CRYPTOCURRENCY FORECASTING WITH MACHINE LEARNING TECHNIQUES

by

Olga Diukanova

Supervisor: Nicola Rubino

A thesis submitted in conformity with the requirements
for the MSc in Economics, Finance and Computer Science

University of Huelva & International University of Andalusia

September 2023

Abstract

The main contribution of this study entails an exhaustive analysis of high frequency time series of cryptocurrencies with the largest market capitalization in 2022, namely Bitcoin, Ether and Tether through the application of a set of time series methods aimed at finding the best performing forecasting methodology.

To uncover the hidden interdependencies and interactions among these three cryptocurrencies, the persistence, cointegration, causality, mutual information, variance decomposition, transmission of impulse responses, short-term and long-terms spillover effects were evaluated. Results of GARCH-DCC models shown that high historical correlations between Bitcoin and Ether make them good candidates for pairs trading. Johansen Cointegration test allowed to devise a stationary dynamic portfolio consisting of a certain number of Bitcoin, Ether and Tether shares suggested by a cointegrating vector.

In case of trading the individual cryptocurrencies, pairs trading or portfolio trading profits depend on the correctness of forecasted price moves. For these reasons prices of each cryptocurrency were forecasted by comparing performance of the traditional linear Autoregressive integrated moving average (ARIMA) model with the Long short-term memory (LSTM), Support Vector Regression (SVR) and Random Forest regression models, that are established machine learning tools to grasp non-linear complex interactions. The forecasting accuracy of individual models was contrasted against the hybridization framework. A novel approach of hybridization suggested in this study permitted to forecast the linear portion of cryptocurrencies' time series with ARIMA, whereas the non-linear part is captured with LSTM, SVR and Random Forest models.

The results of computer simulations revealed that LSTM model ensures the highest among all models tested, at least 98.5% accuracy of forecasting for all three cryptocurrencies. Although the Hybrid Arima-Random Forest provided the highest accuracy for BTC and ETH prices, LSTM was the only model that was able to deliver high accuracy of USDT price predictions.

JEL classification: **C320, G170 ,C580, G190**

Keywords: causality, cointegration, order of integration, ARIMA, SVR, Random Forest, Hybrid modelling techniques

Contents

Figures

Tables

Introduction

Over the last decade cryptocurrencies have gained much popularity as an alternative to regular currencies. Prices of cryptocurrencies are characterized by nonlinearity, uncertainly and high volatility, which affects the predictive capacity of time series forecasting models and means high levels of risk for investors.

The majority of studies dealing with price prediction, volatility, seasonality and other trends in cryptocurrencies were focused on bitcoin that still dominates the market. Up to now cryptocurrency time series have been analyzed with a wide range of empirical models that provide a certain knowledge base about their accuracy, adequacy and appropriateness (e.g. Sebastião and Godinho (2021), Jang and Lee (2018), Liu et al (2021), McNally, Roche and Caton (2018), Kinateder and Papavassilioub (2021) and others. Nevertheless, under the constantly changing market conditions, boom-bust nature of cryptocurrency markets, emergence of new crypto coins, a search of an adequate methodology for cryptocurrency price forecasting keeps being highly relevant. Recent breakthroughs in artificial intelligence lead to wide adoption of this new technology in the financial market. Machine learning (a sub-field of artificial intelligence) gained wide recognition among individuals and financial institutions for its prediction accuracy.

This study is an example of cross-sectional research of three cryptocurrencies with the largest market capitalization in 2022, namely Bitcoin, Ethereum and Tether. The main objective of this study is to uncover the hidden interdependencies and interactions among these three cryptocurrencies and to identify the most accurate method for price forecasting. This study is organized the following way:

In the first chapter the persistence, cointegration, bivariate and multivariate causality relationships, mutual information, transmission of impulse responses, short-term and long-terms spillover effects in cryptocurrency prices and returns were evaluated. Knowing how the cryptocurrency prices can interact is vital for devising of effective investment strategies. Results of GARCH-DCC model permitted to suggest Bitcoin and Ether for pairs trading; Tether can be used as a base currency in both pairs. A dynamic portfolio of three cryptocurrencies was built based on the results of the Johansen Cointegration test.

In the second chapter cryptocurrency prices were forecasted by comparing performance of the Long short-term memory (LSTM), Support Vector Regression (SVR) and Random Forest regression models, that are established machine learning tools to grasp non-linear complex interactions. Price forecasts were contrasted against the predictions of the traditional linear Autoregressive integrated moving average (ARIMA) model.

In order to combine the classic and machine learning approaches on financial time series forecasting, Hybrid ARIMA-Random Forest, Hybrid Arima-SVR and Hybrid Arima-LSTM models were developed. The hybrid combination is expected to be successful, since it permits decomposition of time series in linear and nonlinear trends that are forecasted separately. According to this approach, linear part is predicted with ARIMA, and non-linear part, i.e. residuals of ARIMA model, are forecasted with machine learning models that are known to outperform the traditional models in capturing nonlinear relationships. Consecutively, the combination of the two types of models should encompass with high accuracy both linear and nonlinear tendencies of the time series in the hybrid framework.

The study allows us to make several noteworthy contributions to the existing empirical literature:

First, it provides a thorough investigation of interdependencies and interactions among three cryptocurrencies with the largest market capitalization. The interdependencies in cryptocurrency market unavoidably warrant a causal analysis, that was not done in existing literature in such depth as in this study.

Although majority of studies focus on Bitcoin, the present study considers less popular for prediction Ether and Tether and investigates their long-term and short term interactions among themselves and with Bitcoin.

High correlation and mutual information between BTC and ETH can be used in devising an investment strategy can consist in spread trading by simultaneously taking opposite positions on BTC and ETH in order to take advantage of the small differences that arise between these underlyings over time. In contrast, negative correlations between USDT and BTC or ETH permit to diversify or mitigate the risk associated with a portfolio.

Johansen cointegration analysis permitted to construct the stationary portfolio of Bitcoin, Ether and Tether. Investors can profit from purchasing/selling this portfolio when price is low/high and get a profit when its price returns to the mean or crosses above/below a certain level. Cointegrated assets are also used for hedging and pairs trading, limiting the potential risks and loses.

Second, as an remarkable application in the cryptocurrency market, this study identifies the most accurate method of price forecasting for Bitcoin, Ether and Tether, contrasting the traditional linear ARIMA model, three machine learning models and three hybrid techniques.

The results of computer simulations revealed that the least predictable cryptocurrency is USDT whose price trajectory that is characterized by a high frequency of oscillations of different amplitude around the mean.

The most predictable cryptocurrency is Ether. ARIMA, Random Forest, Hybrid Arima-Random Forest, Hybrid Arima-SVR, LSTM and hybrid ARIMA-LSTM models ensured almost 100% accuracy of predictions on training and test data.

As for Bitcoin, ARIMA, Random Forest, Hybrid Arima-Random Forest, Hybrid Arima-SVR, LSTM and Hybrid Arima-LSTM had accuracy of predictions exceeding 97%.

A comparison among the models permits to conclude that the LSTM model had the highest predictive capacity for all three cryptocurrencies, followed by the Random Forest and Hybrid Arima-Random Forest models.

No previous study to the best of author's knowledge and through search in peer-reviewed databases has empirically explored the interdependencies and interactions among Bitcoin, Ether and Tether and combined the ARIMA with machine learning regression techniques for their price prediction.

The results of this study is especially relevant for finance practitioners and cryptocurrency traders.

The full model code that is written in R is provided on GitHub Repository with the link https://github.com/ODknv/ML4cryptos.

Background and Literature Review

This thesis aims to add to the scientific discussions on cryptocurrency forecasting.

A cryptocurrency is volatile digital security designed to operate as a tool of exchange that uses secure cryptography (Gandal and Halaburda 2016). Cryptocurrencies seek to benefit from security, anonymity, lack of central control and slowly evolving governmental regulation. Although cryptocurrencies have been increasingly used by investors for portfolios diversification and hedging, thanks to their high rates of return and low correlation with the other types of assets, they are viewed as a highly speculative asset (Corbet et al. (2018), Chuen et al. (2017), Liu and Tsyvinski (2021). Because of high volatility cryptocurrencies so far did not became a standard globally accepted medium of exchange. The prices of cryptocurrencies are characterized by nonlinear trends and high volatility that increases investor risk.

Apparently, there is no consensus on the real value of cryptocurrencies among the scientific community. The prevailing conclusion is that the exchange rate of the majority of cryptocurrencies is determined mainly by the ratio of demand and supply, which in its turn is influenced by the public regulation of crypto markets, inflation, interest rate, technological developments, such as improvements in blockchain technology, security concerns, institutional interest, high profile support, competition from other cryptocurrencies, and market sentiment, CNBC (2019), Selmi et al (2018), Cheah and Fry (2015), Blau (2018), Liu and Tsyvinski (2021).

The total capitalization of the global cryptocurrency market peaked at over $2.9 trillion in November 2021. However, because of the high volatility inherent to the crypto markets, at the end of 2022, crypto market cap fell to $798 billion (Forbes, 2023). Bitcoin (BTC), Ether (ETH) and Tether (USDT) had the largest market capitalization in 2022[1].

Bitcoin is by far the largest in terms of volumes traded and market capitalization crypto coin (CoinMarketCap, 2023, https://coinmarketcap.com) out of nearly 10,000 cryptocurrencies that exist today (Best, 2022). Bitcoin was created by an unknown entity under the name of Satoshi Nakamoto and was released in January 2009. In proof-of-work based systems such as Bitcoin, participants contribute to broadcasting and verifying transactions. “Miners” do the computational work required to assemble new, valid blocks and commit them to the shared ledger. In proof-of-work systems mining does not serve the purpose of verifying transactions (as this activity is fairly

1 StealthEX ranking (December 28, 2022) based on market capitalization: Top 10 Cryptocurrencies of 2022 by Market Cap**,** https://stealthex.io/blog/top-10-cryptocurrencies-of-2022-by-market-cap/

light computationally), but of building a credible commitment against an attack. Since blocks are chained together, the audit trail formed overtime becomes more difficult to tamper with as more blocks are added, and computing power has been sunk to support it. As a result, in proof-of-work systems, a blockchain is only as secure as the amount of computing power dedicated to mining it. This generates economies of scale and a positive feedback loop between network effects and security: as more participants use a crypto token, its value increases, which in turn attracts more miners (due to higher rewards), ultimately increasing the security of the shared ledger.

Ether is the coin to invests in the Ethereum cryptocurrency. Ethereum is a decentralized computing network built on blockchain technology also makes it possible to create and run applications, smart contracts and other transactions on the network. Bitcoin doesn't offer these functions. Ethereum uses the proof of stake instead of the proof-of-work consensus mechanism, which permits it to process transactions much faster, however, it is less secure against attacks.

Tether token is a stablecoin backed by fiat currencies like U.S. dollars and Euro. Tether hypothetically keeps a value equal to one of those denominations. In theory, this means Tether's value is supposed to be more stable than other cryptocurrencies, and this feature is favored by investors who are wary of the extreme volatility of crypto coins and tokens. Tether tokens are supported by multiple blockchains (e.g., Bitcoin, Ethereum, TRON, EOS, Algorand, Solana, and Bitcoin Cash). So, the token designers wanted to facilitate Tether integration into global cryptocurrency market and adoption by investors.

Almost all cryptocurrencies, including the abovementioned, are secured by blockchain networks. A blockchain is a decentralized ledger of all transactions across a peer-to-peer network. Blockchain works by having transactions recorded on "blocks" which are linked together with a cryptographic hash function so that the transaction history cannot be changed without ruining the entire blockchain. Cryptographic proof is provided by a peer-to-peer network consisting of nodes (servers that store the entire transaction history/blockchain) and miners that generate new blocks within the network. Together, they form 'the Blockchain', a validated public ledger consisting of all previous events. It can be used to register any asset by anyone and anywhere (Swan, 2015).

Under the constantly changing market conditions, the boom-bust nature of cryptocurrency markets, rapid emergence of new crypto coins and tokens, a search of an adequate method for cryptocurrency price forecasting is highly relevant as it will permit to reduce investor risk. Investors regularly monitor the markets in a pursuit of financial assets that will out- or underperform in the market in order to create investment strategies. Even small improvements in predictive power can generate large profits. Therefore, correct forecasting of future values in a time series keeps on being among the most researched topics in financial investment.

Bitcoin, Ether and Tether are supposed to compete with each other and share the same price formation determinants. Considering that Bitcoin continues to be a dominant cryptocurrency and is often used as a medium of exchange to buy other cryptocurrencies, price developments of Bitcoin are likely to impact prices of other cryptocurrencies. Therefore, it is important to capture to the fullest information, patterns, dependencies, desirably from the different perspectives, about the stocks before predicting their price movements. Unlike traditional equity market, many cryptocurrencies generally are correlated among themselves and have low correlation with the traditional assets (Lahajnar S.& Rozanec A., 2020). Nonetheless, as proven by Hassler and Hosseinkouchack (2022) and Yule (1926), correlation may be misleading in nonlinear systems such as stock markets, which gives rationale to gauge cryptocurrency interactions with Granger causality, cointegration and transfer entropy methods. For example, Obeng, C., & Attor, C. (2022) conducted the cointegration test and impulse response analysis with VECM model for the top 9 cryptocurrencies' closing prices. They found that any movement in the price level Bitcoin and Ether dominate the market share at 41 % and 17%, respectively, leads to a change in the price or returns of the remaining altcoins. Bação et al (2018) give evidence about the significant lagging feedback from Ethereum, Ripple, Litecoin and Bitcoin Cash to Bitcoin.

Several studies were dedicated to forecasting prices of cryptocurrencies employing both the traditional techniques for time series forecasting and modern machine learning models. Alahmari (2019) defended the supremacy of ARIMA models in predicting prices of Bitcoin, XRP and Ether on a daily time series. Kumar (2019) found out that ARIMA framework outperforms Neural Network approaches in forecasting the returns of Bitcoin and Ethereum on a shorter time-horizon, whereas the opposite holds true in a long run. Rebane et al. (2018) acknowledged that Recurrent Neural Networks generated superior results comparing to ARIMA models for Bitcoin price prediction. Tripathi&Tripathi (2022) compared Random Walk, ARIMA, artificial neural network (ANN) and ensemble model in forecasting prices of Bitcoin and Ripple and demonstrates that the forecasting accuracy of the ensemble model is better than all the component models (Random Walk, ARIMA and ANN). Chen et. al. (2022) acknowledged that LSTM-ARIMA outperforms ARIMA model in forecasting the portfolio prices of bitcoin and gold.

However, few studies until now have forecasted cryptocurrency prices with a hybrid framework. Nevertheless, the hybrid combination is expected to be especially successful in predicting the highly volatile and non-linear cryptocurrency prices, since it permits decomposition of time series in linear and nonlinear trends that can be forecasted separately. According to the hybrid approach proposed by Zhang (2003) linear part is predicted with ARIMA, and non-linear part, i.e. residuals of ARIMA model, are forecasted with machine learning models that are known to capture well the nonlinear interactions. Consecutively, the combination of the two types of models should encompass with high accuracy both linear and nonlinear tendencies of the time series in the hybrid framework.

Chapter 1. Statistical Dependencies and Interactions Among the Selected Cryptocurrencies

This chapter is dedicated to the investigation of the correlation, covariation, cointegration, causality and spillover effects in cryptocurrency prices and returns of the 3 cryptocurrencies with the largest market capitalization in 2022, namely Bitcoin, Ether and Tether.

Cryptocurrency markets are characterized by fluctuating interdependencies that give rise to bubbles or crashes. The interdependencies in cryptocurrency market unavoidably call for a causal analysis and for investigation of short-term correlations and long-term volatility spillovers among the mainstream cryptocurrencies. Knowledge of the types and patterns of cryptocurrency interactions is vital for planning effective investment strategies or crypto trading tactics.

The causality, impulse response and spillover tests are applied to the log returns of crypto coins that are stationary, whereas the cointegration test is applied to the non-stationary Adjusted Closing Prices.

The volatility of low frequency daily historical data of cryptocurrencies can be the best investigated with the GARCH family of heteroscedasticity models whereas the 5-year length of the data justifies the power of investigating the unit root tests, cointegration and causality relationships.

The Figure 1 illustrates that Tether had high frequency of oscillations during 2018-mid-2020, and then it stabilizes. In contrast, BTC and ETH had much higher frequency of oscillations starting from 2021. Overall, prices of BTC and ETH follow quite similar trend of different magnitude. This could imply a potential linear relationship between these two coins.

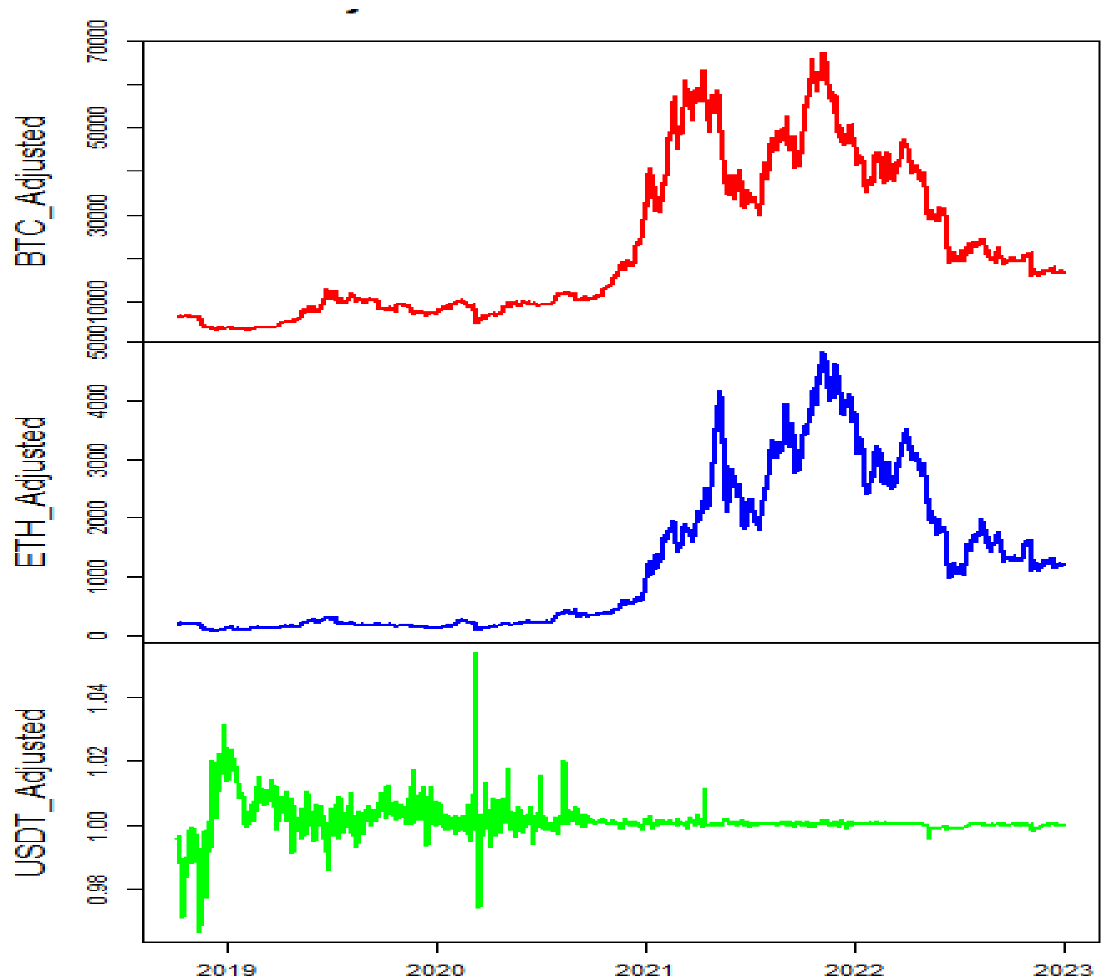


Figure 1 Dynamics of BTC, ETH and USDT Adjusted Close Prices, USD

Source: author's visualization in R using the data of Yahoo finance

1.1. Order of Integration and Structural Breaks

As shown in the Figure 2 below, for all three cryptocurrencies, autocorrelations are significant for the first 30 lags, and the first lag of both the standard and partial correlogram appear closely equal to one, which suggests non-stationarity:

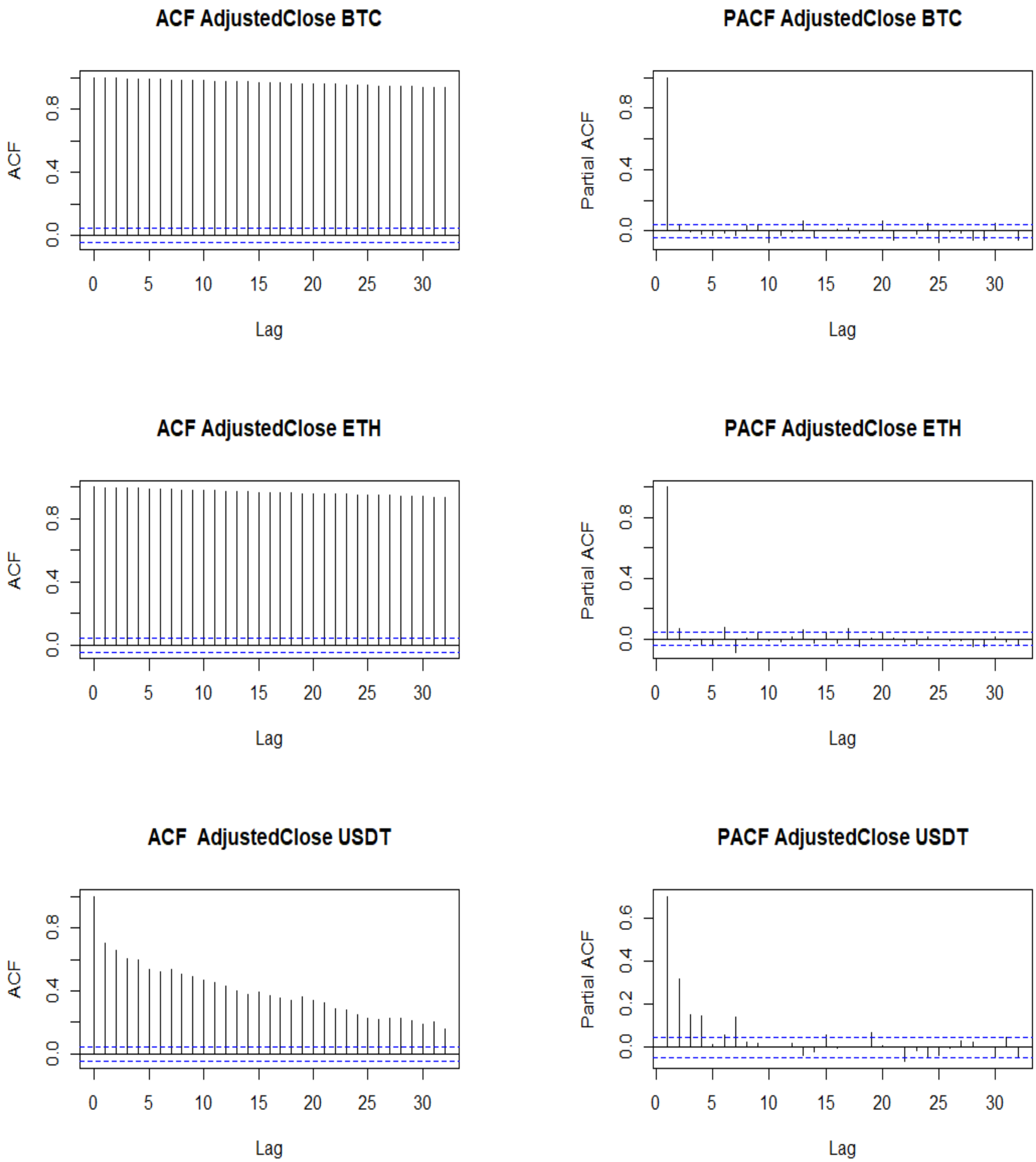


Figure 2 ACF and PACF of Adjusted Close Prices

Source: author's estimates in R using the data of Yahoo finance

Analyzing the ACF and PACF charts on Figure 2 it's difficult to guess the order to integration of the time series, therefore, the stationarity tests were applied upon the differenced series. Apparently, the first differencing of the Adjusted Close Prices was sufficient to make the time series stationary, as illustrated on the Figure 3 and Figure 4 below:

