoin hedge categorical economic uncertainty? A quantile analysis. *SAGE Open* 11: 21582440211016377. [CrossRef]
- Moschini, Gian Carlo, and Robert J. Myers. 2002. Testing for constant hedge ratios in commodity markets: A multivariate garch approach. *Journal of Empirical Finance* 9: 589–603. [CrossRef]
- Nieppola, Olli. 2009. *Backtesting Value-at-Risk Models, Thesis Paper: Department of Economics*. Espoo: Aalto University. Available online: <https://aaltodoc.aalto.fi/handle/123456789/181> (accessed on 30 March 2021).

- Plakandaras, Vasilios, Elie Bouri, and Rangan Gupta. 2021. Forecasting bitcoin returns: Is there a role for the us–china trade war? *Journal of Risk* 23: 3. [[CrossRef](#)]
- Quah, Tong-Seng, and Bobby Srinivasan. 1999. Improving returns on stock investment through neural network selection. *Expert Systems with Applications* 17: 295–301. [[CrossRef](#)]
- Selvamuthu, Dharmaraja, Vineet Kumar, and Abhishek Mishra. 2019. Indian stock market prediction using artificial neural networks on tick data. *Financial Innovation* 5: 16. [[CrossRef](#)]
- Shahzad, Syed Jawad Hussain, Elie Bouri, Mobeen Ur Rehman, and David Roubaud. 2021. *The Hedge Asset for Brics Stock Markets: Bitcoin, Gold, or Vix*. Kiel: The World Economy.
- Wang, Gangjin, Yanping Tang, Chi Xie, and Shou Chen. 2019. Is bitcoin a safe haven or a hedging asset? evidence from china. *Journal of Management Science and Engineering* 4: 173–88. [[CrossRef](#)]
- Wang, Pengfei, Wei Zhang, Xiao Li, and Dehua Shen. 2019. Is cryptocurrency a hedge or a safe haven for international indices? A comprehensive and dynamic perspective. *Finance Research Letters* 31: 1–18. [[CrossRef](#)]
- White, Halbert. 1988. Economic prediction using neural networks: The case of ibm daily stock returns. Paper presented at IEEE 1988 International Conference on Neural Networks, San Diego, CA, USA, July 24–27; pp. 451–8. vol. 2. [[CrossRef](#)]
- Yao, Jingtao, Chew Lim Tan, and Hean-Lee Poh. 1999. Neural networks for technical analysis: A study on klcj. *International Journal of Theoretical and Applied Finance* 2: 221–41. [[CrossRef](#)]
- Zhang, Guoqiang, B. Eddy Patuwo, and Michael Y. Hu. 1998. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting* 14: 35–62. [[CrossRef](#)]