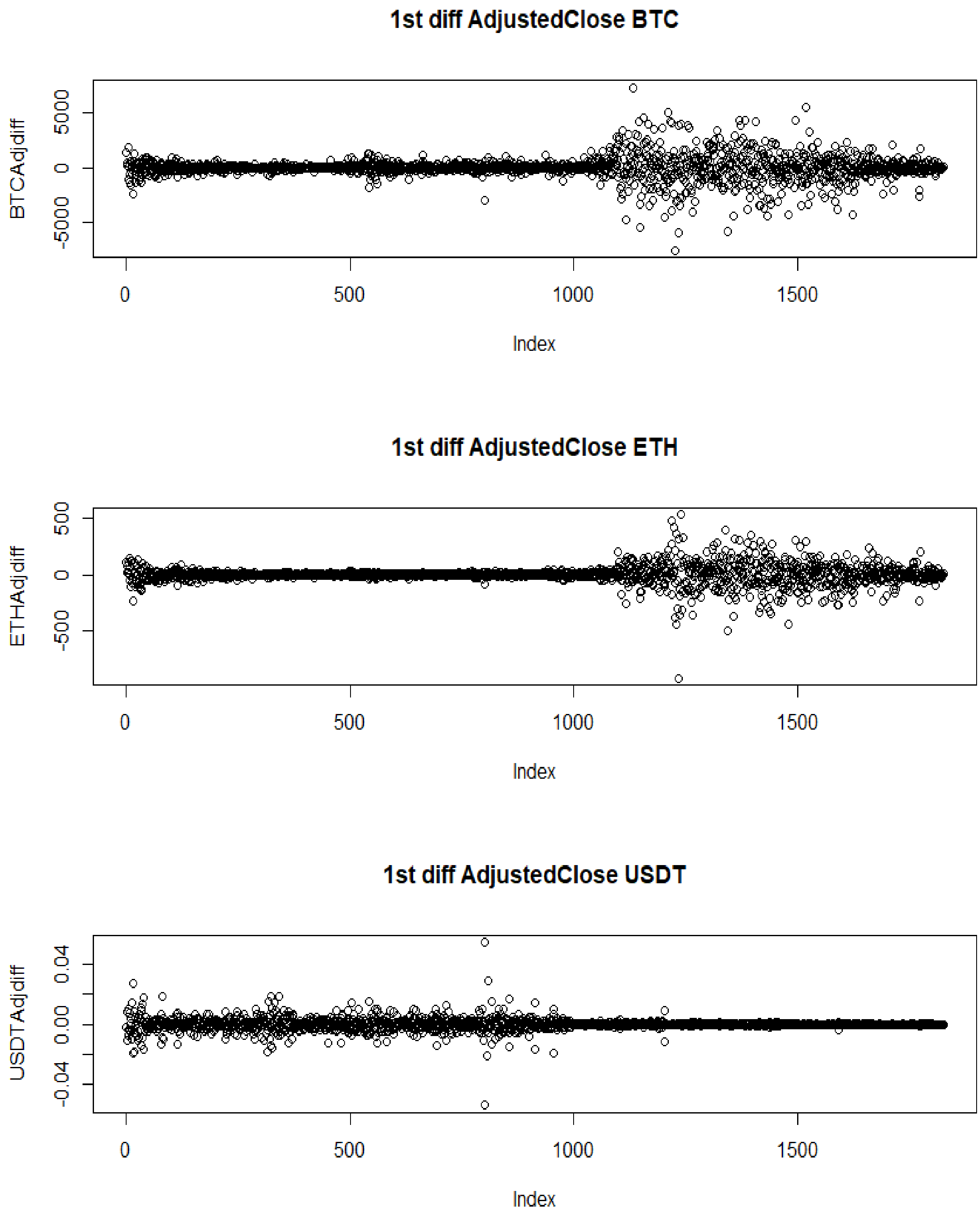


Figure 3 The first differences of Adjusted Close Prices

Source: author’s visualization in R using the data of Yahoo finance

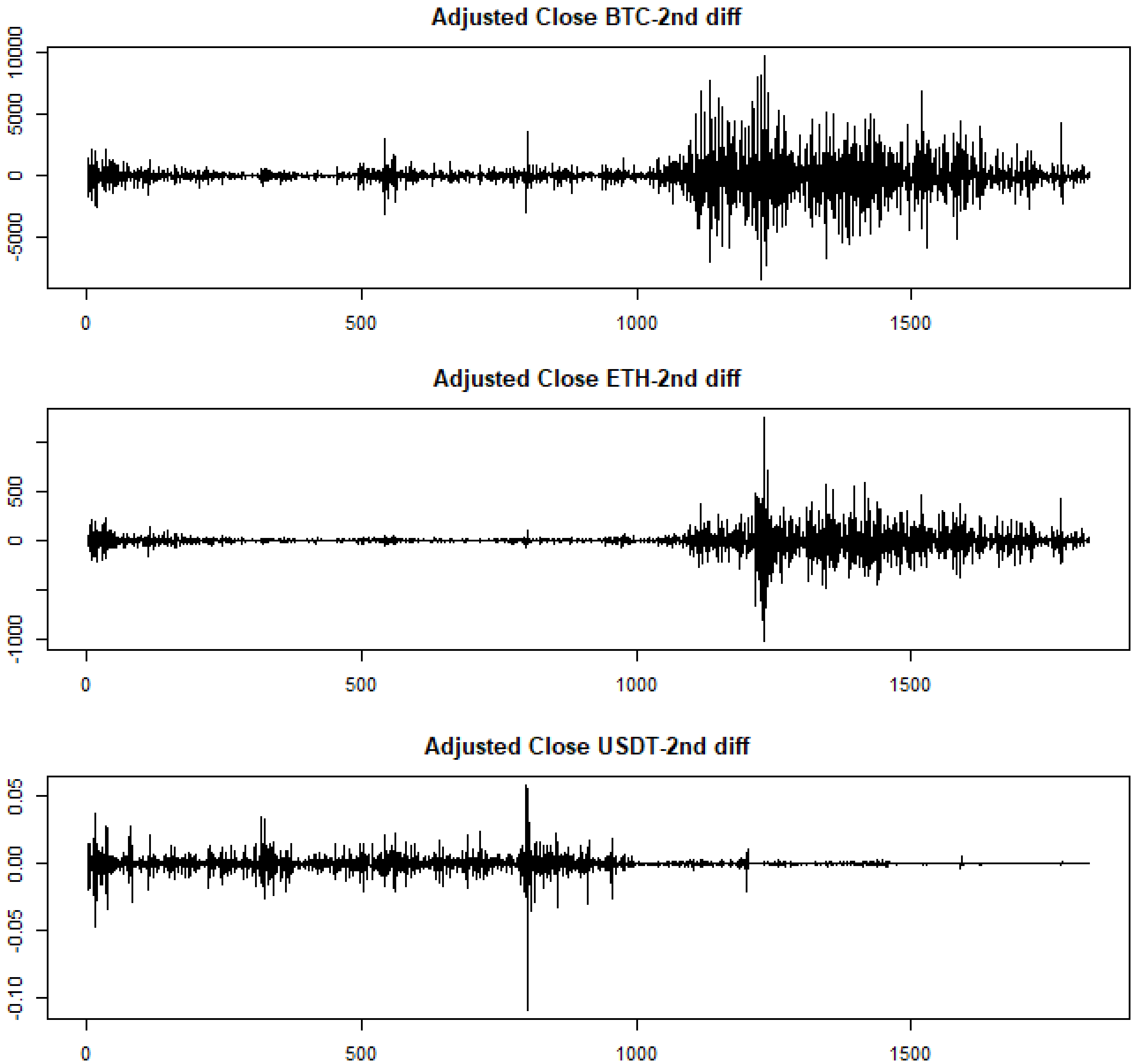


Figure 4 The second differences of Adjusted Close Prices

Source: author's visualization in R using the data of Yahoo finance

Given that the ACF charts suggest long memory in BTC, ETH and USDT (Figure 2), the strength of long memory or persistence was quantified with the Hurst exponent. The Hurst exponent is commonly used in quantitative finance to analyze the behavior of financial markets and make predictions about future trends. The Hurst exponent is calculated using by rescaled range analysis. The basic idea behind this method is to calculate the range of the cumulative sum of a time series and compare it to the standard deviation of the series. Hurst exponent always ranges between 0 and 1. It's values in the range (0.5; 1] suggests that that time series is persistent which roughly translates to trending, i.e. the stock prices are likely to continue in the same direction (either increasing or decreasing) for a longer period of time. Values lower than 0.5 indicate that time-series are anti-persistent which roughly translates to mean reverting. Values around 0.5 imply that the time series is random walk and prediction of future based on past data is not possible.

Values of the Hurst exponent suggest that time series of all three cryptocurrencies display long-term memory, as shown in Table 1:

Table 1 Hurst exponent for the Adjusted Close Prices

Bitcoin Simple R/S Hurst estimation:	0.885
Ether Simple R/S Hurst estimation:	0.882
Tether Simple R/S Hurst estimation:	0.772

Source: author's estimates in R using the data of Yahoo finance

As a matter of a fact, the Hurst exponentiation, the stationarity tests and the ACF diagrams can imply that the series behave in a quasi-unit root way (they have long memory, although will eventually converge to the series average). That means that the AR(1) component of any model may not be statistically exactly equal to 1 but close to it.

Seasonal and Trend Decomposition based on Loess test terminated with the message that time series of cryptocurrencies prices under investigation are "not periodic or have less than two periods". The Classical Seasonal Decomposition by Moving Averages test of R neither detected the seasonality in the data, therefore there is no need to further investigate seasonality effects with the current degree of data disaggregation.

The Stationarity Tests for the Adjusted Close Prices of the selected cryptocurrencies are presented below, Table 2:

Table 2 Stationarity Tests

Crypto Coin	Test results	Stationarity	Order of Integration
BTC	Dickey-Fuller = -1.4168, Lag order = 12, p-value = 0.8252 KPSS Level = 11.31, Truncation lag parameter = 8, p-value = 0.01 KPSS Trend = 1.6892, Truncation lag parameter = 8, p-value = 0.01	non-stationary (ADF and KPSS)	**1**

	Number of differences required for a stationary series: ndiffs 1 nsdiffs 0		
BTC-1st order of differencing	Dickey-Fuller = -11.403, Lag order = 12, p-value = 0.01 alternative hypothesis: stationary KPSS Trend = 0.14279, Truncation lag parameter = 8, p-value = 0.05594	stationary (ADF and KPSS)	**1**
BTC-2nd order of differencing	Dickey-Fuller = -20.004, Lag order = 12, p-value = 0.01 alternative hypothesis: stationary KPSS Trend = 0.0033351, Truncation lag parameter = 8, p-value = 0.1	stationary (ADF and KPSS)	**1**
ETH	Dickey-Fuller = -1.9082, Lag order = 12, p-value = 0.6172 KPSS Level = 11.257, Truncation lag parameter = 8, p-value = 0.01 KPSS Trend = 1.9068, Truncation lag parameter = 8, p-value = 0.01 Number of differences required for a stationary series: ndiffs 1 nsdiffs 0	non-stationary (ADF and KPSS)	**1**
ETH-1st order of differencing	Dickey-Fuller = -12.021, Lag order = 12, p-value = 0.01 alternative hypothesis: stationary KPSS Trend = 0.11426, Truncation lag parameter = 8, p-value = 0.1	stationary (ADF and KPSS)	**1**
ETH-2nd order of differencing	Dickey-Fuller = -19.256, Lag order = 12, p-value = 0.01 alternative hypothesis: stationary KPSS Trend = 0.0031969, Truncation lag parameter = 8, p-value = 0.1	stationary (ADF and KPSS)	**1**
USDT	Dickey-Fuller = -5.8318, Lag order = 12, p-value = 0.01 KPSS Level = 1.0949, Truncation lag parameter = 8, p-value = 0.01 KPSS Trend = 0.28446, Truncation lag parameter = 8, p-value = 0.01 Number of differences required for a stationary series: ndiffs 1 nsdiffs 0	stationary (ADF), non-stationary (KPSS)	**1**
USDT-1st order of differencing	Dickey-Fuller = -14.751, Lag order = 12, p-value = 0.01 alternative hypothesis: stationary KPSS Trend = 0.0054941, Truncation lag parameter = 8, p-value = 0.1	stationary (ADF and KPSS)	**1**

USDT-2nd order of differencing	Dickey-Fuller = -22.857, Lag order = 12, p-value = 0.01 alternative hypothesis: stationary KPSS Trend = 0.002608, Truncation lag parameter = 8, p-value = 0.1	stationary (ADF and KPSS)	**1**

Source: author's estimates in R using the data of Yahoo finance

Since in case of USDT the results of ADF and KPSS stationarity tests seem to contradict each other[2], it requires the more profound testing for the order of integration, unit roots and structural breaks in the data.

The bootUR package of R that performs the Bootstrap Unit Root Tests with False Discovery Rate control for multiple testing by controlling the false discovery rate (Moon and Perron, 2012 and Romano, Shaikh and Wolf, 2008) suggested zero order of integration for USDT and the first order of integration for BTC and ETH Adjusted Close Prices, see Figure 5:

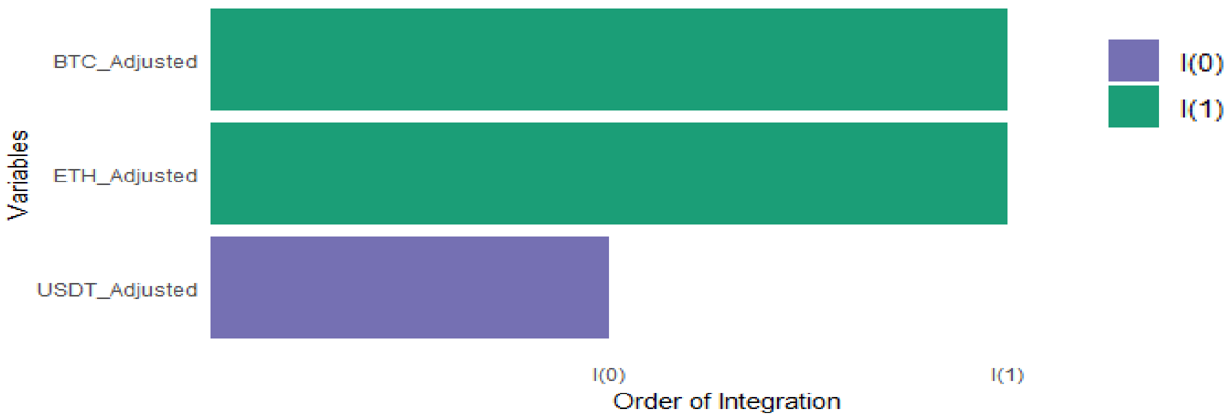


Figure 5 Order of integration of Adjusted Close Prices

Source: author's visualization in R using the data of Yahoo finance

Structural break is the abrupt change in a time series at a point in time and they are a frequent reason of a forecast failure (Hendry 2000; Hendry and Clements 2003). Structural breaks can be caused by economic, technical, or legislative changes or large economic shocks. The longer the time series, the higher the probability that it was affected by the major disruptive events. Detecting the existence and the date of structural breaks is necessary for understanding the drivers of structural changes and their effects on a time series analysis.

The results of the Zivot-Andrews, (Zivot and Andrews, 1992) Unit Root Test for the Adjusted Close Prices of three cryptocurrencies are shown in Table 3 below:

[2] Tether is a "stable coin" whose price is indexed to the US dollar and therefore its normal to expect that it exhibits a different return and volatility characteristics than other cryptocurrencies.

Table 3 Zivot-Andrews Unit Root Test

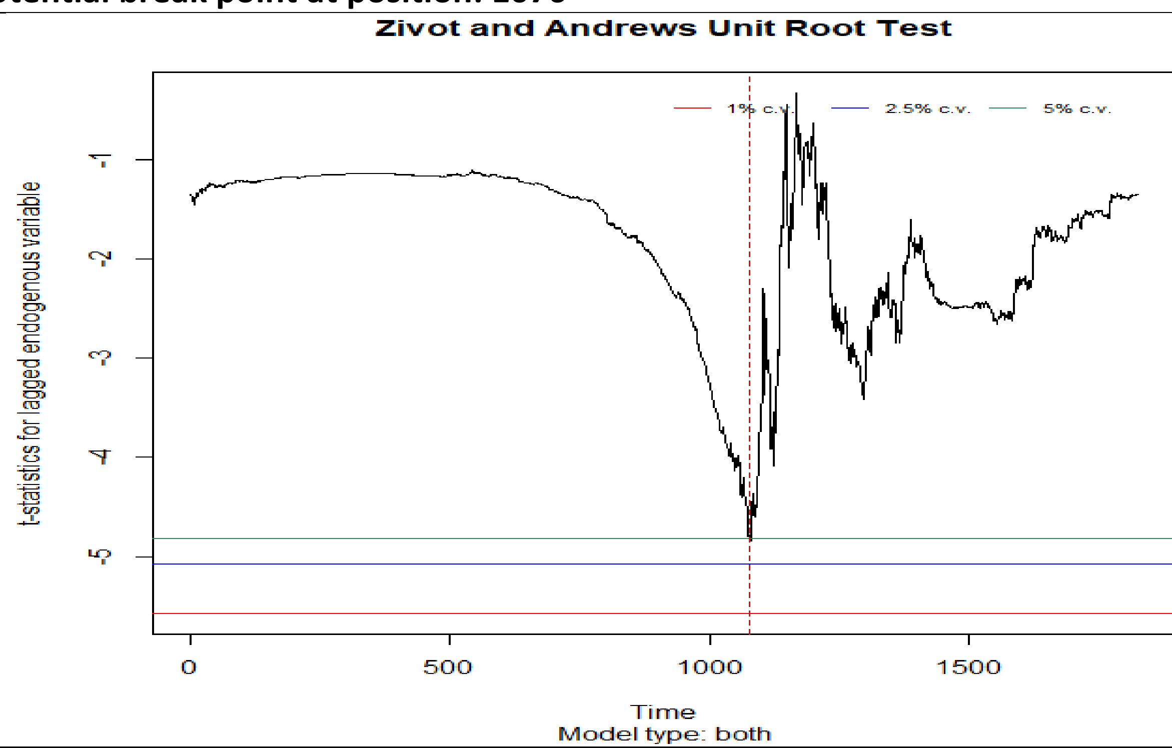
BTC

```
ZAbtc =ur.za(BTCdf$BTC_Adjusted,   model ="both", lag=NULL); summary(ZAbtc)
Call:lm(formula = testmat)
Residuals:
    Min      1Q  Median      3Q     Max
-7486.7  -209.5    -8.9   234.6  6984.3
Coefficients:
              Estimate    Std. Error     t value   Pr(>|t|)
(Intercept)  59.619167    64.372724      0.926     0.354
y.l1          0.984376     0.003227    305.035   < 2e-16 ***
trend         0.140727     0.099206      1.419     0.156
du          767.692766   165.365546      4.642     3.69e-06 ***
dt           -1.194078     0.241505     -4.944     8.34e-07 ***
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1001 on 1821 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9965,      Adjusted R-squared:  0.9965
F-statistic: 1.31e+05 on 4 and 1821 DF,  p-value: < 2.2e-16
```

Teststatistic: -4.8415

Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82

Potential break point at position: 1076



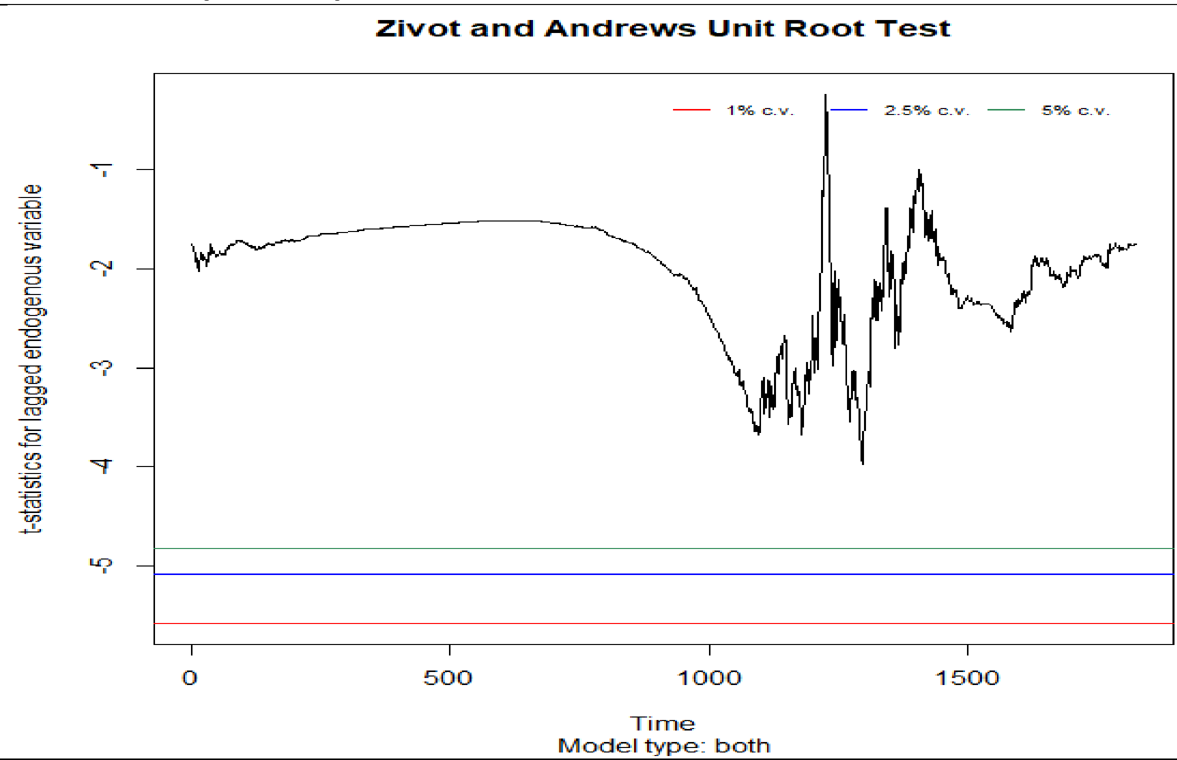
ETH

```
ZAeth =ur.za(ETHdf$ETH_Adjusted,   model ="both", lag=NULL); summary(ZAeth)
Call:lm(formula = testmat)
Residuals:
   Min      1Q  Median      3Q     Max
-896.34  -13.63   -3.22   14.19  541.87
Coefficients:
               Estimate    Std. Error  t value    Pr(>|t|)
(Intercept) -2.473265   4.425430   -0.559     0.57632
y.l1          0.988103   0.002994   330.017   < 2e-16 ***
trend         0.015897   0.006521   2.438     0.01487 *
du           39.884427  11.774046   3.387     0.00072 ***
dt           -0.126616   0.030621  -4.135     3.71e-05 ***
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 79.52 on 1821 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9956,      Adjusted R-squared:  0.9956
F-statistic: 1.035e+05 on 4 and 1821 DF,  p-value: < 2.2e-16
```

Teststatistic: -3.9736

Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82

Potential break point at position: 1297



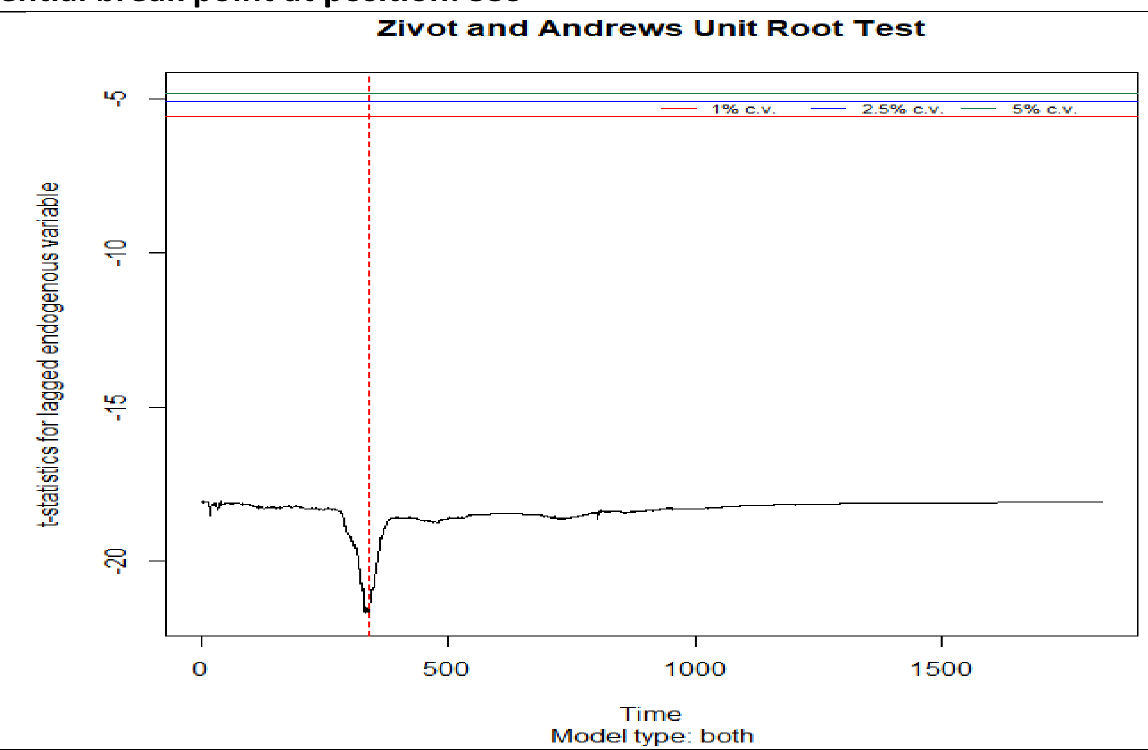
USDT

```
ZAusdt=ur.za(USDTdf$USDT_Adjusted, model ="both", lag=NULL); summary(ZAusdt)
Call: lm(formula = testmat)
Residuals:
     Min        1Q    Median        3Q       Max
-0.033334 -0.000859  0.000057  0.000693  0.052936
Coefficients:
              Estimate    Std. Error  t value  Pr(>|t|)
(Intercept)  4.120e-01   1.895e-02   21.742   < 2e-16 ***
y.l1         5.899e-01   1.885e-02   31.289   < 2e-16 ***
trend       -1.406e-05   2.072e-06   -6.787   1.55e-11 ***
du           5.151e-03   4.849e-04   10.622   < 2e-16 ***
dt           1.199e-05   2.055e-06    5.835   6.36e-09 ***
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.003528 on 1821 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.5274,      Adjusted R-squared:  0.5263
F-statistic:   508 on 4 and 1821 DF,  p-value: < 2.2e-16
```

Teststatistic: -21.7496

Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82

Potential break point at position: 339



Source: author’s estimates in R using the data of Yahoo finance

As follows from Table 3, Zivot-Andrews (Z-A) test for BTC suggests structural break in data on 11/12/2020 (potential break point at position: 1076). In absolute values, t-statistics of Z-A test is lower than the critical values of (1%, 5% and 10%) levels, that permits to accept the Null hypothesis of unit root. The Zivot-Andrews test for ETH suggests that there is break in data on 20/07/2021 (potential break point at position: 1297). The T-statistics of Z-A test is lower than critical values (in absolute value), accepting the Null hypothesis of unit root. The Zivot-Andrews test for USDT suggests that there is break in data on 05/12/2018 (potential break point at position: 339). In absolute numbers, the T-statistics of Z-A test is higher than critical values, so it rejects the Null hypothesis of unit root.

1.2. Correlation Analysis

Correlation analysis has detected significant positive correlation (0.832) between the log returns of BTC and ETH, and also high positive correlation between the Adjusted Close Prices of BTC and ETH (0.923), which means that BTC and ETH tend to move together, whereas USDT has a low negative correlation with BTC and ETH, see Table 4:

Table 4 Correlations among cryptocurrencies

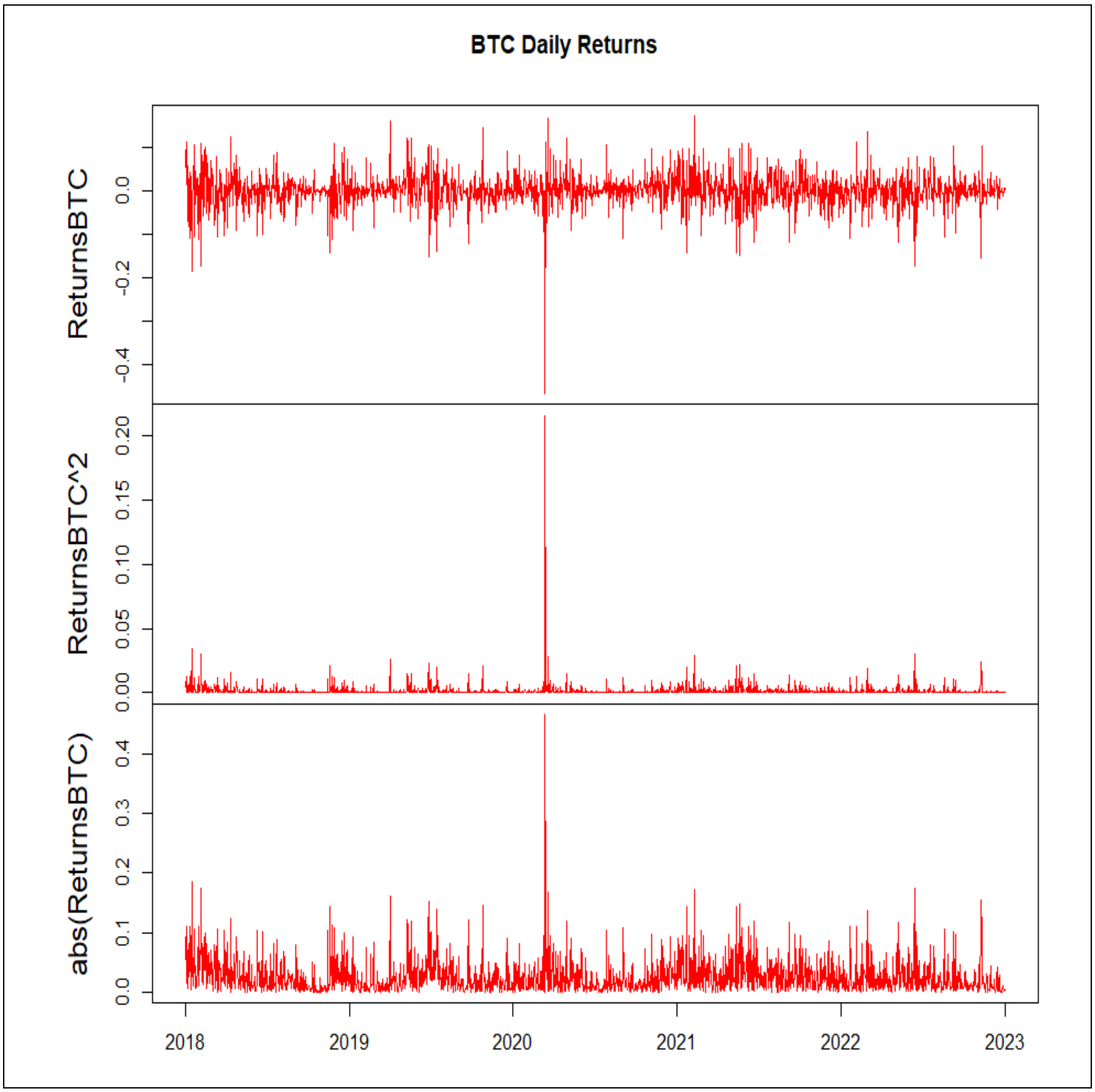
Log Returns			
	BTC	**ETH**	**USDT**
BTC	1.000	0.832	-0.026
ETH	0.832	1.000	-0.060
USDT	-0.026	-0.060	1.000
Adjusted Close Prices			
BTC	1.000	0.923	-0.155
ETH	0.923	1.000	-0.153
USDT	-0.155	-0.153	1.000

Source: author's estimates in R using the data of Yahoo finance

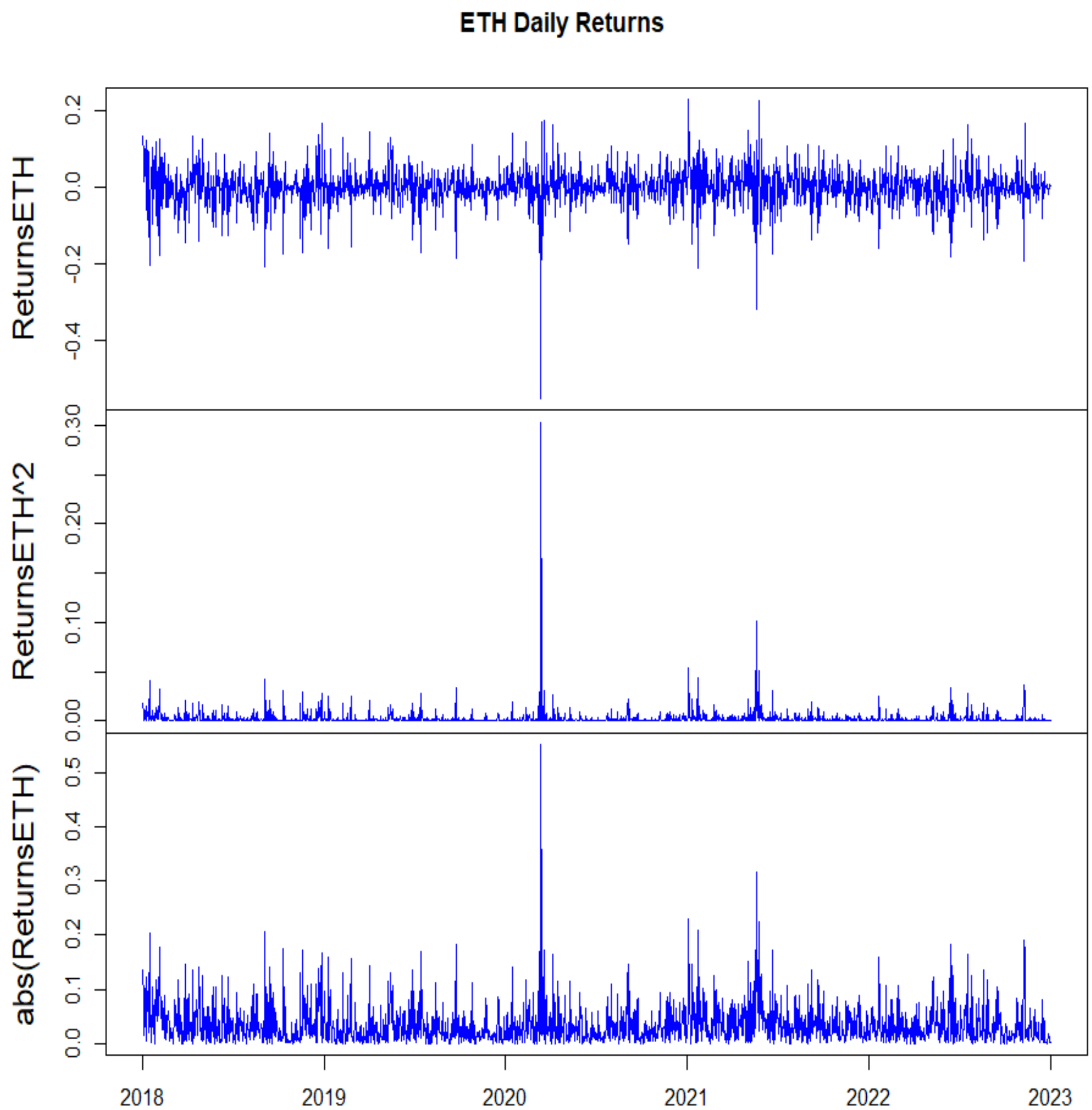
Considering high correlation between BTC and ETH, a strategy can consist in spread trading by taking simultaneously the opposite positions on BTC and ETH in order to take advantage of small differences that arise between these underlyings over time. Negative correlations between USDT and BTC or ETH can be used to diversify, or mitigate, the risk associated with a portfolio, so that a fall in price of one cryptocurrency will at least partially offset with a rise in price of another cryptocurrency.

1.3. Autocorrelation in Log Returns and the Autoregressive Conditional Heteroscedasticity

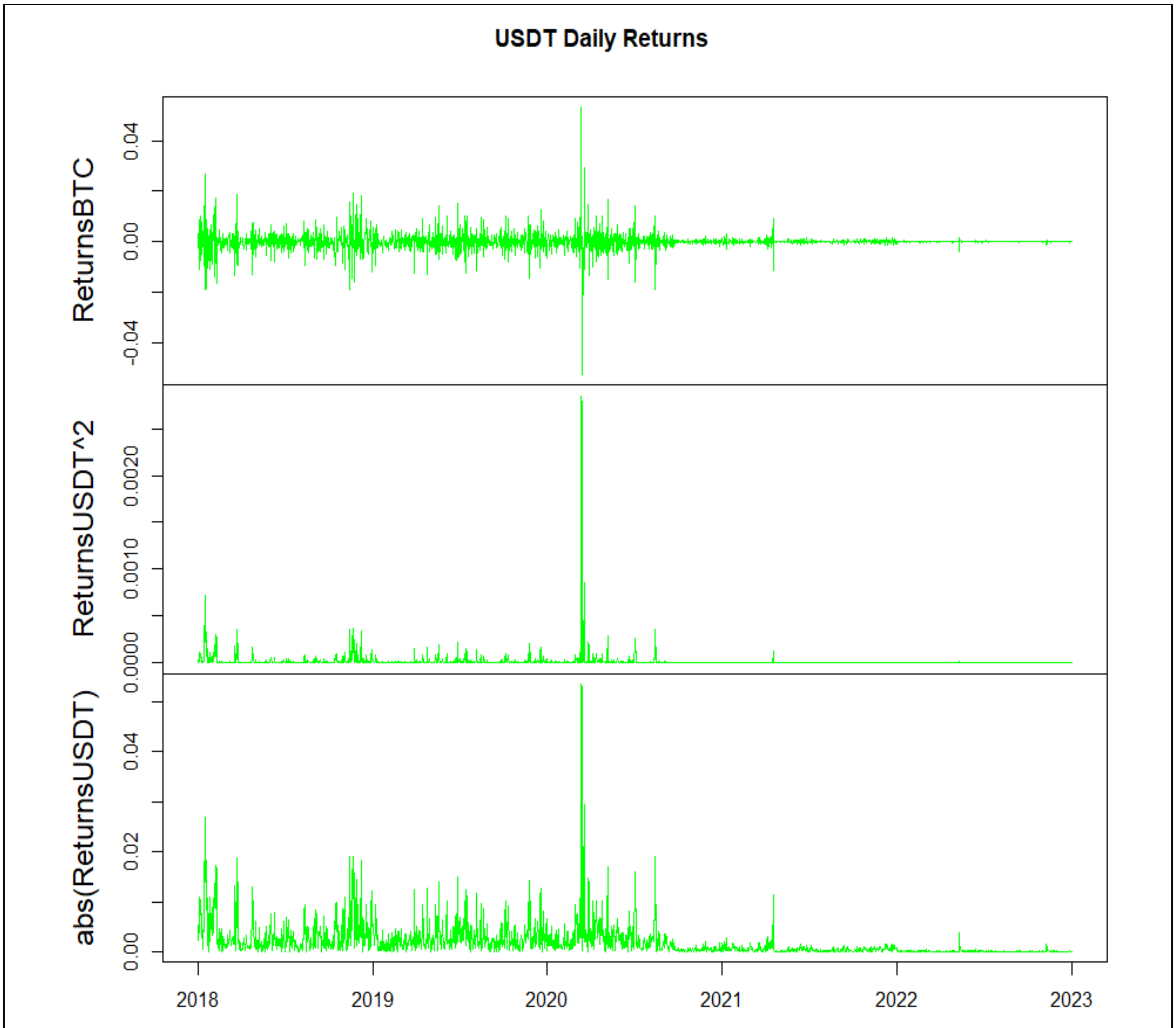
Figure 6 illustrates the dynamics of log returns, squared log returns and absolute values of log returns of the three cryptocurrencies. The Ljung-Box test confirmed the absence of autocorrelation in the square log returns of all cryptocurrencies. ARCH-LM Test applied to the log returns manifested the presence of correlation between the volatility of time series, the so-called ARCH effects, Figure 6. Existence of ARCH effects is important for the GARCH modelling that describes a changing pattern in variance, and was carried further in this study.



Auto correlation in square log retruns: **Box-Ljung test with 12 lags**	**Autoregressive conditional heteroscedasticity ARCH LM-test with 12 lags**
X-squared = 44.898, df = 12, p-value = 1.073e-05	Chi-squared = 38.518, df = 12, p-value = 0.0001263



Auto correlation in square log retruns: **Box-Ljung test with 12 lags**	**Autoregressive conditional heteroscedasticity ARCH LM-test with 12 lags**
X-squared = 59.601, df = 12, p-value = 2.668e-08	Chi-squared = 49.238, df = 12, p-value = 1.9e-06



Auto correlation in square log retruns: **Box-Ljung test with 12 lags**	**Autoregressive conditional heteroscedasticity** **ARCH LM-test with 12 lags**
X-squared = 518.57, df = 12, p-value < 2.2e-16	Chi-squared = 524.14, df = 12, p-value < 2.2e-16

Figure 6 Log Returns of selected cryptocurrencies, USD

Source: author's visualization in R using the data of Yahoo finance

The above charts show that volatility of all three cryptocurrencies exhibits clustering and has varied considerably over time. Although the magnitude of volatility is significantly lower for the USDT log returns than for the BTC and ETH log returns, all three cryptocurrencies display quite similar patterns with a large spike in volatility around March 2020 during the market turmoil.

The descriptive statistics for the log returns is presented in Table 5:

Table 5 Descriptive Statistics for the Log Returns

	BTC	ETH	USDT
Mean	0.001	0.001	0
Variance	0.001	0.002	0
Standard Deviation	0.038	0.05	0.004
Skewness	-1.211	-1.151	0.328
Excess Kurtosis	16.462	12.406	59.775

Source: author's estimates in R using the data of Yahoo finance

As shown in Table 5, Ether has the most volatile returns. In contrast, Tether has very stable returns because it's a stablecoin. All cryptocurrencies exhibit the high degree of excess kurtosis. That is common among the cryptocurrencies and is an indicator for heavy tails, that is a requirement for GARCH models.

1.4. Mutual Information Between the Cryptocurrency Pairs

The value of mutual information between the cryptocurrency pairs was calculated for further analysis. This value is a measure of mutual dependence between the two variables. Mutual information quantifies the amount of information that one variable provides about the other variable, and it does so without relying on any assumptions e.g., linearity. The Fastmit package of R was used to quantify the mutual information among the adjusted close prices of cryptocurrencies. This estimator is based on k-nearest neighbor method proposed by A. Kraskov et al. (2004). Mutual information, MI(X; Y), is the expected value of the logarithmic likelihood ratio between the joint probability distribution of random variables X and Y, (p(X, Y)) and the product of their marginal distributions (p(X) and p(Y)):

$$\mathrm{MI}(X;\ Y) = \mathrm{E}[\log(\mathrm{p}(X, Y) / (\mathrm{p}(X) * \mathrm{p}(Y)))] \quad (1)$$

In accordance with the results shown in Table 6, mutual information is especially strong between the BTC and ETH, and weak between the ETH and USDT, Table 6:

Table 6 Mutual information between cryptocurrencies pairs

	ETH	USDT
BTC	1.781611	0.3483559
ETH		0.2946063

Source: author's estimates in R using the data of Yahoo finance

1.5. Bilateral Granger Causality and Shannon Transfer Entropy Analysis

Positive causality and positive correlation is expected for the assets that are suitable for the pairs trading. Conversely, negative causality is anticipated in cases of competing financial entities such as equities and bonds, that can be used for portfolio diversification.

The causality between cryptocurrencies in a short run was tested with the Granger causality test. This test makes use of Student's t-statistic and F-statistic tests and shows to which extent the value movement of one variable in the past can explain the value movement of another variable. The mathematical formulation of multivariate Granger causality is defined as follows (Granger, 1969):

$$X_t = \sum_{\tau=1}^{p} A_\tau Y_{t-\tau} + \varepsilon_t, \tag{2}$$

where X_t is a d-dimensional multivariate time series with $t = 1,2, \dots, T$, p is the maximum number of lagged observations; A is the coefficient matrix and ε_t is the error term. The Granger causality is tested with the assumption of linear regression since it is not sensitive to nonlinear causal relationships.

The bilateral Granger Causality tests (Granger, 1969) that were ran on the log returns of different pairs of cryptocurrencies show that among that among the tested pairs only ETH can predict USDT with the lag 2, Table 7:

Table 7 Bilateral Granger Causality between the Log Returns

Causality between BTC and ETH, lag=3	
Dependent vairable BTC F 1.459 Pr(>F) 0.2239	Dependent vairable ETH F 1.3779 Pr(>F) 0.2478
Causality between BTC and USDT, lag=2	
Dependent vairable USDT F 2.4491 Pr(>F) 0.08666	Dependent vairable BTC F 0.6071 Pr(>F) 0.5451
Causality between USDT and ETH, lag=2	
Dependent vairable USDT F 3.7716 Pr(>F) 0.0232 *	Dependent vairable ETH F 0.7576 Pr(>F) 0.469

Source: author’s estimates in R using the data of Yahoo finance

This causality can be explained by the fact that USDT more often uses Ethereum than Bitcoin blockchains[3]. In fact, the causality between Bitcoin and USDT can be considered as “marginally significant”.

For this test, lags were selected based on the Akaike (AIC), Schwarz (SC or BIC), Hannan-Quinn (HQ) and Final the Prediction Error (FPE) information criteria.

Furthermore, the causality between the log returns for the pairs of selected cryptocurrencies was investigated with Shannon Transfer entropy method (Shannon, 1948). It is a nonlinear generalization of the Granger causality test that stems from information theory, and therefore is model-free and accounts for both linear and nonlinear causal effects. Shannon transfer entropy is given by the following equation:

$$T_{J\to I}(k,l) = \sum_{i,j} p(i_{t+1}, i_t^{(k)}, j_t^{(l)}) \cdot \log\left(\frac{p\left(i_{t+1}\,\middle|\, i_t^{(k)}, j_t^{(l)}\right)}{p\left(i_{t+1}\,\middle|\, i_t^{(k)}\right)}\right), \quad (3)$$

where $T_{J\to I}$ measures the information flow from process *J* to process *I*. *J* and *I* denote two discrete random variables with marginal probability distributions *p(i)* and *p(j)* and joint probability distribution *p(i,j)* whose dynamical structures correspond to stationary Markov processes of order k. In the bivariate case, information flow from process *J to* process *I* is measured by quantifying the deviation from the generalized Markov property $p\left(i_{t+1}\,\middle|\, i_t^{(k)}\right) = p(i_{t+1}|i_t^{(k)}, j_t^{(l)})$ relying on the Kullback-Leibler distance (Schreiber 2000, Behrendt et al, 2019).

The bilateral tests ran on the log returns of different pairs of cryptocurrencies confirmed that among all tested pairs only ETH log returns can predict USDT log returns with 10% statistical significance, Table 8:

Table 8 Causality with Shannon Transfer entropy between the Log Returns

Direction	TE	Eff. TE	Std.Err.	p-value	sig
ETH–>BTC	0.0028	0.0000	0.0016	0.8767	
	Bootstrapped TE Quantiles (300 replications):				
	0%	**25%**	**50%**	**75%**	**100%**
	0.0008	0.0033	0.0042	0.0056	0.0107
BTC–>ETH	**TE**	**Eff. TE**	**Std.Err.**	**p-value**	**sig**
	0.0012	0.0000	0.0017	1.0000	
	Bootstrapped TE Quantiles (300 replications):				

[3] https://tether.to/en/how-it-works/, https://kriptomat.io/cryptocurrencies/tether/what-is-tether/,https://www.coinbase.com/price/tether

	0% 25% 50% 75% 100%
	0.0014 0.0034 0.0041 0.0053 0.0109
BTC–>USDT	**TE Eff. TE Std.Err. p-value sig**
	0.0074 0.0024 0.0019 0.1200
	Bootstrapped TE Quantiles (300 replications): **0% 25% 50% 75% 100%**
	0.0012 0.0034 0.0048 0.0062 0.0112
USDT–>BTC	**TE Eff. TE Std.Err. p-value sig**
	0.0046 0.0002 0.0017 0.4700
	Bootstrapped TE Quantiles (300 replications): **0% 25% 50% 75% 100%**
	0.0015 0.0035 0.0045 0.0056 0.0100
USDT–>ETH	**TE Eff. TE Std.Err. p-value sig**
	0.0041 0.0000 0.0017 0.5067
	Bootstrapped TE Quantiles (300 replications): **0% 25% 50% 75% 100%**
	0.0015 0.0030 0.0041 0.0053 0.0111
ETH–>USDT	**TE Eff. TE Std.Err. p-value sig**
	0.0079 0.0030 0.0020 0.0967 .
	Bootstrapped TE Quantiles (300 replications): **0% 25% 50% 75% 100%**
	0.0014 0.0035 0.0048 0.0063 0.0117

Source: author's estimates in R using the data of Yahoo finance

1.6. Multivariate Granger Causality Test with a VAR Model

This study continues with testing a multivariate Granger causality with a multivariate Vector Autoregression (VAR) model (Sims, 1980). The model was built for the stationary first differences in Adjusted Close prices of BTC, ETH and USDT. VAR is a multivariate forecasting algorithm that is used when two or more time series influence each other. These models are widely used in financial forecasting since they offer a framework for understanding the intertwined relationships of multivariate time series data in a systematic manner. Several valuable insights can be derived by analyzing the dynamic relationships among the variables in the model.

By the definition, a variable X Granger-causes Y if Y can be better predicted based on the histories of both X and Y than by using the history of Y alone. The VAR models can be written as follows:

$$y_{j,t} = \mu_j + \sum_{j=1}^{m} \sum_{i=1}^{n} \varphi_{ji} \, Y_{t-j} \; + \sum_{j=1}^{m} \varepsilon_{jt} \quad (4)$$

where $y_{j,t}$ is the endogenous variable j at time t, φ_{ji} (L) is the matrix in the backshift operator (L), μ_j is the constant value, and ε_{jt} is the error term. The main results of the multivariate VAR model and its tests statistic are given in :

Table 9:

Table 9 VAR for the Stationarized Adjusted Close Prices

Selection of lag	AIC(n) HQ(n) SC(n) FPE(n) 6 6 3 6
On the whole, the estimation results have rather low Adjusted R-squared coefficients because cryptocurrency data tend to have much noise.	

```
Estimation results for equation BTCAdjdiff:
===========================================
BTCAdjdiff = BTCAdjdiff.l1 + ETHAdjdiff.l1 + USDTAdjdiff.l1 + BTCAdjdiff.l2
+ ETHAdjdiff.l2 + USDTAdjdiff.l2 + BTCAdjdiff.l3 + ETHAdjdiff.l3 + USDTAdjd
iff.l3 + BTCAdjdiff.l4 + ETHAdjdiff.l4 + USDTAdjdiff.l4 + BTCAdjdiff.l5 + E
THAdjdiff.l5 + USDTAdjdiff.l5 + BTCAdjdiff.l6 + ETHAdjdiff.l6 + USDTAdjdiff
.l6 + BTCAdjdiff.l7 + ETHAdjdiff.l7 + USDTAdjdiff.l7 + const

                 Estimate Std. Error t value Pr(>|t|)
BTCAdjdiff.l1   6.603e-02  3.667e-02   1.801  0.07190 .
ETHAdjdiff.l1  -1.339e+00  4.681e-01  -2.861  0.00428 **
USDTAdjdiff.l1 -1.257e+03  6.960e+03  -0.181  0.85665
BTCAdjdiff.l2  -1.900e-02  3.654e-02  -0.520  0.60324
ETHAdjdiff.l2   2.171e-01  4.647e-01   0.467  0.64043
USDTAdjdiff.l2 -2.870e+03  7.940e+03  -0.361  0.71783
BTCAdjdiff.l3   6.399e-02  3.670e-02   1.744  0.08136 .
ETHAdjdiff.l3  -5.866e-01  4.638e-01  -1.265  0.20618
USDTAdjdiff.l3 -3.130e+03  8.220e+03  -0.381  0.70342
BTCAdjdiff.l4  -9.657e-03  3.663e-02  -0.264  0.79206
ETHAdjdiff.l4   6.564e-01  4.633e-01   1.417  0.15670
USDTAdjdiff.l4 -2.165e+03  8.387e+03  -0.258  0.79628
BTCAdjdiff.l5   1.047e-01  3.659e-02   2.863  0.00425 **
ETHAdjdiff.l5  -1.480e+00  4.624e-01  -3.200  0.00140 **
USDTAdjdiff.l5 -1.300e+03  8.203e+03  -0.158  0.87412
BTCAdjdiff.l6  -6.187e-02  3.671e-02  -1.685  0.09209 .
ETHAdjdiff.l6   1.420e+00  4.643e-01   3.058  0.00226 **
USDTAdjdiff.l6 -6.262e+02  7.916e+03  -0.079  0.93696
BTCAdjdiff.l7   2.150e-02  3.678e-02   0.585  0.55895
ETHAdjdiff.l7  -1.249e+00  4.681e-01  -2.668  0.00770 **
USDTAdjdiff.l7  8.520e+02  6.925e+03   0.123  0.90210
const           9.820e-01  2.339e+01   0.042  0.96651
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 997.4 on 1797 degrees of freedom
Multiple R-Squared: 0.03123,   Adjusted R-squared: 0.01991
F-statistic: 2.758 on 21 and 1797 DF,  p-value: 3.23e-05

Estimation results for equation ETHAdjdiff:
===========================================
ETHAdjdiff = BTCAdjdiff.l1 + ETHAdjdiff.l1 + USDTAdjdiff.l1 + BTCAdjdiff.l2
+ ETHAdjdiff.l2 + USDTAdjdiff.l2 + BTCAdjdiff.l3 + ETHAdjdiff.l3 + USDTAdjd
iff.l3 + BTCAdjdiff.l4 + ETHAdjdiff.l4 + USDTAdjdiff.l4 + BTCAdjdiff.l5 + E
THAdjdiff.l5 + USDTAdjdiff.l5 + BTCAdjdiff.l6 + ETHAdjdiff.l6 + USDTAdjdiff
.l6 + BTCAdjdiff.l7 + ETHAdjdiff.l7 + USDTAdjdiff.l7 + const
```

```
                  Estimate Std. Error t value Pr(>|t|)    
BTCAdjdiff.l1   5.967e-03  2.882e-03   2.070 0.038577 *  
ETHAdjdiff.l1  -9.335e-02  3.680e-02  -2.537 0.011279 *  
USDTAdjdiff.l1 -2.228e+02  5.471e+02  -0.407 0.683937    
BTCAdjdiff.l2  -8.656e-03  2.872e-03  -3.013 0.002620 ** 
ETHAdjdiff.l2   8.259e-02  3.653e-02   2.261 0.023880 *  
USDTAdjdiff.l2 -2.318e+02  6.242e+02  -0.371 0.710441    
BTCAdjdiff.l3   6.701e-03  2.885e-03   2.323 0.020297 *  
ETHAdjdiff.l3  -2.496e-02  3.646e-02  -0.685 0.493740    
USDTAdjdiff.l3 -1.888e+02  6.461e+02  -0.292 0.770176    
BTCAdjdiff.l4  -3.345e-03  2.879e-03  -1.162 0.245439    
ETHAdjdiff.l4   5.841e-02  3.642e-02   1.604 0.108919    
USDTAdjdiff.l4 -1.424e+02  6.593e+02  -0.216 0.828998    
BTCAdjdiff.l5   1.172e-02  2.876e-03   4.075 4.79e-05 ***
ETHAdjdiff.l5  -1.836e-01  3.635e-02  -5.050 4.86e-07 ***
USDTAdjdiff.l5 -1.576e+00  6.449e+02  -0.002 0.998050    
BTCAdjdiff.l6  -9.560e-03  2.886e-03  -3.313 0.000942 ***
ETHAdjdiff.l6   1.928e-01  3.650e-02   5.281 1.44e-07 ***
USDTAdjdiff.l6  5.016e+02  6.223e+02   0.806 0.420286    
BTCAdjdiff.l7   1.771e-03  2.891e-03   0.612 0.540312    
ETHAdjdiff.l7  -3.327e-02  3.680e-02  -0.904 0.366090    
USDTAdjdiff.l7 -7.735e+01  5.444e+02  -0.142 0.887027    
const           1.393e-02  1.838e+00   0.008 0.993954    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 78.41 on 1797 degrees of freedom
Multiple R-Squared: 0.04837,   Adjusted R-squared: 0.03725 
F-statistic:  4.35 on 21 and 1797 DF,  p-value: 1.915e-10

Estimation results for equation USDTAdjdiff: 
============================================ 
USDTAdjdiff = BTCAdjdiff.l1 + ETHAdjdiff.l1 + USDTAdjdiff.l1 + BTCAdjdiff.l
2 + ETHAdjdiff.l2 + USDTAdjdiff.l2 + BTCAdjdiff.l3 + ETHAdjdiff.l3 + USDTAd
jdiff.l3 + BTCAdjdiff.l4 + ETHAdjdiff.l4 + USDTAdjdiff.l4 + BTCAdjdiff.l5 + 
ETHAdjdiff.l5 + USDTAdjdiff.l5 + BTCAdjdiff.l6 + ETHAdjdiff.l6 + USDTAdjdif
f.l6 + BTCAdjdiff.l7 + ETHAdjdiff.l7 + USDTAdjdiff.l7 + const 

                  Estimate Std. Error t value Pr(>|t|)    
BTCAdjdiff.l1   2.482e-08  1.240e-07   0.200    0.841    
ETHAdjdiff.l1  -1.097e-06  1.584e-06  -0.693    0.489    
USDTAdjdiff.l1 -5.802e-01  2.355e-02 -24.640  < 2e-16 ***
BTCAdjdiff.l2   1.056e-08  1.236e-07   0.085    0.932    
ETHAdjdiff.l2  -1.365e-07  1.572e-06  -0.087    0.931    
USDTAdjdiff.l2 -3.659e-01  2.686e-02 -13.621  < 2e-16 ***
BTCAdjdiff.l3  -7.896e-08  1.241e-07  -0.636    0.525    
ETHAdjdiff.l3   1.054e-06  1.569e-06   0.671    0.502    
USDTAdjdiff.l3 -2.917e-01  2.781e-02 -10.489  < 2e-16 ***
BTCAdjdiff.l4   3.028e-08  1.239e-07   0.244    0.807    
ETHAdjdiff.l4  -2.270e-07  1.567e-06  -0.145    0.885    
USDTAdjdiff.l4 -1.620e-01  2.837e-02  -5.710 1.32e-08 ***
BTCAdjdiff.l5  -8.532e-08  1.238e-07  -0.689    0.491    
ETHAdjdiff.l5   1.040e-06  1.564e-06   0.665    0.506    
USDTAdjdiff.l5 -2.000e-01  2.775e-02  -7.207 8.41e-13 ***
BTCAdjdiff.l6   1.377e-07  1.242e-07   1.108    0.268    
ETHAdjdiff.l6  -1.090e-06  1.571e-06  -0.694    0.488    
USDTAdjdiff.l6 -1.893e-01  2.678e-02  -7.069 2.23e-12 ***
BTCAdjdiff.l7  -9.978e-08  1.244e-07  -0.802    0.423    
ETHAdjdiff.l7   1.387e-06  1.584e-06   0.876    0.381    
USDTAdjdiff.l7 -4.217e-02  2.343e-02  -1.800    0.072 .  
const          -1.030e-05  7.912e-05  -0.130    0.896    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.003374 on 1797 degrees of freedom
Multiple R-Squared: 0.2722,   Adjusted R-squared: 0.2637 
```

```
F-statistic:    32 on 21 and 1797 DF,  p-value: < 2.2e-16
```

Being tested with 8 lags, the residuals of VAR model are not autocorrelated, however, they do show serial correlation at smaller lags:

```
                    Portmanteau Test (adjusted) with 8 lags
data:  Residuals of VAR object VARmodel
Chi-squared = 15.634, df = 9, p-value = 0.07494
```

Based on the results of the ARCH-LM test we can reject the null hypothesis that the process is homoscedastic:

```
                              ARCH (multivariate)
data:  Residuals of VAR object VARmodel
Chi-squared = 1135.9, df = 180, p-value < 2.2e-16
```

As expected, the residuals of this model are not normally distributed:

```
                             JB-Test (multivariate)
data:  Residuals of VAR object VARmodel
Chi-squared = 150686, df = 6, p-value < 2.2e-16

$Skewness
                          Skewness only (multivariate)

data:  Residuals of VAR object VARmodel
Chi-squared = 1522.1, df = 3, p-value < 2.2e-16

$Kurtosis
                          Kurtosis only (multivariate)

data:  Residuals of VAR object VARmodel
Chi-squared = 149164, df = 3, p-value < 2.2e-16
```

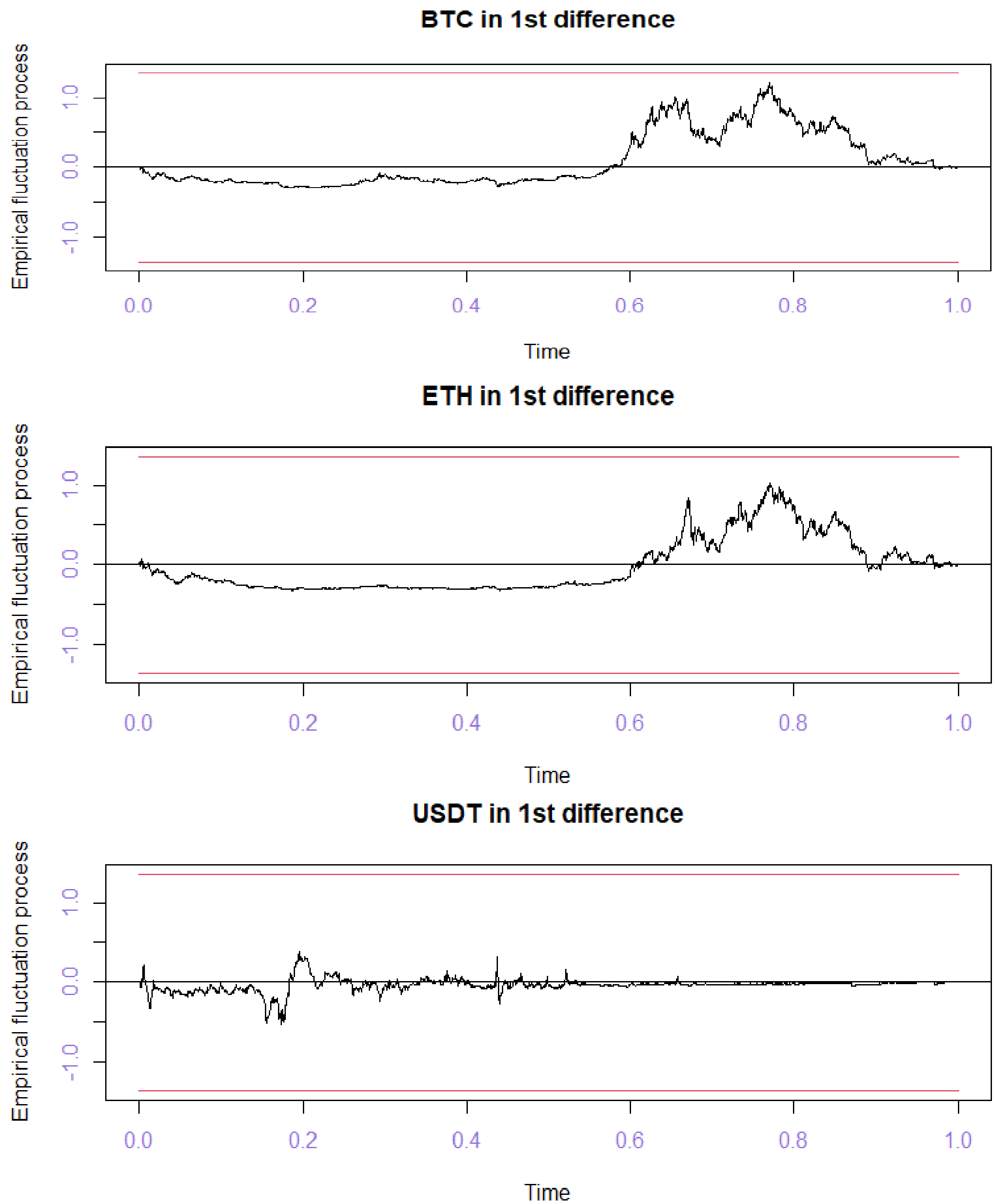
Stability test

The p-value of the M-fluctuation test permits us to reject the null hypothesis is that all model parameters are constant throughout the entire sample period:

```
                              M-fluctuation test

f(efp) = 1.99, p-value = 0.04683
```

The OLS-based CUSUM test indicates that none of the points in the charts out steps the red critical bounds, so there are no structural brakes in the time series of th e stationarized cryptocurrency prices:



Source: author's estimates in R using the data of Yahoo finance

As can be seen from the **Error! Not a valid bookmark self-reference.**, the multivariate Granger causality test that was based on the multivariate VAR model by contrasting each cryptocurrency against the two others, demonstrated that ETH Granger causes BTC and USDT, while USDT Granger causes ETH and BTC (in first differences), Table 10:

Table 10 The multivariate Granger causality test

$Granger

Granger causality H0: BTC do not Granger-cause ETH USDT data: VAR object VARmodel F-Test = 1.5701, df1 = 12, df2 = 5406, p-value = 0.09282 $Instant H0: No instantaneous causality between: BTC and ETH USDT data: VAR object VARmodel Chi-squared = 751.87, df = 2, p-value < 2.2e-16
$Granger **Granger causality H0: ETH do not Granger-cause BTC USDT** data: VAR object VARmodel F-Test = 2.1491, df1 = 12, df2 = 5406, p-value = 0.01164 $Instant H0: No instantaneous causality between: ETH and BTC USDT data: VAR object VARmodel Chi-squared = 752.27, df = 2, p-value < 2.2e-16
$Granger **Granger causality H0: USDT do not Granger-cause BTC ETH** data: VAR object VARmodel F-Test = 2.8128, df1 = 12, df2 = 5406, p-value = 0.0007577 $Instant H0: No instantaneous causality between: USDT and BTC ETH data: VAR object VARmodel Chi-squared = 7.3486, df = 2, p-value = 0.02537

Source: author's estimates in R using the data of Yahoo finance

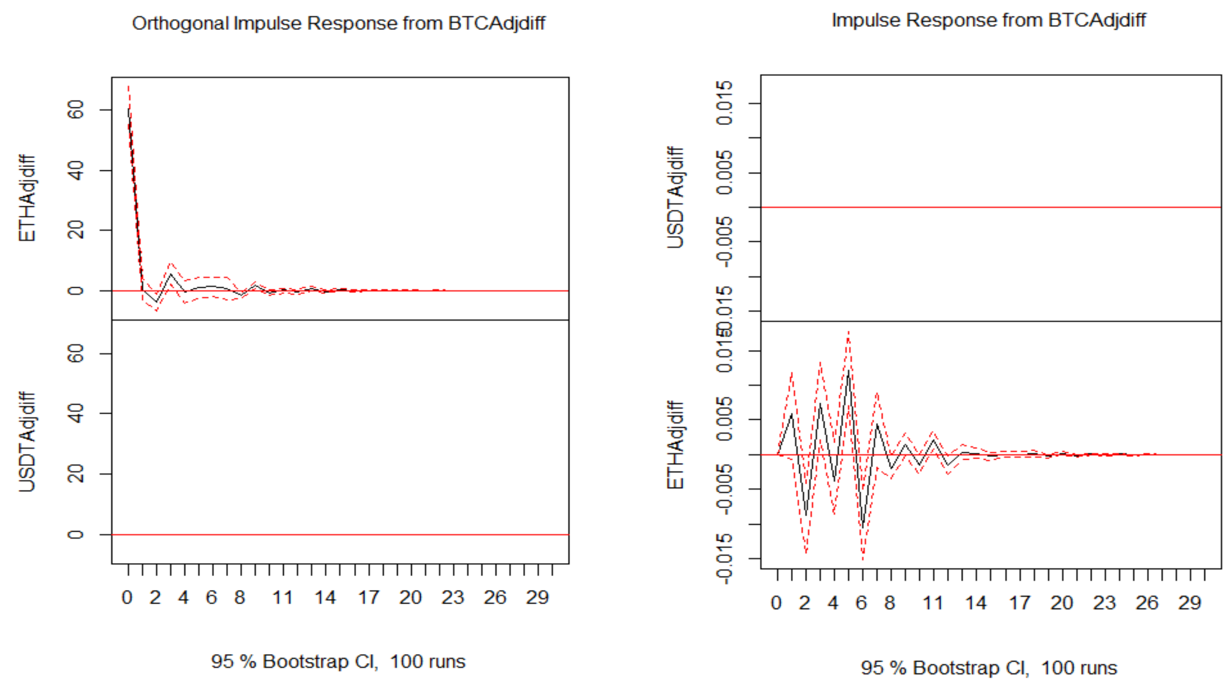
1.7. Analysis of Short-Run Dynamic Interactions with Impulse Response Functions

For the investigation of the short-run dynamic interactions, the impulse response functions (IRF) were applied to the VAR model. An IRF describes the evolution of a VAR model's reaction to a one-standard-deviation shock to each variable (Lütkepohl, 2010). This measure shows how the shock on one variable impacts another variable. In contrast, the forecast error variance decomposition (FEVD) provides information about the relative importance of each shock in affecting all variables in the system.

The impulse response function procedure starts with the unit root process. Since the impulse response function is the coefficient of vector moving average in the VAR model, the VAR model has to be transformed as the vector moving average:

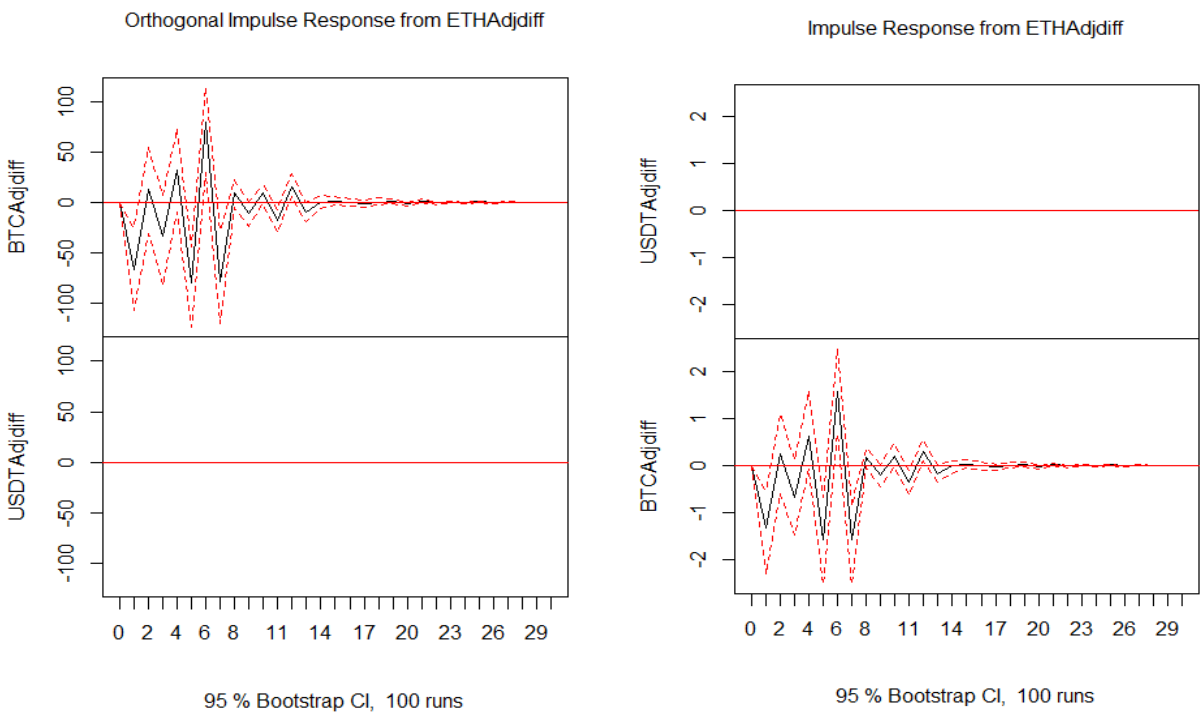
$$y_t = \mu + \sum_{i=0}^{\infty} \varphi_i \, \varepsilon_{t-i} \tag{5}$$

Where y_t is the vector of the time-series variables, μ is the mean of y_t, φ_i is the impulse response function (impact spillover), and ε_t is the vector of the error terms. Given that the impulse responses one draws from the model are conditional on the ordering of the variables[4], different options were tested, as shown in
Figure 7:

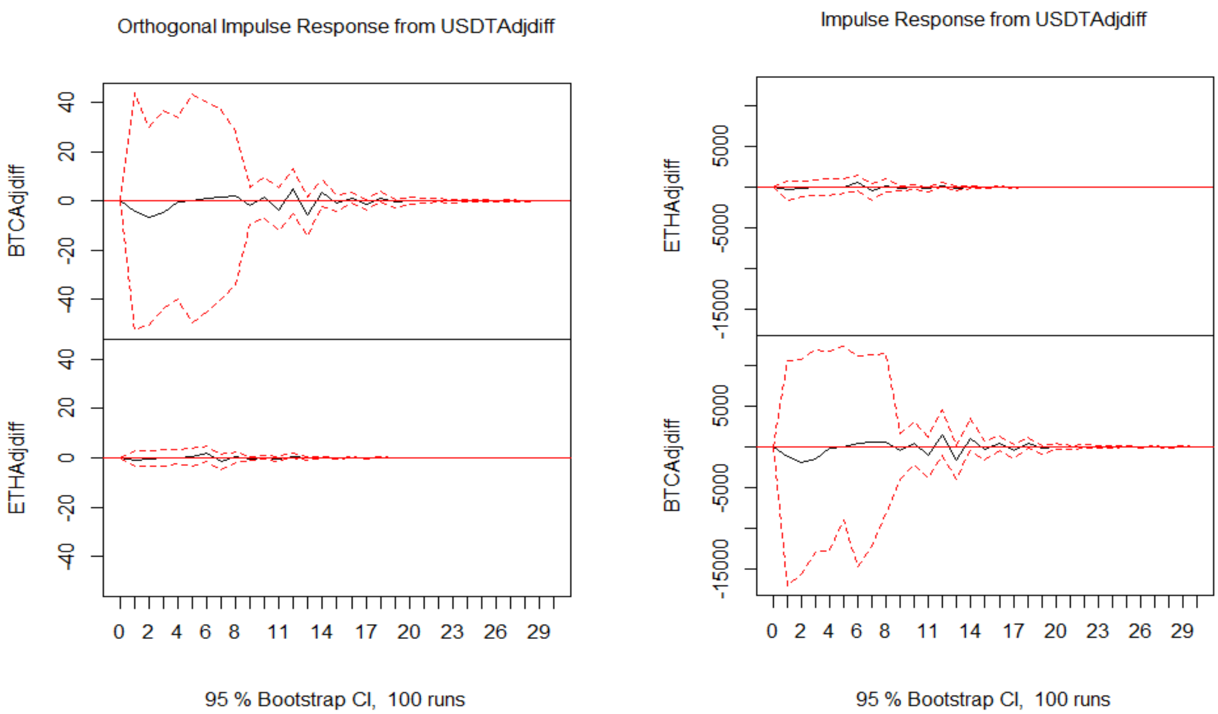


a)

[4] Orthogonalization is done using the Cholesky decomposition of the estimated error covariance matrix and interpretations may change depending on variable ordering.



b)



c)

Figure 7 Orthogonal impulse response shocks

Source: author's estimates in R using the data of Yahoo finance

Modelling results show that the orthogonalized impulse from the stationarized BTC price does not affect USDT but generates a noticeable drop in ETH in the first 2 lags if ETH comes in a first order in the IRF equation, and after the impulse response function converges to zero. However, if Ether appears on the second position in the model, BTC shock causes rather small up and down movements in ETH that fade away after the first 14 lags,

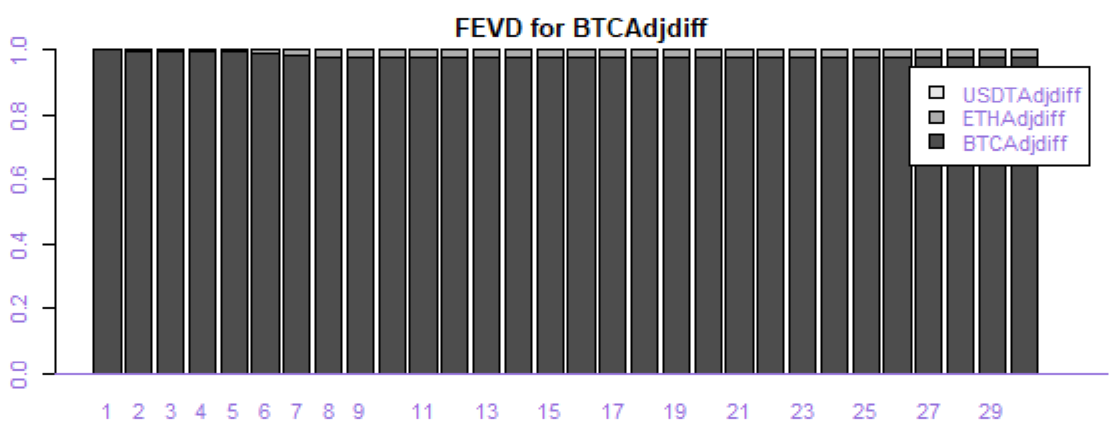
Figure 7 a). Similarly, the impulse from ETH does not affect USDT, however, it generates up and down oscillations in BTC during the first 14 lags. Although the pattern of such movements does not depend on the ordering of BTC, the magnitude is much stronger when BTC is on the first order in the IRF equation,

Figure 7 b). Impulse response from Tether produces similar patterns of impulse responses to ETH and BTC, however Bitcoin response is much stronger if it comes second in the IRF equation. For Ether the opposite holds true,

Figure 7 c).

1.8. Forecast error variance decomposition

The forecast error variance decomposition is based upon the orthogonalized impulse response coefficient matrices and provides information about the relative importance of each innovation in affecting the forecast error variance of all response variables in the system. In other words, FEVD decomposes the variance of the forecast error into the contributions from specific exogenous shocks. By doing so, it demonstrates how important a shock is in explaining the variations of the variables in the model and shows how this importance changes over time. For example, some shocks may not be responsible for variations in the short-run but may cause longer-term fluctuations. FEVD analysis for BTC, ETH and USDT is provided in Figure 8:



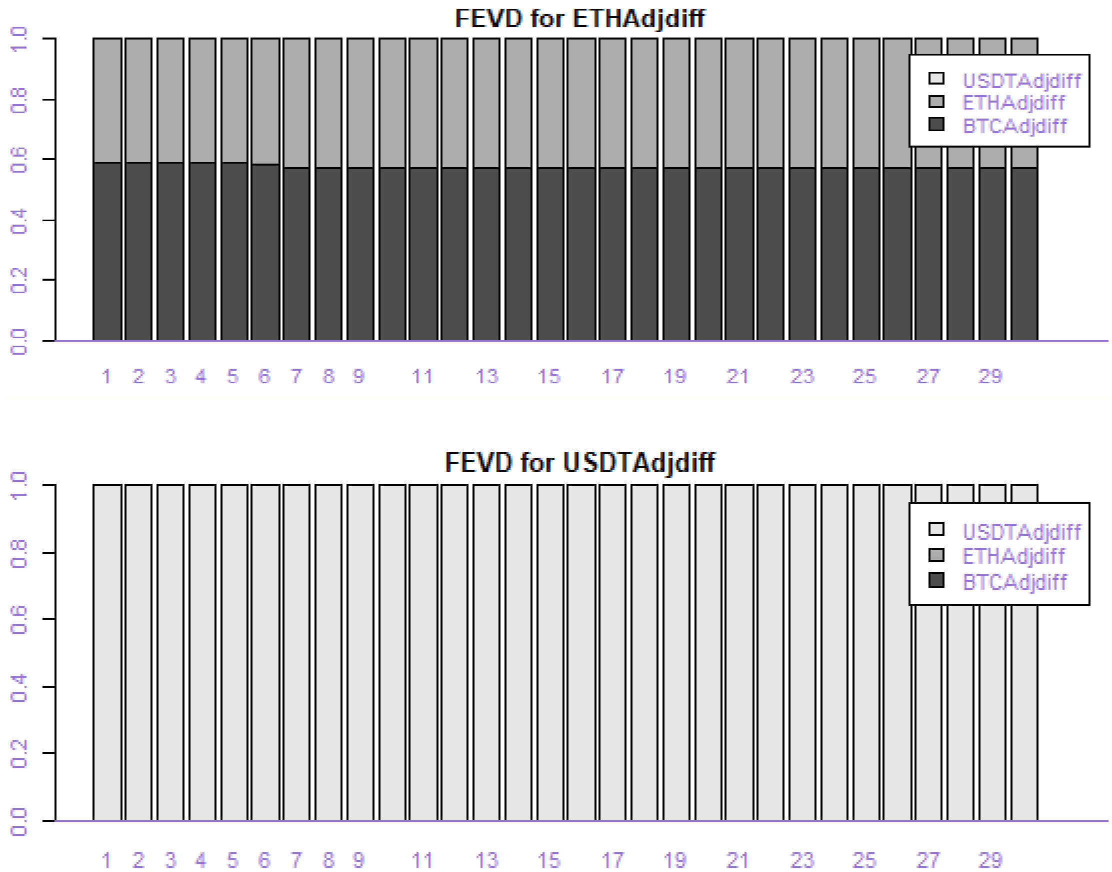


Figure 8 The forecast error variance decomposition

Source: author's estimates in R using the data of Yahoo finance

As follows from Figure 8, through the whole horizon of forecasting, the majority of the variation in stationarized BTC prices comes from the shocks to the BTC itself and only starting from the period 6, ETH contributes to the formation of BTC price by less than 1%. No influence of USTD on BTC was detected. Through the whole horizon of forecasting, about 40% of the variation in stationarized ETH prices came from the chocks to ETH itself and the remaining 60% was influenced by BTC. The shock to stationarized USDT prices affects only themselves. All the impacts are stable through the model horizon.

1.9. Johansen Cointegration Test

In order to analyze the long-run relationships among the cryptocurrencies Johansen Cointegration Test (Johansen, 1991) was run on Adjusted Close Prices of BTC, ETH and USDT. This test permits to determine if three or more time series are cointegrated. In contrast to correlation, cointegration is a long-term relationship to prices. Cointegration can be seen as a measure of similarity of assets in terms of risk exposure profiles. The prices of cointegrated assets are tethered due to the stationarity of the spread.

A stationary series can be formed by taking a linear combination of the underlying cointegrated series. The cointegration test assesses the validity of a cointegrating relationship, using a maximum likelihood estimates approach. The test statistics indicates a presence of at most 2 integrating relationships. Obtained coefficients permit to build a portfolio of the three currencies, as indicated in Table 11:

Table 11 Johansen Cointegration Test on Adjusted Close Prices

Test results	
Selection of lag	AIC(n) HQ(n) SC(n) FPE(n) 8 7 4 8
Test Results for the maximum eigenvalue	Test type: maximal eigenvalue statistic (lambda max) , with linear trend Eigenvalues (lambda): [1] **0.05218298** 0.01279724 0.00294845 Values of test statistic and critical values of test: test 10pct 5pct 1pct r <= 2 | 5.39 6.50 8.18 11.65 r <= 1 | 23.49 12.91 14.90 19.19 r = 0 | 97.76 18.90 21.07 25.75 Eigenvectors, normalised to first column: (These are the cointegration relations) BTC_Adjusted.l3 ETH_Adjusted.l3 USDT_Adjusted.l3 BTC_Adjusted.l3 1.000000e+00 1.0000 1.000000 ETH_Adjusted.l3 1.261644e+01 -14.6042 6.179327 USDT_Adjusted.l3 3.861888e+07 2637.8679 -9325.913866 Weights W: (This is the loading matrix) BTC_Adjusted.l3 ETH_Adjusted.l3 USDT_Adjusted.l3 BTC_Adjusted.d -1.486087e-05 1.259883e-02 -1.517032e-03 ETH_Adjusted.d 1.934659e-06 1.337080e-03 -6.268513e-06 USDT_Adjusted.d -4.612314e-09 1.011913e-10 2.842483e-10

Based on the results of Cointegration Tests, the following dynamic portfolio for the selected crypto currencies can be built:

Portfolio = 1.000*BTC+ 1.261644e+01*ETH+ 3.861888e+07*USDT

ADF test for the Portfolio	Augmented Dickey-Fuller Test data: Portfolio3 Dickey-Fuller = -5.8658, Lag order = 12, **p-value = 0.01** alternative hypothesis: stationary

In the Johansen test a cointegrating vector determines the number of shares (positive or negative) in the portfolio (i.e., spread). To buy or sell the spread, an investor longs or shorts the assets by the number of shares suggested by the cointegrating relationships.

The dynamics of a stationary portfolio of 3 currencies is illustrated below, Figure 9:

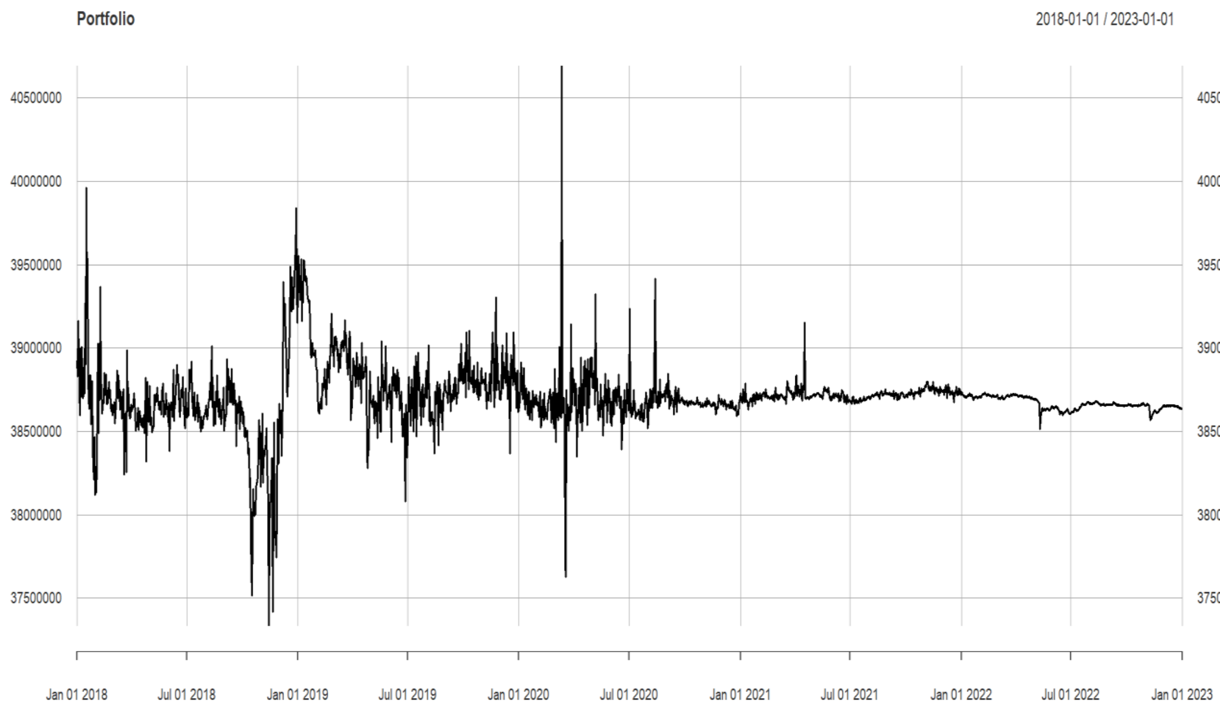


Figure 9 Dynamic of a portfolio of BTC, ETH and USDT

Source: author's estimates in R using the data of Yahoo finance

In a stationary process mean and variance remain constant over time. Any deviation from these expected values is a case for statistical abnormality, and hence is the case for pairs trading.

Investors can profit from purchasing this stationary portfolio when its price is low and get a profit when its price returns to the mean or crosses above a certain level. Similarly, an investor can short sell the spread when its price is high and get a profit when price reverts to the mean or crosses below a certain threshold.

Pairs trading with cointegrated assets permit to mitigate the potential losses and risks, since if one asset is underperforming, the other(s) may absorb the losses. It means, that regardless of the market conditions, a trader can get positive returns whenever price of one stock deviates from the mean. Hedging is realized by selling the overvalued security and purchasing the undervalued security, thereby, limiting the chances of loss.

1.10. Analysis of Long-Run Volatility Spillovers with GARCH-DCC Model

The volatility spillover models are widely used for the analysis of dynamic linkages among stocks since these models describe a shock's impact of one variable to another through the error terms. The volatility forecasting permit to estimate swings in cryptocurrency prices and log returns, which is useful for developing quantitative financial trading strategies, such as pairs trading that involves opening a simultaneous long and short positions on two crypto pairs. A pairs trading strategy is implemented based on the correlation between two pairs.

The volatility spillover model applied here is the multivariate GARCH model with dynamic conditional correlations (GARCH DCC), Engle and Sheppard (2001). The GARCH model is a continuation of the autoregressive conditional heteroscedasticity (ARCH) model that supports the conditional variance to change over time as a function of past errors leaving the unconditional variance constant (Bollerslev 1986). This type of model was selected because of the relatively easy parameters estimation and interpretation of results, in contrast with, for example BEKK-GARCH model. The main advantage of GARCH-DCC is that the number of parameters that are estimated in the correlation process are independent on the number of series that are to be estimated, so it gives a large computational advantage when estimating large covariance matrices (Engle 2002).

The idea of the models in this class is that the covariance matrix, H_t , can be decomposed into conditional standard deviations, D_t , and a correlation matrix, R_t . In the DCC-GARCH model both D_t and R_t are designed to be time-varying. The model equations can be specified as follows:

$$r_t = \mu_t + \alpha_t \quad (6)$$

$$a_t = H_t^{1/2} z_t \quad (7)$$

$$H_t = D_t \; R_t \; D_t \quad (8)$$

where r_t is a nx1 vector of log returns of n assets at time t;

a_t is a *nx1* vector of mean-corrected returns of n assets at time t; E[a_t]=0, Cov[a_t]=H_t ;

μ_t is a *nx1* vector of the expected value of the conditional r_t ;

H_t is a *nxn* matrix of conditional variances a_t at time t;

$H_t^{\frac{1}{2}}$ may be obtained by a Cholesky factorization of H_t ;

R_t is a *nxn* conditional correlation matrix of a_t at time t;

z_t is a *nx1* vector of iid errors such that E[z_t]=0, E[$z_t z_t^T$]=I;

D_t is a *nxn* matrix of conditional standard deviations of a_t at time t;

For the best model fit, MGARCH-DCC models were tailored specially for each cryptocurrency for the purpose of the best fit. The model was run on the log returns of each cryptocurrency. Different model settings were tested, and the best were selected based on the AIC information criteria, Table 12.

Table 12 The MGARCH-DCC models characteristics

	BTC	ETH	USDT
	univariate GARCH specification		
variance model	sGARCH	sGARCH	eGARCH
mean model	armaOrder = c(1,0), include mean	armaOrder = c(1,0), include mean	armaOrder = c(1,0), include mean
distribution model	sged	sstd	sstd
	DCC-GARCH specification		
multivariate distribution	mvt	mvt	mvt
The DCC autoregressive order	(1,1)	(1,1)	(1,1)

Source: author's estimates in R using the data of Yahoo finance

In the multivariate GARCH-DCC models that were built, both alpha1 and beta1 (the ARCH and GARCH coefficients) are jointly significant, and dcca1 and dccb1 (the short and long run persistence indicators) are jointly significant too, which means that the models are well fit to the data, Table 13.

Table 13 Results of GARCH-DCC modelling

```
*---------------------------------*
*          DCC GARCH Fit          *
*---------------------------------*
Distribution         :  mvt
Model                :  DCC(1,1)
No. Parameters       :  28
[VAR GARCH DCC UncQ] : [0+22+3+3]
No. Series           :  3
No. Obs.             :  1826
Log-Likelihood       :  17607.09
Av.Log-Likelihood    :  9.64
```

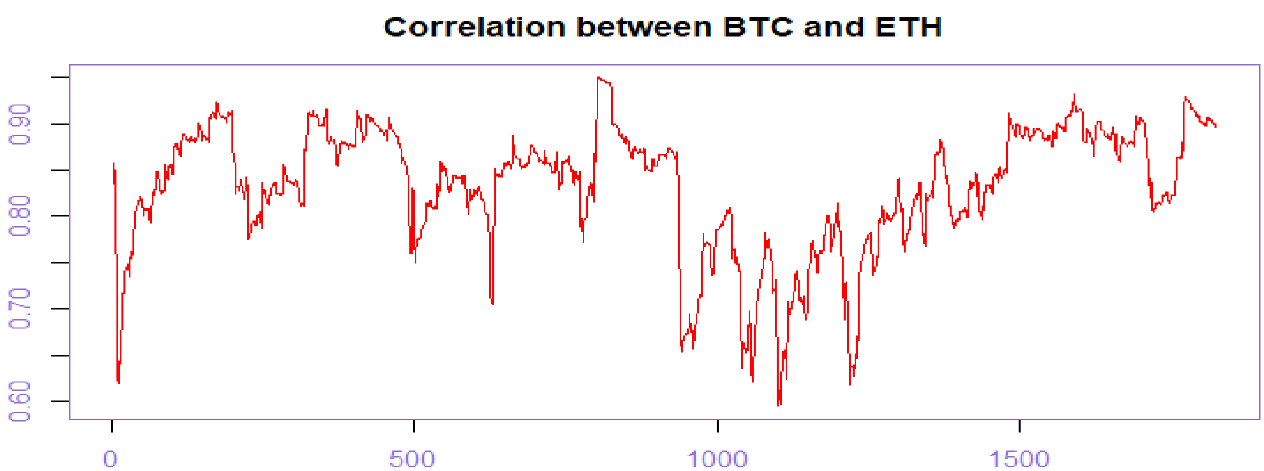
```
Optimal Parameters
------------------------------------
                 Estimate  Std. Error    t value Pr(>|t|)
[BTC].mu        -0.000108    0.000002  -45.02670 0.000000
[BTC].ar1       -0.076700    0.002852  -26.89616 0.000000
[BTC].omega      0.000026    0.000004    7.17055 0.000000
[BTC].alpha1     0.080240    0.007732   10.37788 0.000000
[BTC].beta1      0.907654    0.006976  130.11087 0.000000
[BTC].skew       0.966741    0.008528  113.36285 0.000000
[BTC].shape      0.882343    0.038643   22.83297 0.000000
[ETH].mu         0.000697    0.000987    0.70652 0.479867
[ETH].ar1       -0.069290    0.021259   -3.25934 0.001117
[ETH].omega      0.000177    0.000063    2.81541 0.004871
[ETH].alpha1     0.133915    0.030149    4.44174 0.000009
[ETH].beta1      0.828894    0.036943   22.43687 0.000000
[ETH].skew       0.979882    0.027997   34.99954 0.000000
[ETH].shape      3.331902    0.125382   26.57395 0.000000
[USDT].mu        0.000005    0.000004    1.53320 0.125227
[USDT].ar1      -0.343100    0.027819  -12.33314 0.000000
[USDT].omega    -0.086651    0.007419  -11.68029 0.000000
[USDT].alpha1    0.147751    0.016594    8.90383 0.000000
[USDT].beta1     0.993834    0.000258 3847.62260 0.000000
[USDT].gamma1    0.294172    0.013192   22.30009 0.000000
[USDT].skew      1.056677    0.027530   38.38271 0.000000
[USDT].shape     3.093494    0.152846   20.23928 0.000000
[Joint]dcca1     0.026751    0.007170    3.73099 0.000191
[Joint]dccb1     0.964729    0.011369   84.85643 0.000000
[Joint]mshape    4.000000    0.337457   11.85336 0.000000

Information Criteria
---------------------
Akaike -19.254 Bayes -19.170 Shibata -19.255 Hannan-Quinn -19.223
```

Source: author's estimates in R using the data of Yahoo finance

Results of MGARCH-DCC modelling revealed significant positive historical correlation between BTC and ETH log returns along the model horizon, as can be seen in

Figure 10:



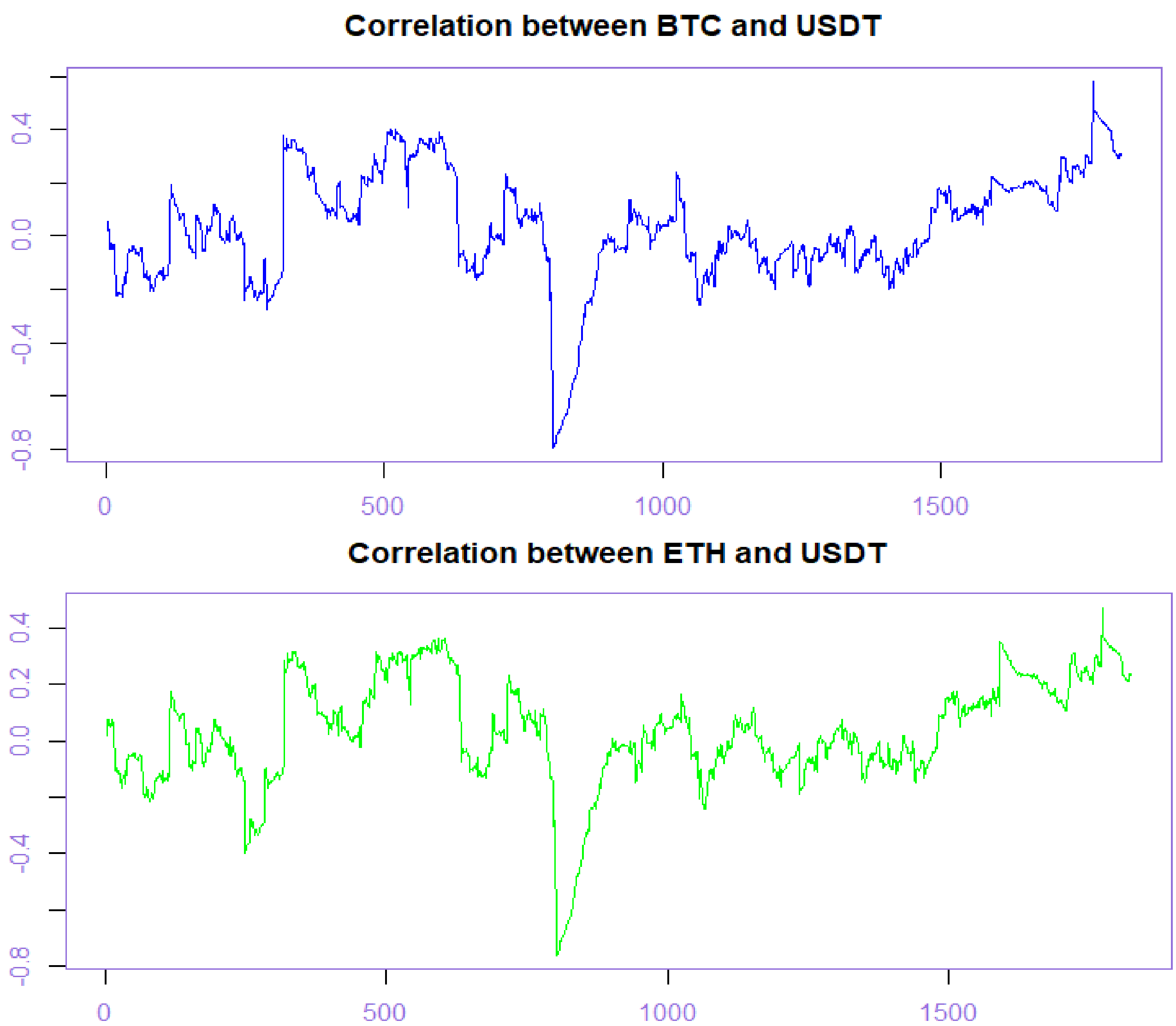


Figure 10 Conditional correlations between the log returns of Bitcoin, Ether and Tether

Source: author's estimates in R using the data of Yahoo finance

Even though the discrepancies between Bitcoin and Ethereum were significant in 2018, they have subsequently evened out. Historically ETH and BTC had high positive correlations which makes them the good candidates for the pairs trading. High positive correlation is the main driver behind the strategy's profits. Relying on the historical notion that the two securities will maintain a specified correlation, the pairs trade is effectuated when this correlation changes. When pairs deviate, a trader would take the long position in the underperforming security and sell short the outperforming security. When both pairs return to their historical correlation, profits are made from price convergence when the underperforming security regains its value and the outperforming security's price deflates. In such manner the disparities between the performance of BTC and ETH cryptocurrencies can be used for creating long and short orders. Since USDT does not have a significant correlation neither with Bitcoin nor with Tether (

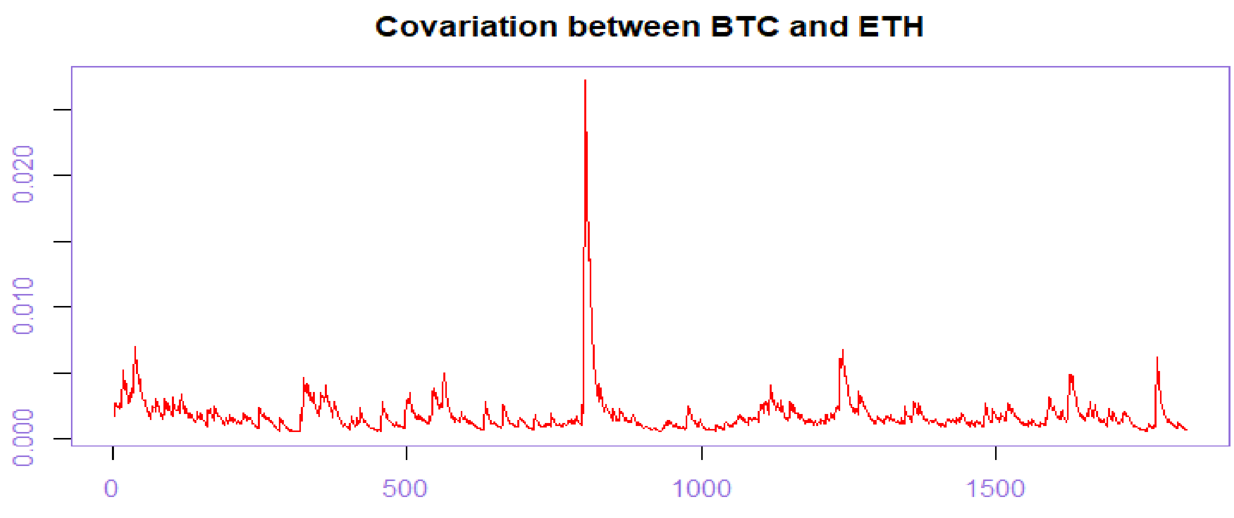
Figure 10), as a stablecoin, it can be used as a base currency that indicates the value of a crypto compared to USD or EURO.

For the log returns of other cryptocurrency pairs were observed insignificant positive or negative correlations. However, during the market crush in 2020 the correlations between ETH and USDT log returns and between BTC and USDT log returns became significantly negative. During the same period correlation between BTC and ETH log returns experienced a positive spike since prices of these two currencies have similar trends. That means that BTC and ETH react in a similar way to the external stimuli, while USDT reacts differently from BTC and ETH to the external shock (see

Figure 10), therefore, USDT is a good candidate for the portfolio diversification and hedging.

As for the long-run covariations, they are generally low positive between BTC and ETH log returns, and close to zero between the other pairs of cryptocurrencies. Only during the market crush in March 2020, a strong positive spike is observed in covariation between BTC and ETH log returns. During the same period, strong drop occurred between ETH and USDT log returns, and also between BTC and USDT log returns, which means that USDT reacts differently to an external shock than BTC and ETH do,

Figure 11:



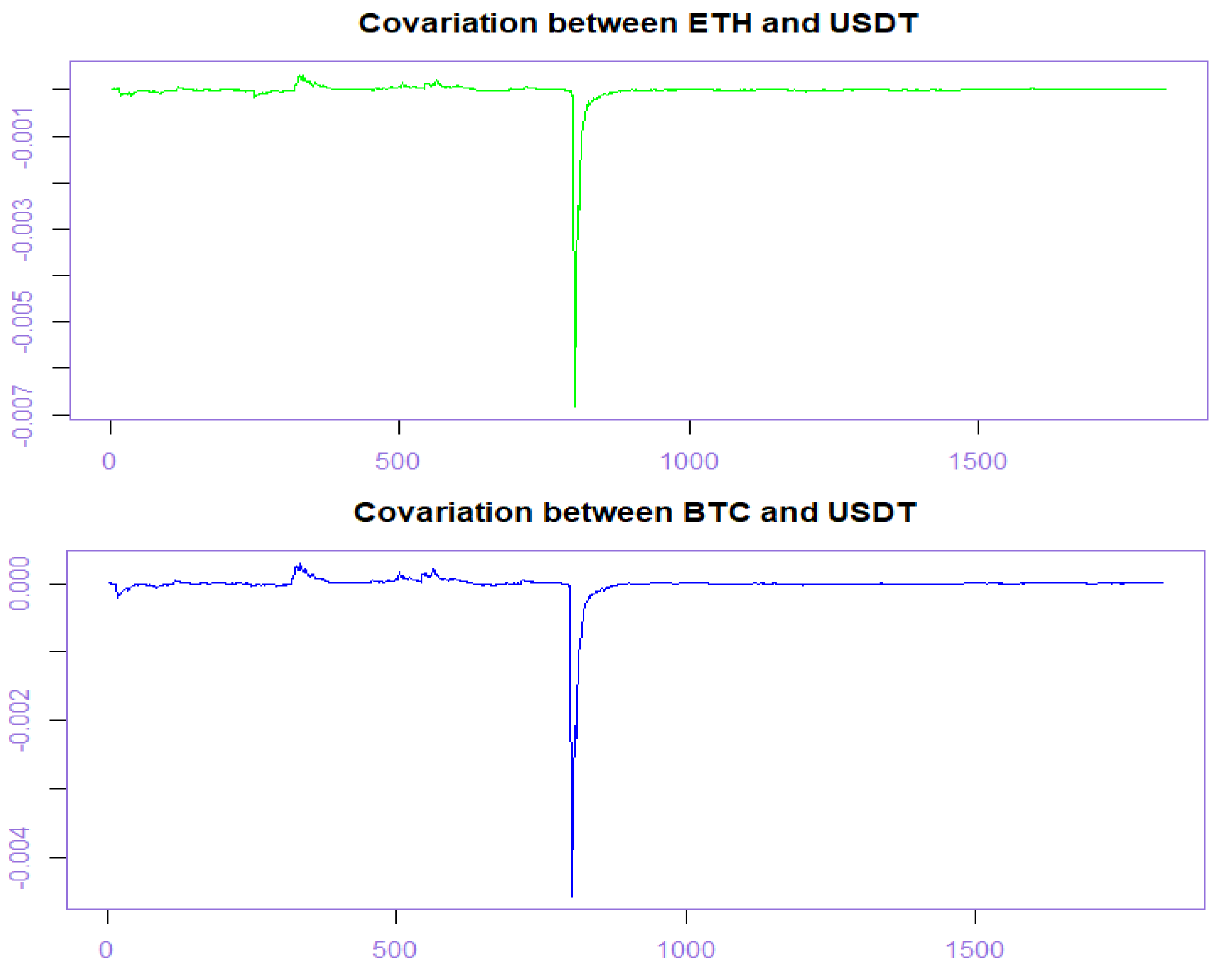


Figure 11 Conditional covariations between the log returns of Bitcoin, Ether and Tether

Source: author's estimates in R using the data of Yahoo finance

Summing up the first chapter, statistical tests conducted in R demonstrated that the three mainstream cryptocurrencies are highly non-linear, non-stationary with high degrees of autocorrelation, long memory, structural breaks and the first order of integration. Seasonal effects were not detected. Volatility of these cryptocurrencies exhibits clustering and has varied considerably over time.

Being a stable coin, price trajectory of USDT is characterized by a high frequency of oscillations of different amplitude around the mean. Although BTC and ETH were created in a different time and in a different manner, their prices follow quite similar predominantly rising trend of different magnitude.

The multivariate Granger causality test conducted in first differences of adjusted close prices demonstrated that ETH can be used in prediction of BTC and USDT, while USDT can be used for the prediction of ETH and BTC.

The forecast error variance decomposition gave evidence that the stationarized prices of Bitcoin and USDT are predominantly influenced by their own past values and are largely insensitive to the variations in other cryptocurrencies. In contrast, about 40% of the variation in stationarized ETH prices came from the shocks to ETH itself and the remaining 60% originated from BTC perturbations.

The long-run dynamic volatility spillovers investigated with the multivariate GARCH-DCC model revealed significant positive conditional correlations between BTC and ETH log returns along the forecasting period of 2018-2023. For the other pairs of cryptocurrencies were observed the fluctuating small positive and negative conditional correlations with dominance of positive values. High historical correlations between Bitcoin and Ether make them good candidates for pairs trading; in both pairs USDT can be used as a base currency indicating the value of the crypto compared to USD. Besides, USDT can be used for diversifying a portfolio consisting of BTC or ETH cryptocurrencies.

Johansen Cointegration test conducted in this study permitted to devise a stationary dynamic portfolio consisting of a certain number of Bitcoin, Ether and Tether shares suggested by a cointegrating vector. Investors can profit from purchasing this stationary portfolio when price is low and get a profit when its price returns to the mean or crosses above a certain level. Similarly, an investor can short sell the spread when its price is high and get a profit when price reverts to the mean or crosses below certain threshold.

In case of trading the individual cryptocurrencies, pairs trading or portfolio trading profits depend on the correctness of forecasted price moves. Forecasting models permit to predict cryptocurrency prices, helping investors to increase their profits. Considering high volatility of cryptocurrency markets, even small increases in model's precision can generate large profits to investors. The next chapter is dedicated to the determination of the most accurate price forecasting method for each cryptocurrency.

Chapter 2. Prediction of Cryptocurrencies Prices with Machine Learning Models

Predicting the evolution of cryptocurrency prices is a challenging task due to their nonlinearity, uncertainly and high volatility.

Various factors like public regulation of crypto markets, inflation, interest rate, technological developments such as improvements in blockchain technology, security concerns, institutional interest, high profile support, competition from other cryptocurrencies, and market sentiment along with the past trends combine to make the cryptocurrency markets very volatile. This makes it difficult to predict the future market scenarios with a high level of accuracy.

The possible solution lies in incorporating the machine learning techniques that are known to frequently perform better than the traditional time series forecasting methods. The machine learning algorithms study the data, discover a pattern in the past values, extrapolate the trend and predict the expected prices.

To combine the classic and machine learning approaches on financial time series forecasting, this work proposes a Hybrid ARIMA-Random Forest, Hybrid Arima-SVR and a Hybrid Arima-LSTM models were developed. The hybrid combination is expected to be successful since it permits decomposition of time series in linear and nonlinear trends that can be forecasted separately. According to this approach, linear part is predicted with ARIMA, and non-linear part, i.e., residuals of an ARIMA model, are forecasted with machine learning models that are known to outperform the traditional models in capturing nonlinear relationships. Consecutively, the combination of the two types of models should encompass with high accuracy both linear and nonlinear tendencies of the time series in the hybrid framework. An approach to establish a hybrid framework between ARIMA and neural networks approach was firstly proposed by Zhang (2003). Afterwards his idea was replicated in a number of studies that combined ARIMA with different ML models (e.g., Siami-Namini & Namin,2018, and Kumar et al, 2014).

Following the approach of Zhang (2003), the time series predictions with the Hybrid model ($\text{Prediction}_t^{Hybrid}$) were made by linearly combining the linear (Linear_t^{ARIMA}) and non-linear ($\text{Nonlinear}_t^{LSTM}$) components predicted by ARIMA and LSTM model:

$$\text{Prediction}_t^{Hybrid} = \text{Linear}_t^{ARIMA} + \text{Nonlinear}_t^{LSTM} \quad (9)$$

This way, the differences between the actual and predicted with ARIMA stock prices, i.e. the residuals act as training data for the ML models.

For all models used for the time series prediction, the split between the testing and training data was set equal to 19%[5].

It is important to emphasize that descriptive statistics of training data significantly differs from that of testing data. This is especially evident in case of USDT data, where test data have much lower variance, excess kurtosis, and different sign of skewness as compared with training data. Therefore, ML models may have difficulties with predicting testing data that are very different from the data the model was trained on.

Table 14 Descriptive statistics of training and testing data

	Training	**Testing**	**Training**	**Testing**	**Training**	**Testing**
	BTC		**ETH**		**USDT**	
Mean	-0.047	0.258	-0.091	0.490	0.053	-0.301
Variance	1.129	0.258	1.082	0.305	1.163	0.013
Standard Deviation	1.063	0.508	1.040	0.552	1.079	0.112
Skewness	1.094	1.229	1.358	1.297	0.646	-1.995
Excess Kurtosis	-0.352	0.252	0.509	0.447	12.498	7.299

Source: author's estimates in R using the data of Yahoo finance

2.1 Selection of Error Measures for Model Comparison

Among the error measures that are applied for the evaluation of models' predictive capacity the most popular are the Mean Absolute Error (MAE), the Mean Squared Error (MSE) and the Root Mean Square Percentage Error (RMSPE).

The formulas for MAE and MSE are given as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\overline{y_i} - y_i|, \quad (10)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y_i})^2, \quad (11)$$

where y_i are the observed values and $\overline{y_i}$ are the predictions. Apart from the evident simplicity, the standard measures of error measurements such as MAE and MSE both are scale dependent, and, therefore may produce biased

[5] Such split was necessary for the LSTM models, where the batch size must be evenly divisible in both the training and testing lengths. For the consistency of research, the same 19% split between the training and testing data was used for all models.

estimates, if compared between ARIMA model that uses as an input series of cryptocurrencies prices and machine learning models that use as an input the residuals generated by ARIMA model, Hyndman, R.J. and A.B. Koehler (2006).

Percentage errors such as MAPE (Mean Absolute Percentage Error) and RMSPE (Root Mean Square Percentage Error) have the advantage of being scale independent, since they are normalized by true observations, y_i:

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - \bar{y_i}|}{y_i}, \tag{12}$$

$$RMSPE = 100 \sum_{i=1}^{n} \frac{1}{n} \frac{|y_i - \bar{y_i}|}{y_i} \tag{13}$$

However, these measurements can get infinite or undefined values if the true observation is zero or close to zero. The MAPE has another disadvantage: it puts a heavier penalty on positive errors than on negative errors.

R-Squared, the coefficient of determination, is the square of correlation. It measures the proportion of the variance in the response variable that can be explained by the predictor variables in the model. When comparing multiple models or selecting the most appropriate model for a specific purpose, R-squared can be useful as it provides a standardized metric that ranges from 0 to 1. The closer to 1 the value of R-Squared, the better is the regression model as most of the variation of actual values from the mean value get explained by the regression model. If the dataset contains outliers or extreme values that might disproportionately affect the model's performance, R-squared can be a good metrics, since it is less sensitive to outliers. However, R-squared can become negative for the test dataset if squared errors are greater than total sum of squares.

The Mean absolute scaled error (MASE) is a measure for determining the effectiveness of forecasts generated through an algorithm by comparing the predictions with the output of a naïve, or random-walk forecasting approach. MASE value greater than 1 indicates the algorithm is performing poorly compared to the naïve forecast. The lower the MASE value, the lower the relative absolute forecast error, and the better the method. MASE is defined as:

$$MASE = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \bar{y_i}|}{\frac{1}{n-1} \sum_{k=2}^{n} \left| y_k^{naive} - \overline{y_{k-1}^{naive}} \right|} = \frac{MAE}{MAE^{naive}}, \tag{14}$$

where the denominator is the mean absolute error of the one-step "naive forecast method" on the training set which uses the actual value from the prior period as the forecast $\overline{y_{k-1}^{naive}}$. MASE is scale-independent, symmetric and does not suffer from the division by zero problem. Hyndman and Koehler (2006) recommend that the MASE becomes the standard when comparing forecast accuracy.

Taking into account the advantages and the shortcomings of the error measures, R-squared and MASE are used to determine models' goodness-of fit to the data and to compare models' performance.

2.2 Prediction of cryptocurrencies prices with ARIMA

After establishing the possible linkages between the crypto coins, cryptocurrency prices were forecasted with the Autoregressive Integrated Moving Average (ARIMA) model that serves as a benchmark for the comparison with the machine learning models.

This model and its extensions have been widely studied and applied for time series forecasting. ARIMA models are quite flexible in that they can represent several different types of time series, i.e., pure autoregressive (AR), pure moving average (MA) and combined AR and MA (ARIMA) series (Box, Jenkins and Reinsel, 2008).

ARMA(p, q) model can be expressed with the following equation:

$$x_t = a + \sum_{i=1}^{p} \varphi_i \, x_{t-i} - \sum_{j=1}^{q} \theta_j \, \varepsilon_{t-j} + \varepsilon_t \qquad (15)$$

In this equation the AR(p) part of the model ($\text{i.e.} \sum_{i=1}^{p} \varphi_i \, x_{t-i}$) describes the time series x_t at time t as a linear regression of the previous p observations, where ε_t is the white noise residual term and φ_i are the AR coefficient. The MA(q) part of the model ($\text{i.e.} \sum_{j=1}^{q} \theta_j \, \varepsilon_{t-j} + \varepsilon_t$) uses dependency between residual errors to forecast values in the next period, where α is the intercept and θ_i is the MA coefficient.

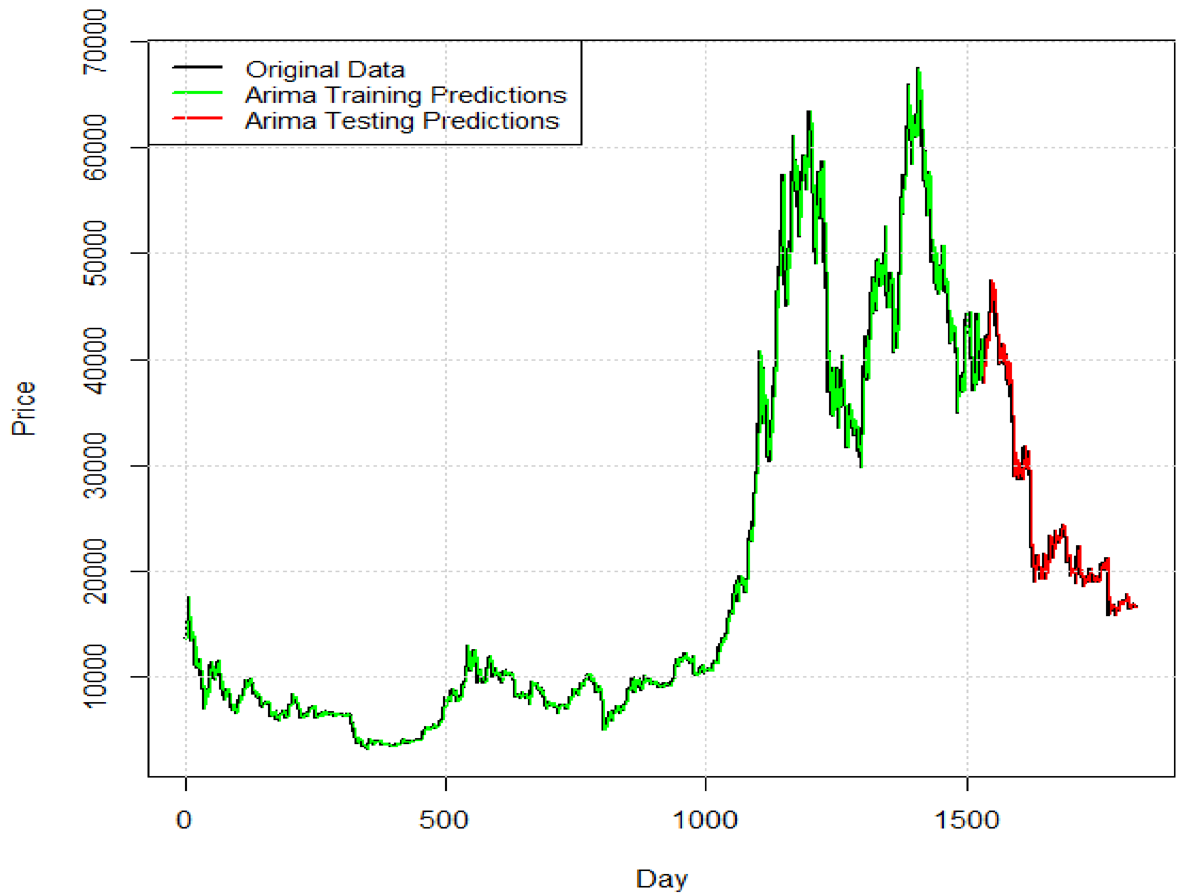
The complete ARIMA model is a generalization of the ARMA model by including integrated components, that are useful when the data are non-stationary. The ARIMA (p, d, q) represent the integration order for AR, MA, and differencing components. The model is well recognized for its forecasting accuracy. However, as a linear model, ARIMA encountered several limitations with non-linear problems.

The ARIMA model built in this study was based on the "one-step forecasts without re-estimation" approach of "Rolling Forecasting Origin" proposed by Hyndman (2014), Hyndman and Athanaspoulos (2014). According to this approach, when evaluation of a single set of training data is completed, the rest of data set one-step forecasts are generated.

In contrast, with "Multi-step forecasts with re-estimation" the model is reconstructed each time when a new observation is added to the training set , so that a "walk-forward model validation" is run.

An *auto.arima* function was applied to determine the best model fit for the cryptocurrencies under study. This function returns the best fit ARIMA model according to either AIC, AICc or BIC value. The *auto.arima* makes the dataset stationary and runs a search over possible model within the order constraints.[6]

As shown in Figure 12 below, ARIMA model has a close fit to the data with almost unitary R^2 both with training and testing predictions in BTC price forecasting:



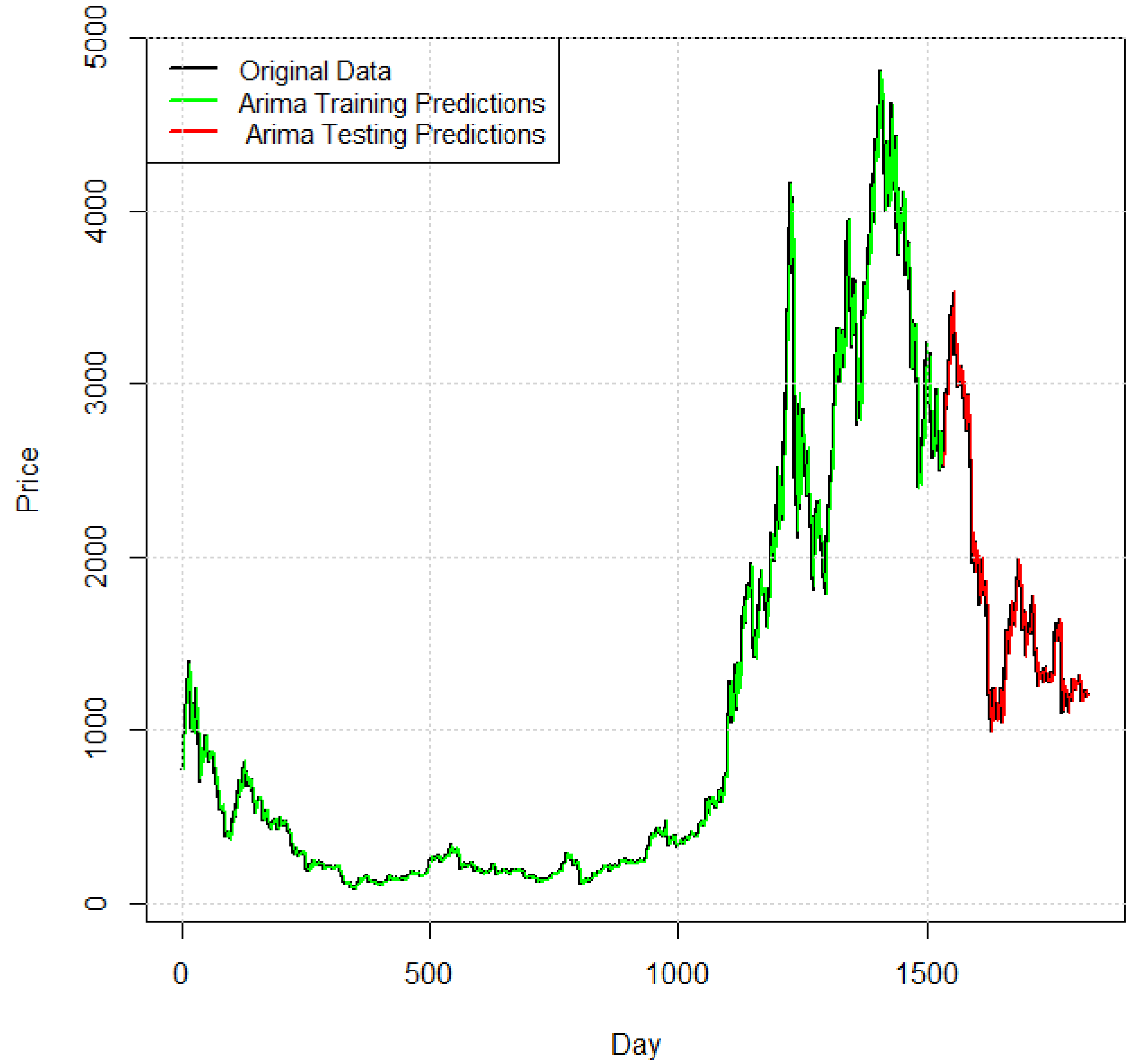
Model performance	R2 Training:	0.9967
	R2 Test:	0.9907
	MASE Training:	0.99765
	MASE Test:	1.00566

Figure 12 BTC price forecasting with ARIMA (1,1,0) model

Source: author's estimates in R using the data of Yahoo finance

[6] https://www.rdocumentation.org/packages/forecast/versions/8.21/topics/auto.arima
In this study were taken into account the possible issues caused by severe overparameterization, which might happen with tools such as *auto.arima*

As shown in Figure 14, the ARIMA model presents high accuracy for the ETH price prediction both with training and testing sets of data with R^2 exceeding 98%:

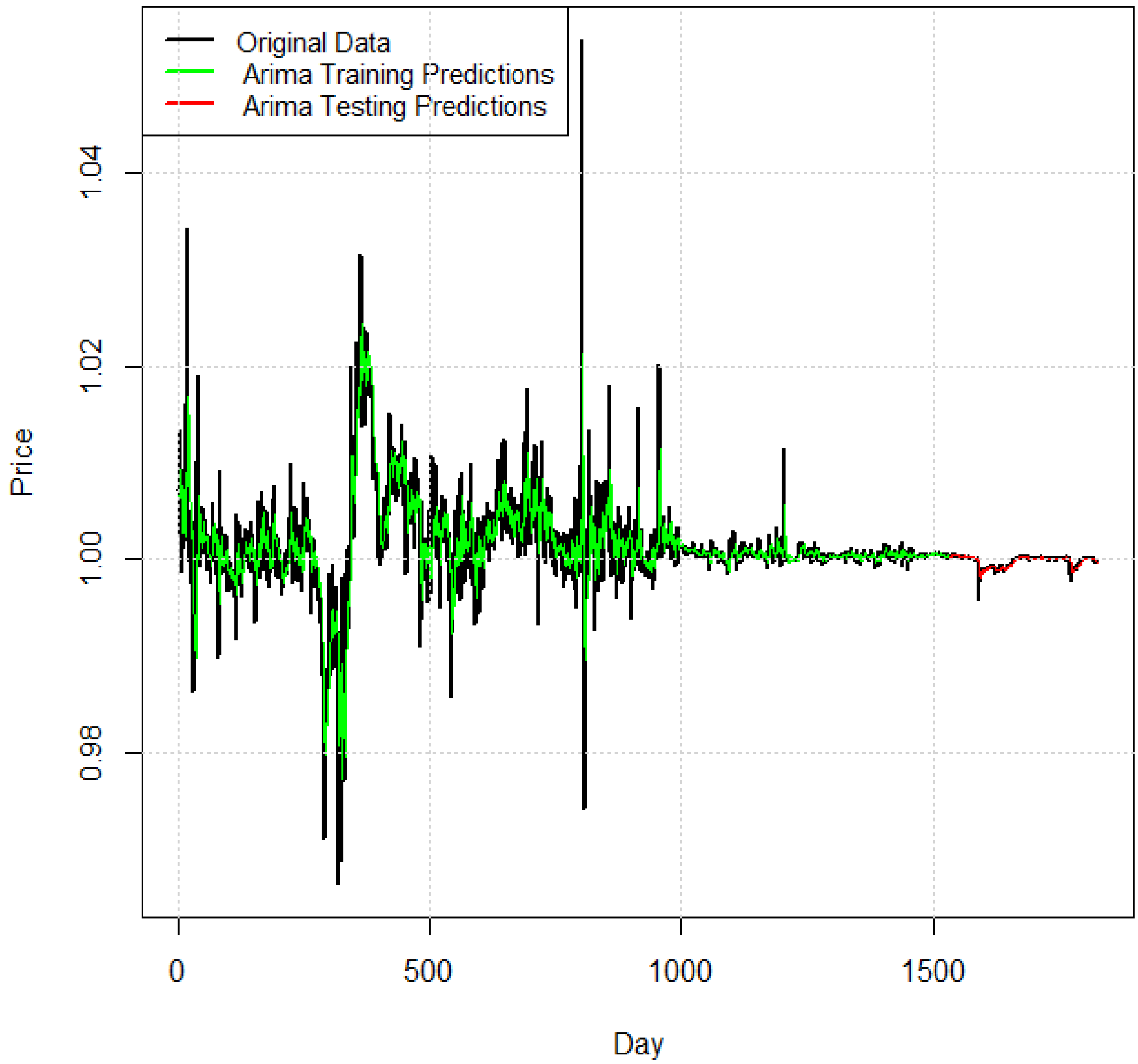


Model	R2	Training:	0.9959
performance	R2	Test:	0.98666
	MASE	Training:	0.98698
	MASE	Test:	1.00393

Figure 13 ETH price forecasting with ARIMA (2,1,2) model

Source: author's estimates in R using the data of Yahoo finance

Nevertheless, the ARIMA model has poor performance with the USDT data, especially in the training predictions. This outcome can be explained by much higher oscillations during the period that was selected for training when compared with the testing period in Figure 14:



Model performance	R2 Training:	0.55607
	R2 Test:	0.71598
	MASE Training:	0.87267
	MASE Test:	1.11748

Figure 14 USDT price forecasting with ARIMA (2,1,3) model

Source: author’s estimates in R using the data of Yahoo finance

Even though ARIMA had quite good values of MASE on a training set, that does not mean that ARIMA had good performance on training data, but that the naïve method of forecasting would perform worse than ARIMA, Figure 14.

As shown in Table 15,BTC, ETH and USDT residuals from all Arima models are most probably a white noise process as they can be represented with an ARIMA(0,0,0) function.

Table 15 Residuals from ARIMA model cryptocurrency price prediction

Series: residualsBTC
ARIMA(0,0,0) with zero mean sigma^2 = 1006711: log likelihood = -15727.11 AIC=31456.22   AICc=31456.23   BIC=31461.77
Series: residualsETH
ARIMA(0,0,0) with zero mean sigma^2 = 6279: log likelihood = -10580.88 AIC=21163.75   AICc=21163.76   BIC=21169.26
Series: residualsUSDT
ARIMA(0,0,0) with zero mean sigma^2 = 0.00001145: log likelihood = 7800.88 AIC=-15599.76   AICc=-15599.76   BIC=-15594.25

Source: author's estimates in R using the data of Yahoo finance

2.3 Prediction of Cryptocurrencies Prices with Random Forest Models

Random Forest Regression is a supervised learning algorithm (Ho,1995). These models are considered powerful and accurate forecasting tools. They usually perform well on large data series with non-linear relationships, handle well interactions and outliers. Random Forests use ensemble learning technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. In other words, a Random Forest model operates by constructing several decision trees during training phase and outputs the mean of the classes as the prediction of all trees. That means that in a regression problem, the range of predictions a Random Forest can make is bound by the highest and lowest labels in the training data. This behavior can become problematic when the training and prediction inputs differ in their range and/or distributions. This is called covariate shift and it is difficult to handle for the most of models.

Random Forest algorithm operates through the following steps:

1. Randomly picks *k* data points from the training set.
2. Builds a decision tree associated with these *k* data points.
3. Chooses the number *N* of trees to build and repeat the two previous steps.
4. For a new data point, one of the *N*-tree trees predicts the value for the data point in question and assigns the new data point to the average across all of the predicted *y* values.

In this exercise, Random Forest models were run with respect to the following technical indicators as features:

- day-, 10 day- and 20 day- Simple Moving Average (SMA5day,SMA10day and SMA20day, correspondingly). This indicator calculates the average of a selected range of prices, usually closing prices, by the number of periods in that range;
- 5 day-, 10 day- and 20 day- Exponential Moving Average (correspondingly, EMA5day,EMA10day and EMA20day). This indicator is similar to Simple Moving Average (SMA), measuring trend direction over a period of time, however, EMA gives more weight to the recent data;
- 14 day -Relative Strength Index (RSI14day)calculates average price gains and losses over a given period of time;
- 2 day -Price Rate of Change (Roc2day) is a momentum-based technical indicator that measures the percentage change in price between the current price and the price a certain number of periods ago;
- 2 day-day Momentum (Momentum2day) is a difference between the current closing price and a closing price "n" periods ago.

Random Forest is known to handle well interactions, which means that correlations among the technical indicators are not expected to affect the model performance.
To construct technical indicators, the start date of all time series was shifted to November 1, 2017.

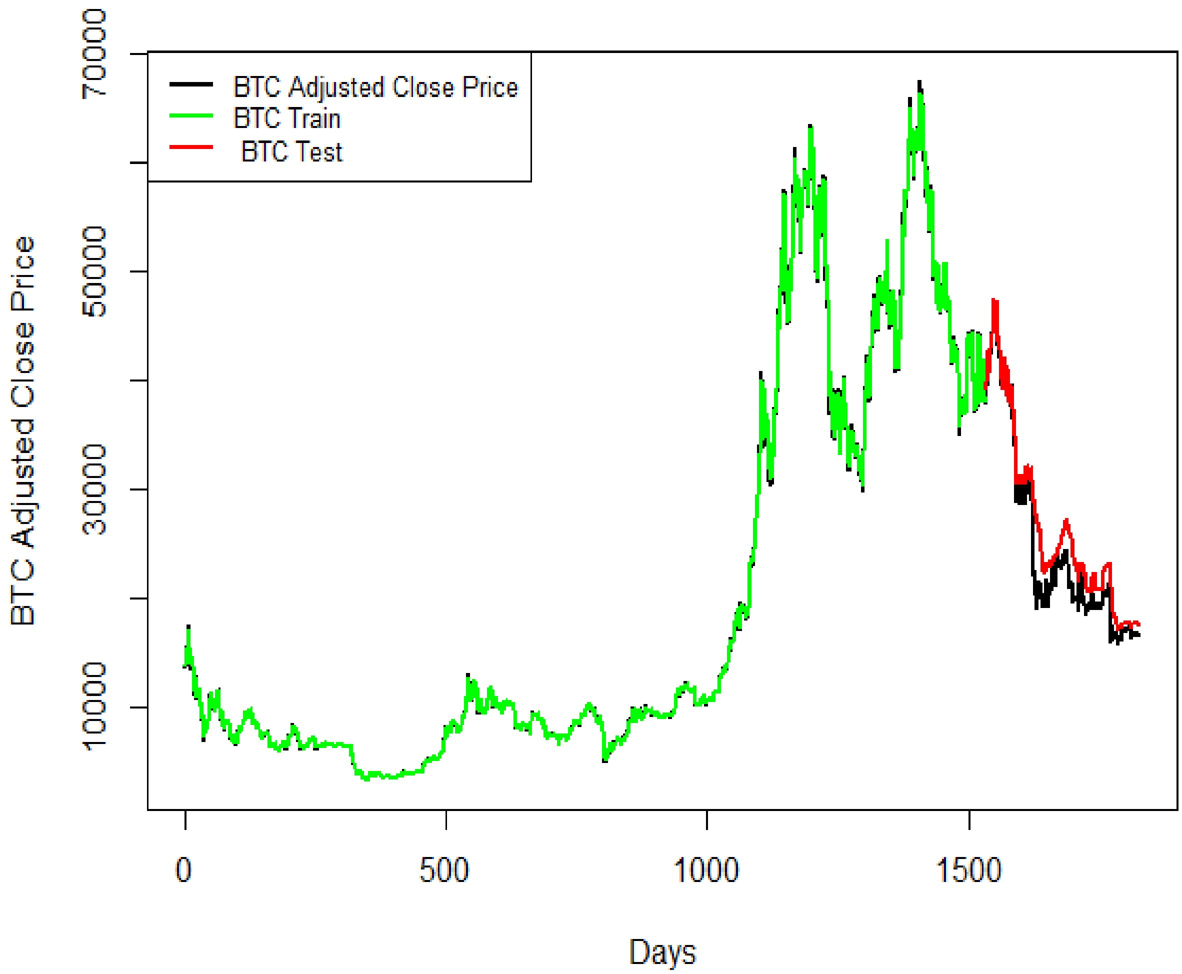
The architecture of the Random forest models is provided below, Table 16:

Table 16 Architecture of Random Forest models:

ntree	number of trees grown	500(BTC, ETH,USDT)
mtry	number of predictors sampled for splitting at each node	3 (BTC, ETH),9 (USDT)
R Package used		Caret(BTC, ETH,USDT)

Source: author’s estimates in R using the data of Yahoo finance

With overall good model performance (Figure 15), Random Forest model yields a bit higher accuracy of BTC price predictions with training data and a bit lower accuracy with test data:



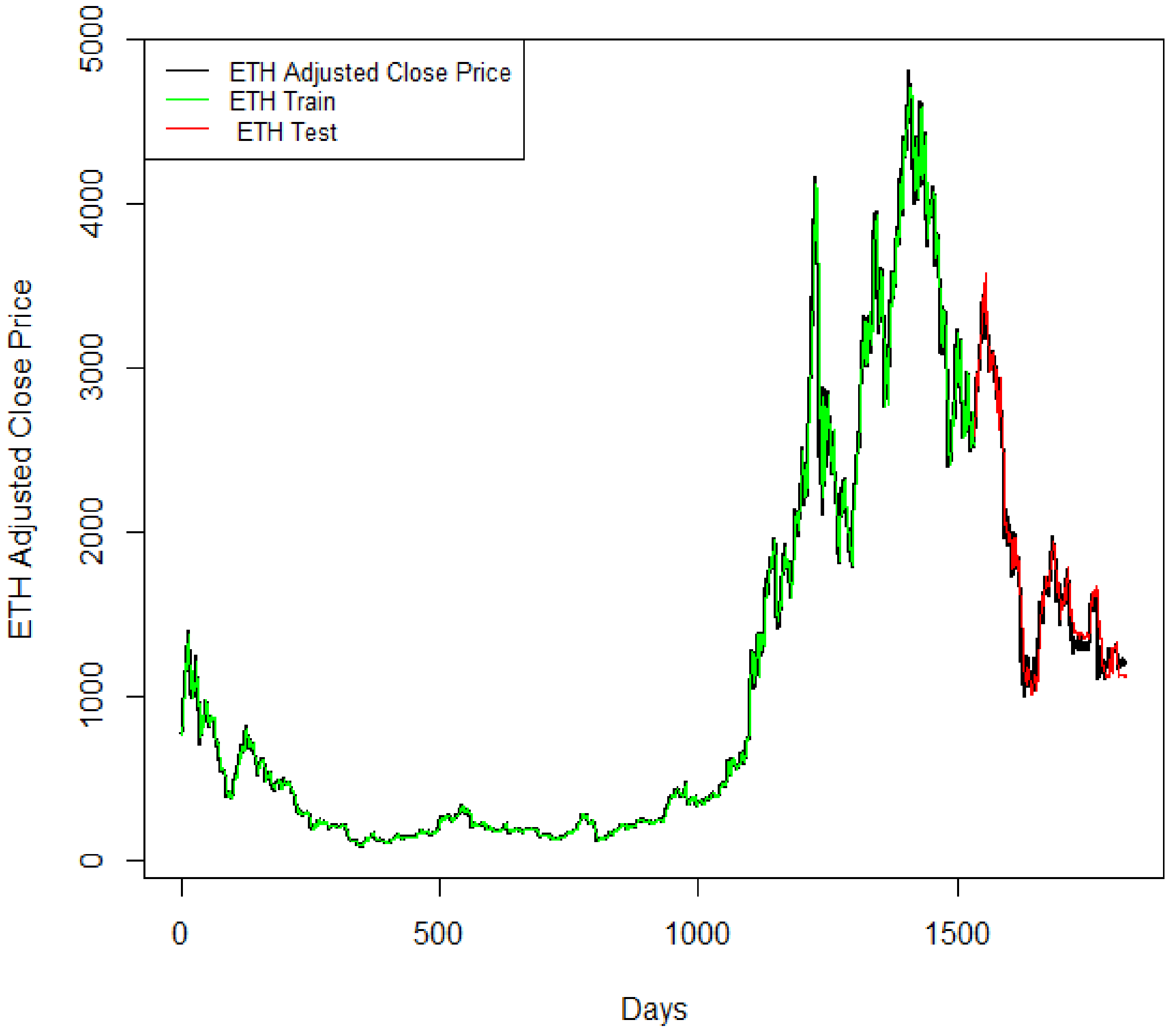
Model performance	R2 Training:	1
	R2 Test:	0.926
	MASE Training:	0.201
	MASE Test:	3.292

Figure 15 BTC Price Forecasts with Random Forest model

Source: author's estimates in R using the data of Yahoo finance

Despite the high values of R2 in the test set predictions of BTC, the MASE measure amounts to 3.29. This does not undermine the predictive capacity of the Random Forest model, such high values of MASE indicate that naïve approach would give better results than Random Forest on a test set of data (or on a part of it), see Figure 15.

Figure 16 shows that Random Forest model yields high accuracy of ETH price predictions both with training and test data:

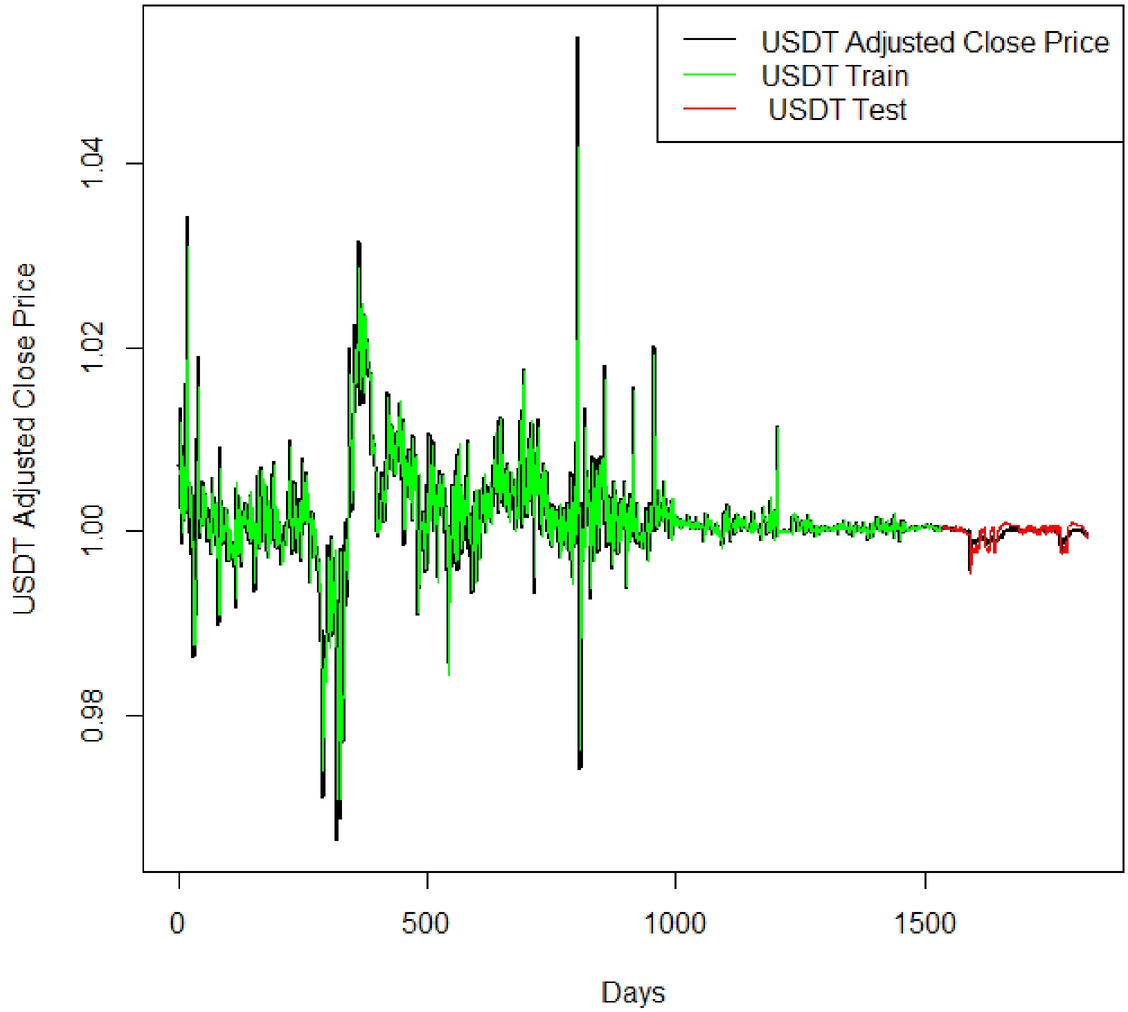


Model performance	R2 Training: 1 R2 Test: 0.99 MASE Training: 0.191 MASE Test: 0.955

Figure 16 ETH Price Forecasts with Random Forest model

Source: author's estimates in R using the data of Yahoo finance

Performance metrics on Figure 17 show that the Random Forest model has 99% accuracy on ETH price predictions with training data (R2 metrics) and lower accuracy of 68% (R2 metrics) with test data:



Model performance	R2	Training:	0.990
	R2	Test:	0.675
	MASE Training:		0.108
	MASE Test:		4.493

Figure 17 USDT Price Forecasts with Random Forest model

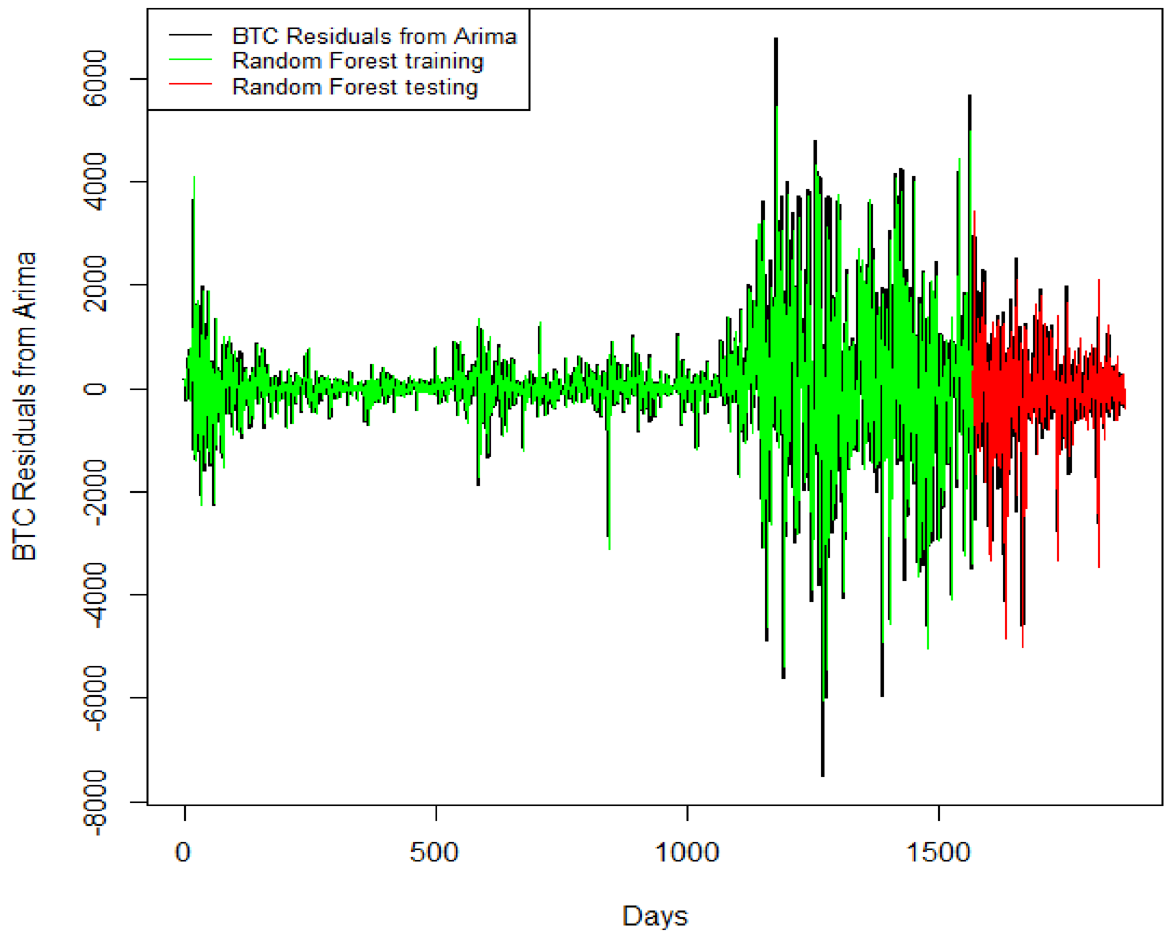
Source: author's estimates in R using the data of Yahoo finance

2.4 Prediction of Cryptocurrencies Prices with Hybrid ARIMA-Random Forest Models

The Hybrid ARIMA-Random Forest model developed in this study permits separate forecasting of linear trends with Arima modelling and of nonlinear trends with Random Forest modelling. The differences between the actual and predicted with

ARIMA stock prices (shown on Figure 12–Figure 14), i.e. the residuals act as training data for the Random Forest model. The architecture of Random Forest model was kept the same as was described in Table 16.

Figure 18 illustrates the fact that Random Forest model has high performance with training data and very low accuracy with test data when forecasting BTC residuals of ARIMA model. Such low performance can be explained by the fact that BTC residuals have a high frequency of oscillations with rapid changes in amplitude:

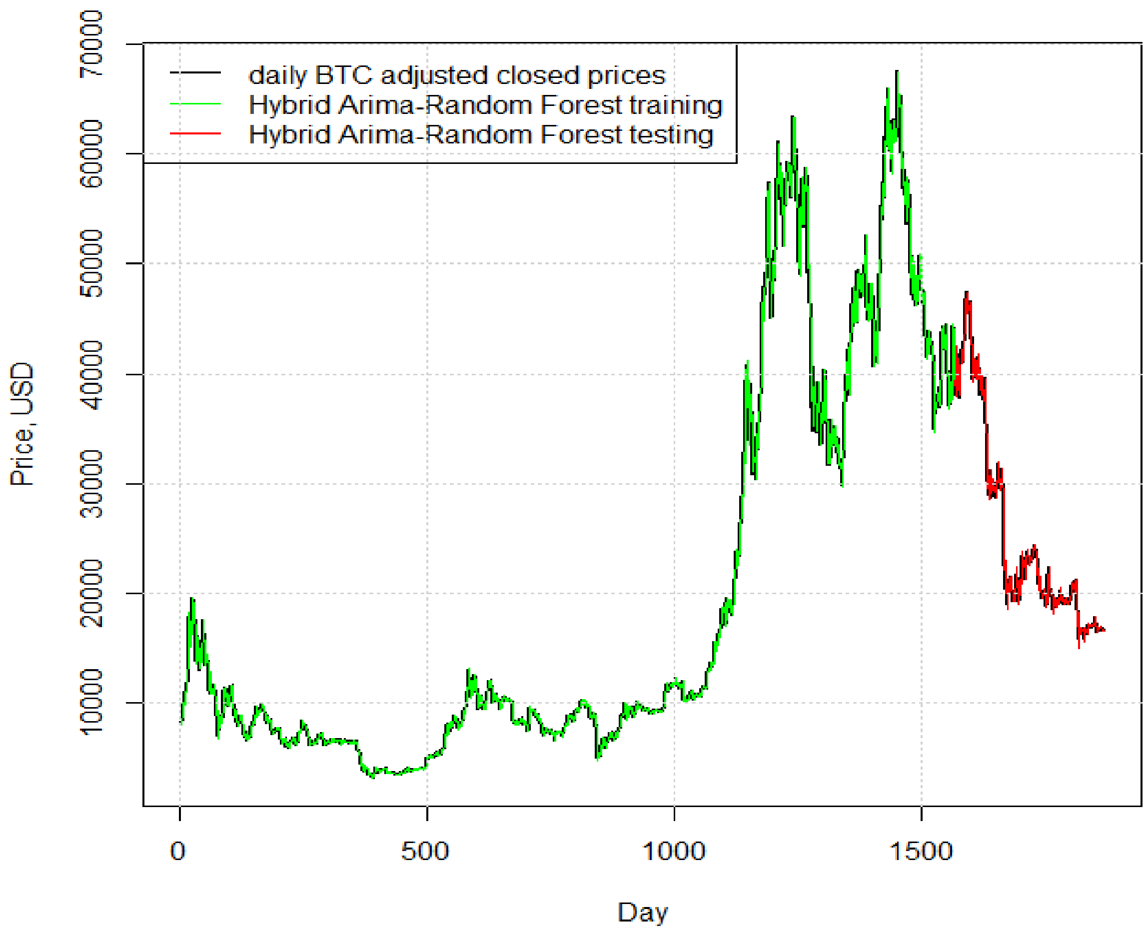


Model performance	$R2$ Training:	0.987
	MASE Training:	0.063
	$R2$ Test:	0.948
	MASE Test:	0.142

Figure 18 Prediction of BTC residuals from ARIMA with the Random Forest model

Source: author's estimates in R using the data of Yahoo finance

As shown on Figure 19, the Random Forest model has almost unitary accuracy of BTC price prediction, both with training and test data:

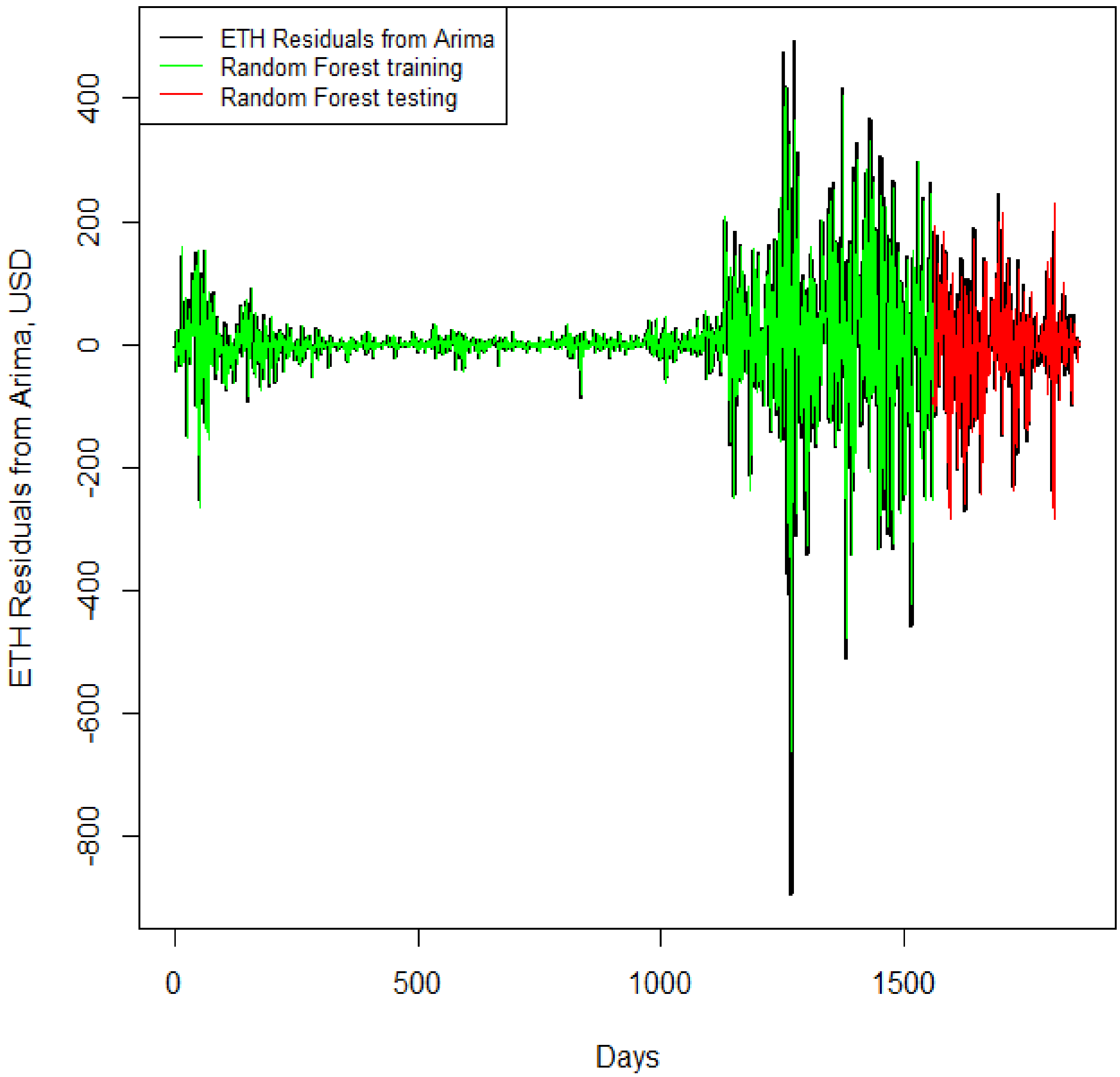


Model performance	R^2 Training: 1 MASE Training: 0.094 R^2 Test: 0.999 MASE Test: 0.233

Figure 19 Prediction of BTC Adjusted Close Prices with the Hybrid ARIMA-Random Forest model

Source: author's estimates in R using the data of Yahoo finance

The Random Forest model has a good accuracy of predicting ETH residuals from ARIMA both with training and test data (Figure 20):

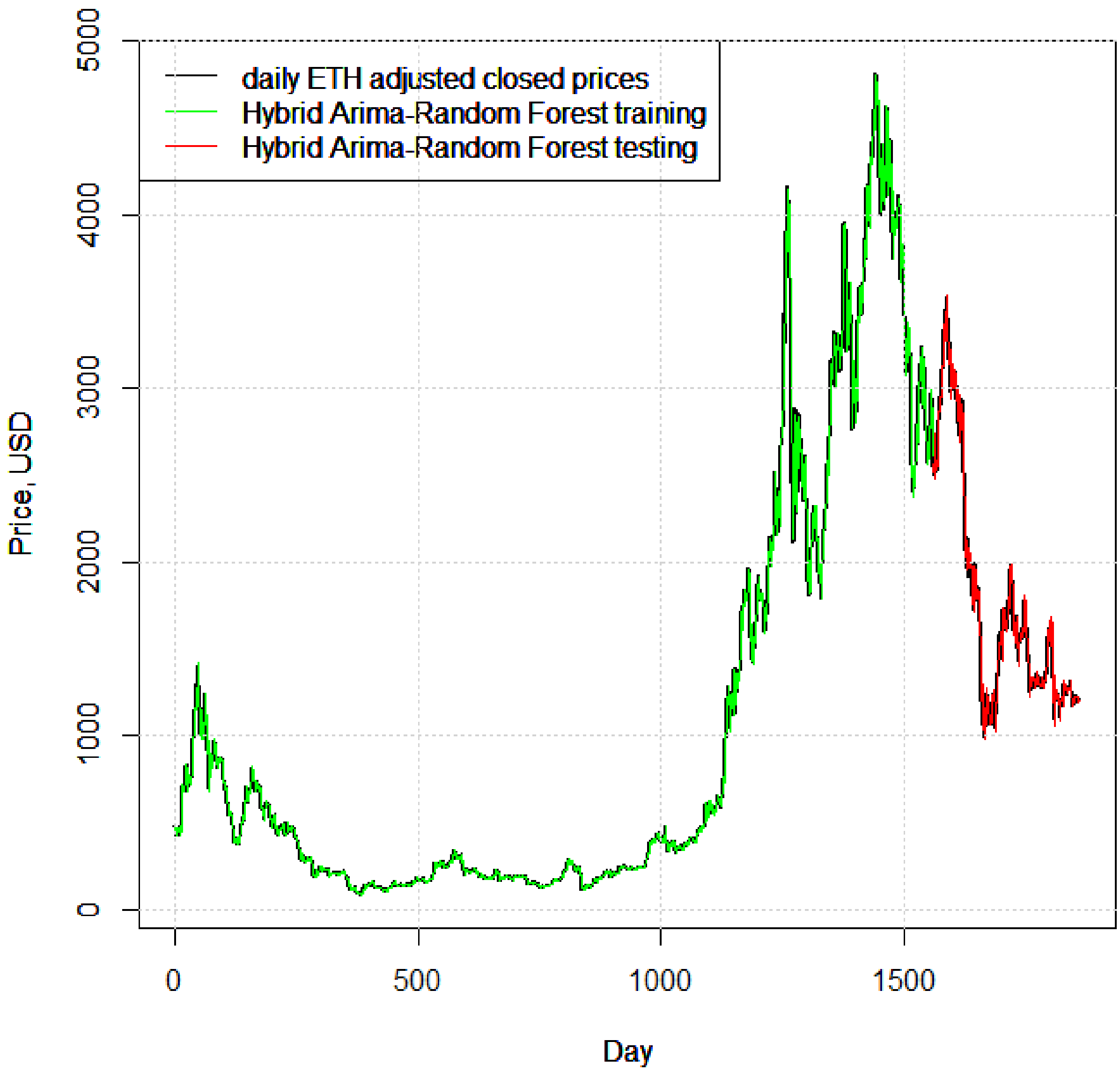


Model performance	R2	Training: 0.981
	R2	Test: 0.947
	MASE	Training: 0.067
	MASE	Test: 0.154

Figure 20 Prediction of ETH residuals from ARIMA with the Random Forest model

Source: author's estimates in R using the data of Yahoo finance

As shown on Figure 21 the Random Forest model has almost unitary accuracy of ETH price prediction both with training and test data:

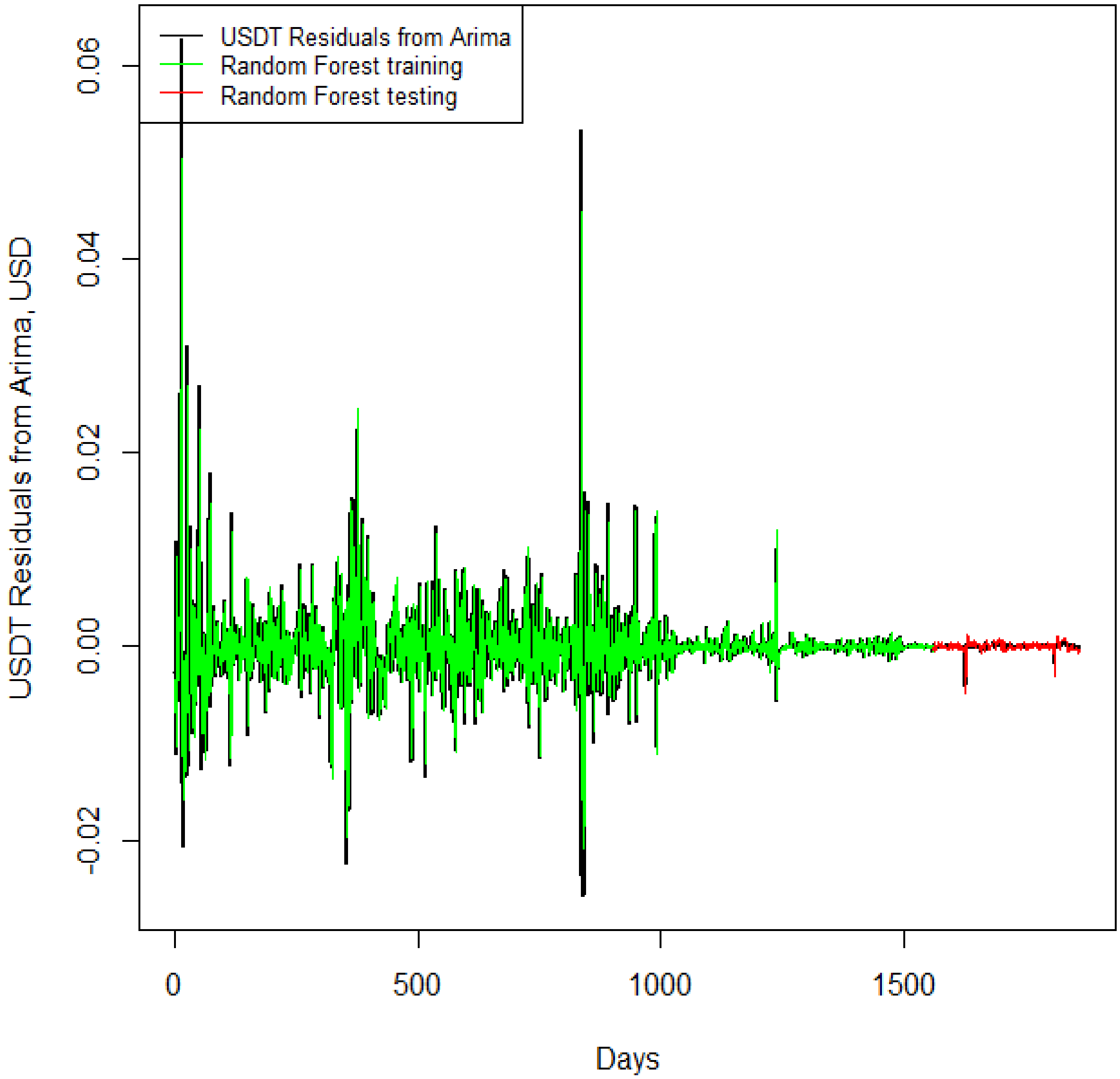


Model performance	$R2$ Training: 1	
	$R2$ Test:	0.999
	MASE Training:	0.096
	MASE Test:	0.222

Figure 21 Prediction of ETH Adjusted Close Prices with the Hybrid ARIMA-Random Forest model

Source: author's estimates in R using the data of Yahoo finance

The Random Forest model has poor accuracy of predicting USDT residuals from the ARIMA model with test data and a high accuracy with training data, since the latter are much more linear than the former (Figure 22):

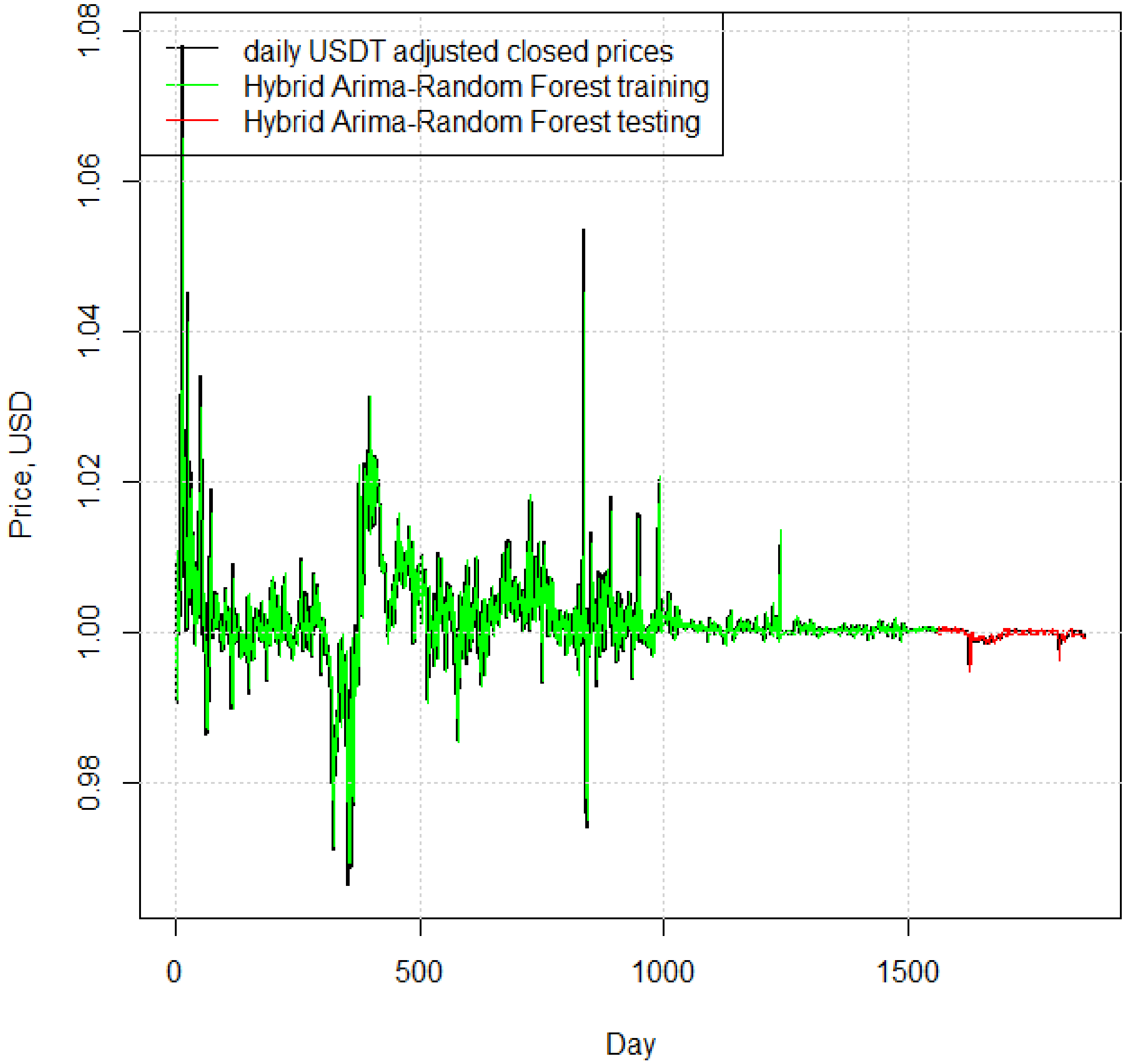


Model performance	R2 Training: 0.977 R2 Test: - 0.122 MASE Training: 0.076 MASE Test: 1.756

Figure 22 Prediction of USDT residuals from ARIMA with the Random Forest model

Source: author's estimates in R using the data of Yahoo finance

The hybrid ARIMA-Random Forest model has almost 99% accuracy of USDT price prediction on train data and almost 70% accuracy with test data (Figure 23):



Model performance	R2 Training: 0.989
	R2 Test: 0.692
	MASE Training: 0.093
	MASE Test: 1.991

Figure 23 Prediction of USDT prices with Hybrid ARIMA-Random Forest model

Source: author's estimates in R using the data of Yahoo finance

The above result is the clear illustration of higher performance of the hybrid framework as compared with individual models, see Figure 14 and Figure 17.

2.5 Prediction of Cryptocurrencies Prices with Support Vector Regression Models

Support Vector Regression (SVR) models work on similar principles as Support Vector Machine (SVM) classification but predicts real values rather than classes. The basic idea behind SVR (Vapnik, 1995) is to find a function that has at most ε deviation from the actually obtained targets for all the training data, and at the same time is as flat as possible. In other words, SVR does not care about errors as long if they are less than ε, but it will not accept any deviation larger than this. A major benefit of using SVR is that it is a non-parametric technique, which means that its output does not depend on distributions of the dependent and independent variables. Instead, the SVR technique depends on kernel functions that are used to map the original input data points into higher dimensional feature spaces. The most used kernel transformations are polynomial kernel and radial kernel. This way an hyperplane can be found out even if the data points are not linearly separable in the original input space. The hyperplane is defined by the observations that lie within a margin optimized by a cost hyperparameter. These observations are called the support vectors. Margin is the distance between the support vector and hyperplane, and support vectors are the closest data points to the hyperplane. Hereby, margin maximization and misclassification fines are balanced by the regularization cost parameter, low values of which penalize samples inside the margins less than its higher values. Higher values of cost parameters can be used to penalize regression in order to avoid over fitting.

In this research for each cryptocurrency SVR models were cross-validated and tuned with respect to the alternative types of kernel functions and the values of hyperparameters. Performance of Linear Kernel, Polynomial Kernel and Radial Basis Function (RBF) Kernel were tested. The lowest RMSE were provided by the Support Vector Machines with Radial Basis Function Kernel (*method = 'svmRadialSigma'* of the caret package of R) that tunes over the alternative values of cost parameter and the RBF kernel parameter sigma. RBF kernel can be mathematically represented with the following formula:

$$K(x, x_k) = \exp\left(\frac{-\|x_k - x_l\|^2}{\sigma^2}\right), \tag{16}$$

where σ is the variance hyperparameter and $\|x_k - x_l\|$ is the Euclidean distance between two points x_k and x_l.

Cost parameter controls the flexibility of the model and its ability to work with unseen data. Large values of the cost parameter lead to the smaller margins and less flexible model, which increases the possibility of model overfitting. In contrast, small values of the cost parameter lead to larger margins and higher level of misclassification, which can also result in poor model performance.

In the case of RBF kernels the sigma parameter that determines the bandwidth of kernel function must be tuned. The sigma parameter controls the level of non-linearity introduced in the model. If the sigma value is low, the SVM algorithm would ignore the points that are far from the decision boundary. Large values of sigma permit the SMV to consider the data points that are far from the decision boundary, thus increasing the level of non-linearity of the model.

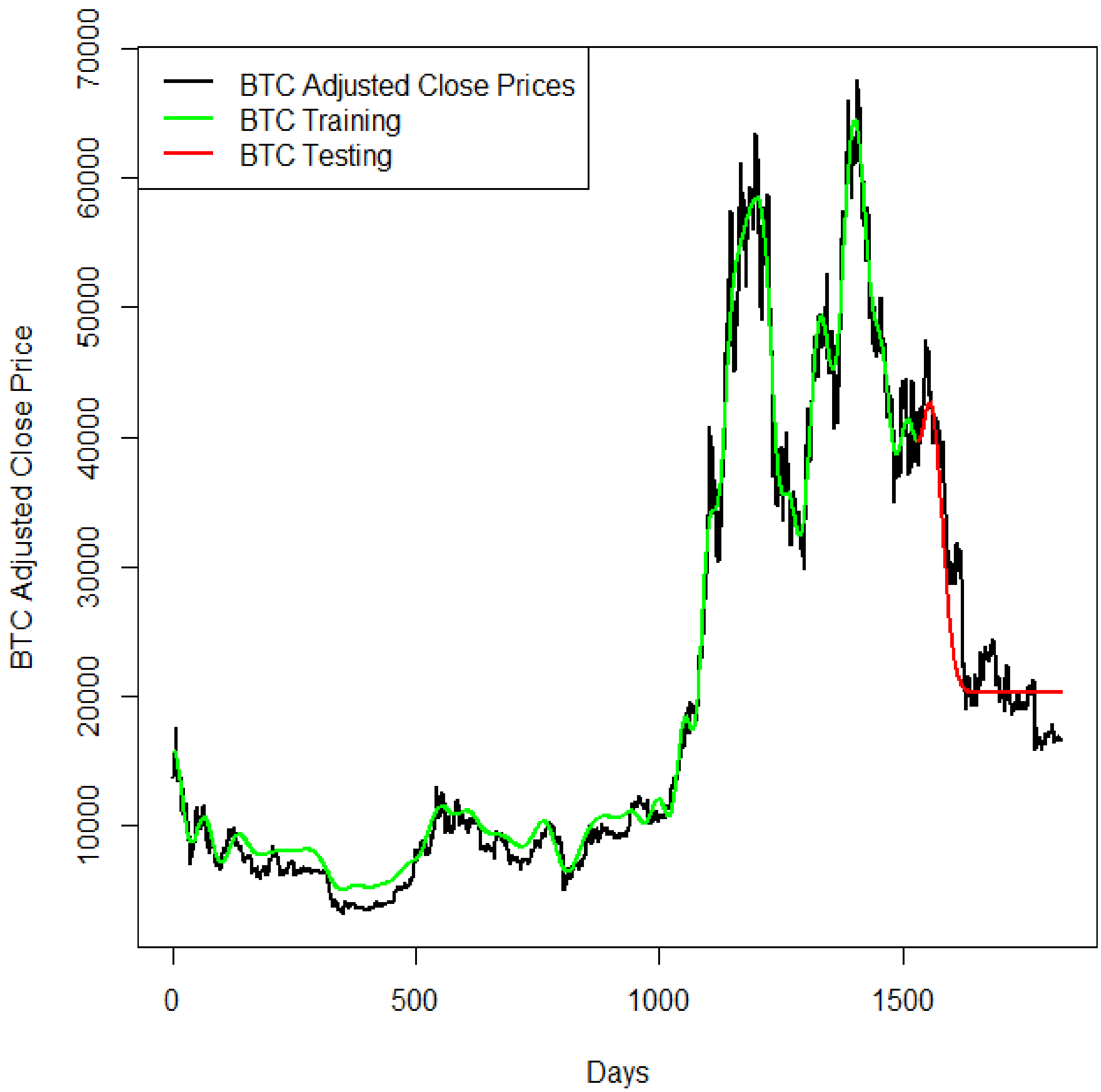
In this exercise Support Vector Regressions were run with respect to the number of observation as a predictor.

After running many rounds of cross validation and tuning for each cryptocurrency, the final architecture of SVR for each cryptocurrency is listed in Table 17.

Table 17 Architecture of SVR models:

BTC	
Method	svmRadial (Support Vector Machines with Radial Basis Function Kernel)
Sigma	110 (tested for the values from 0.1 to 150)
Cost	40 (tested for the values from 0.1 to 256)
ETH	
Method	svmRadial (Support Vector Machines with Radial Basis Function Kernel)
Sigma	120 (tested for the values from 0.1 to 150)
Cost	40 (tested for the values from 0.1 to 256)
USDT	
Method	svmRadial (Support Vector Machines with Radial Basis Function Kernel)
Sigma	110 (tested for the values from 0.1 to 300)
Cost	40 (tested for the values from 0.1 to 256)

The R2 performance metrics on Figure 24 manifests quite high predictive capacity of SVR model for BTC prices, although this model visibly had difficulties with capturing the spikes of the test data set:

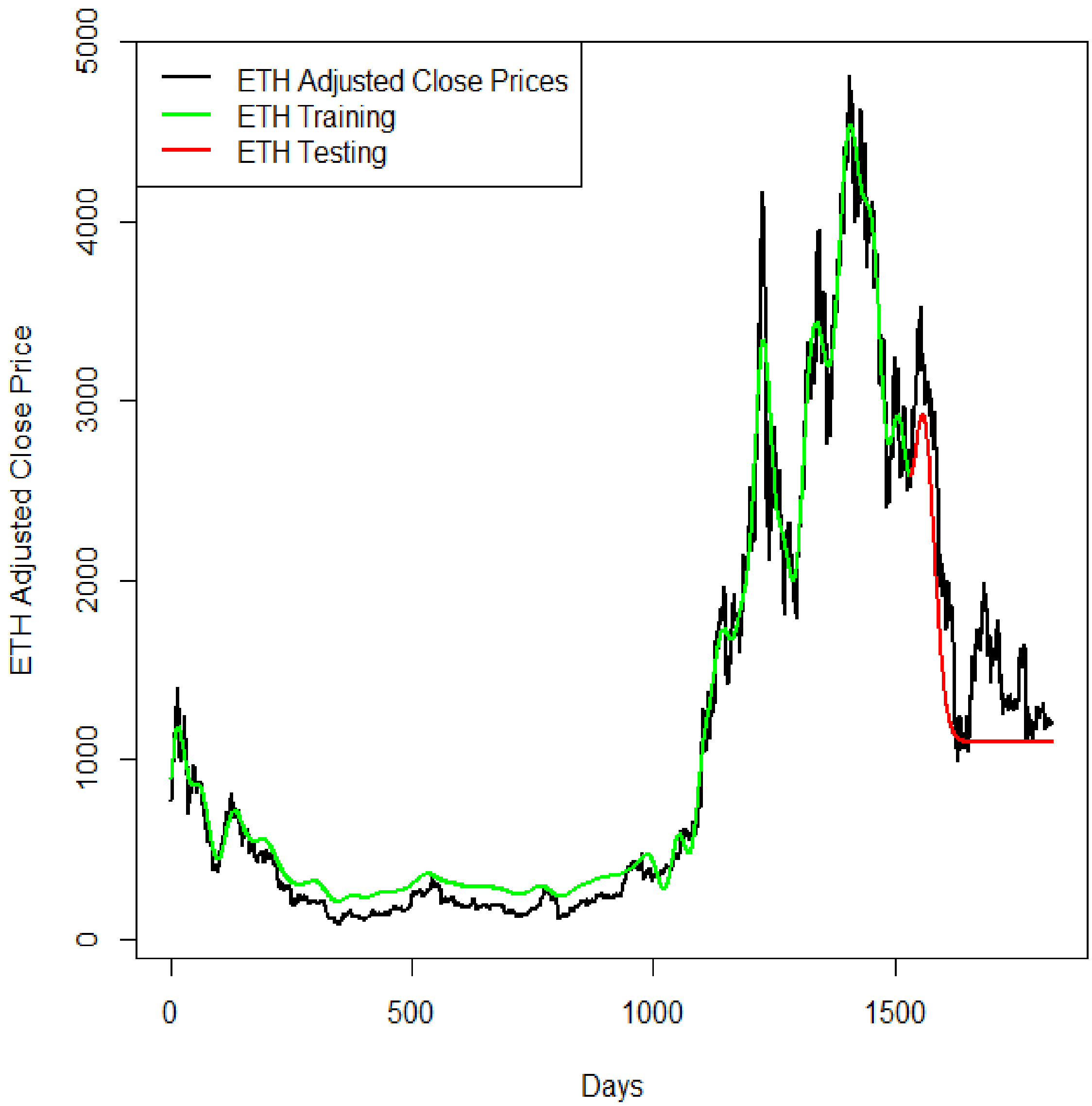


Model performance	R2 Training:	0.9904
	R2 Test:	0.863
	MASE Training:	2.472
	MASE Test:	4.4537

Figure 24 Prediction of BTC prices with SVR model

Source: author's estimates in R using the data of Yahoo finance

SVR model quite moderate accuracy of ETH price predictions on test data, as shown by performance metrics on Figure 25:

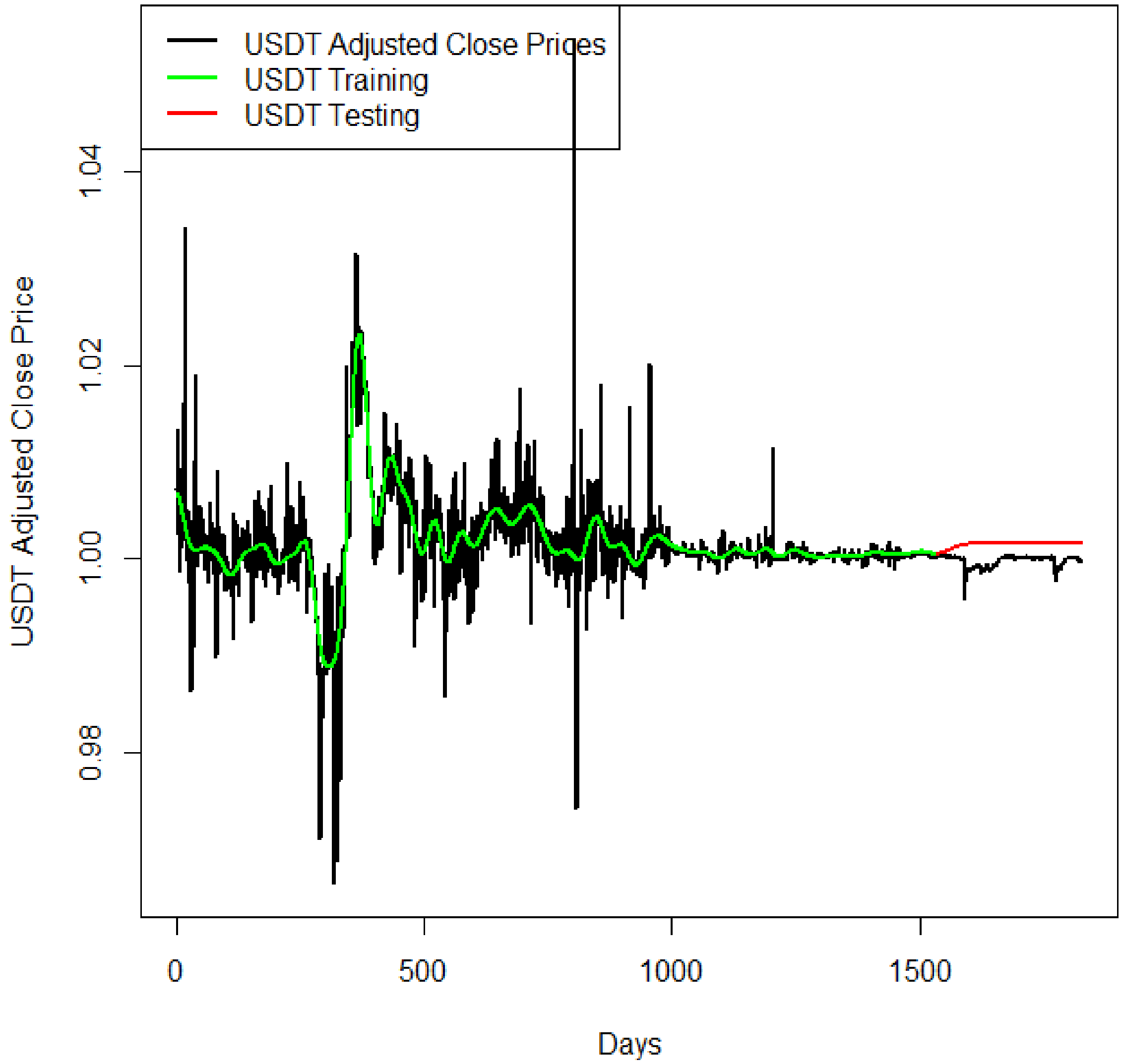


Model performance	R2	Training:	0.9876
	R2	Test:	0.6463
	MASE	Training:	2.7457
	MASE	Test:	6.1816

Figure 25 Prediction of ETH prices with SVR model

Source: author's estimates in R using the data of Yahoo finance

It was already noted in case of BTC and ETH forecasting, that the SVR model had difficulties with capturing the data with high frequency and large magnitude of oscillations. This observation becomes especially evident with the poor capacity of SVR in predicting the highly oscillating USDT prices, see Figure 26:



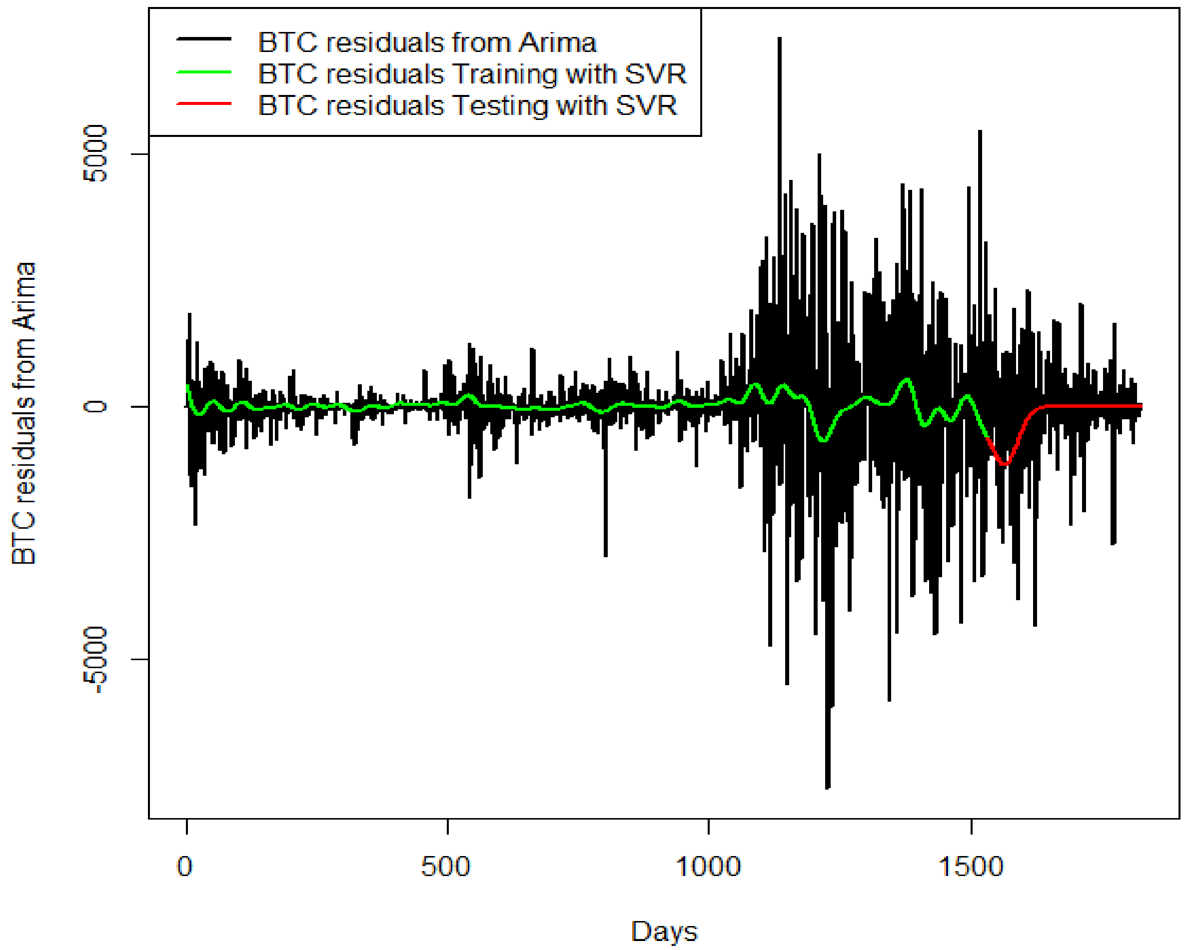
Model performance	R2 Training:	0.5647
	R2 Test:	0.1238
	MASE Training:	0.8135
	MASE Test:	14.8688

Figure 26 Prediction of USDT prices with SVR model

Source: author's estimates in R using the data of Yahoo finance

2.6 Prediction of Cryptocurrencies Prices with Hybrid ARIMA-SVR Models

As illustrated on Figure 27, SVR has very low accuracy of forecasting BTC residuals from the ARIMA model. These residuals are characterized of high frequency and high magnitude of oscillations, that are evidently are difficult to capture for a SVR model:

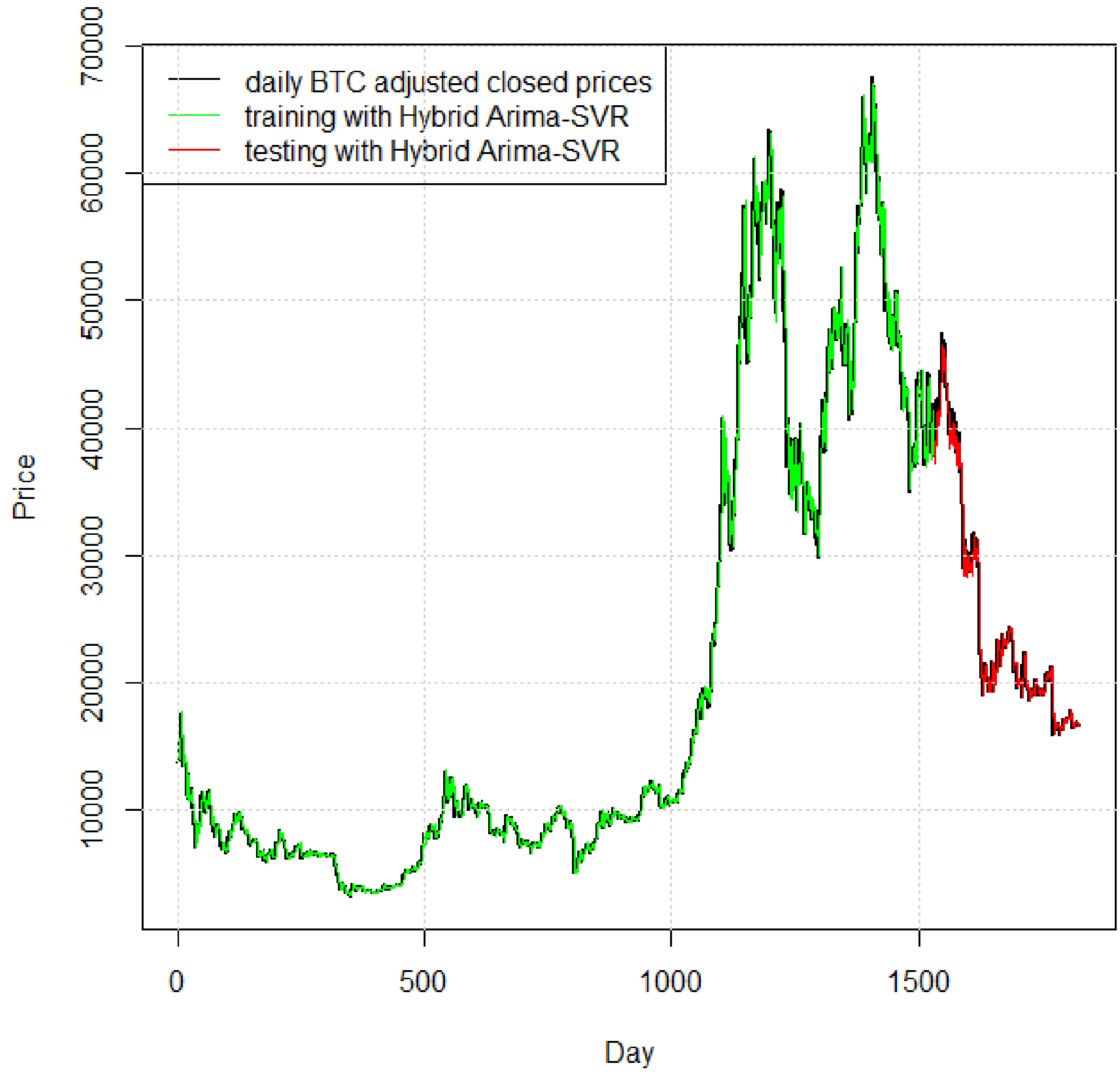


Model performance	R2 Training: 0.031 R2 Test: -0.1946 MASE Training: 0.6544 MASE Test: 0.7414

Figure 27 Prediction of BTC residuals from ARIMA with the SVR model

Source: author's estimates in R using the data of Yahoo finance

Nevertheless, the Hybrid Arima-SVR model (Figure 28) has almost unitary accuracy (in terms of R^2) of BTC price prediction:

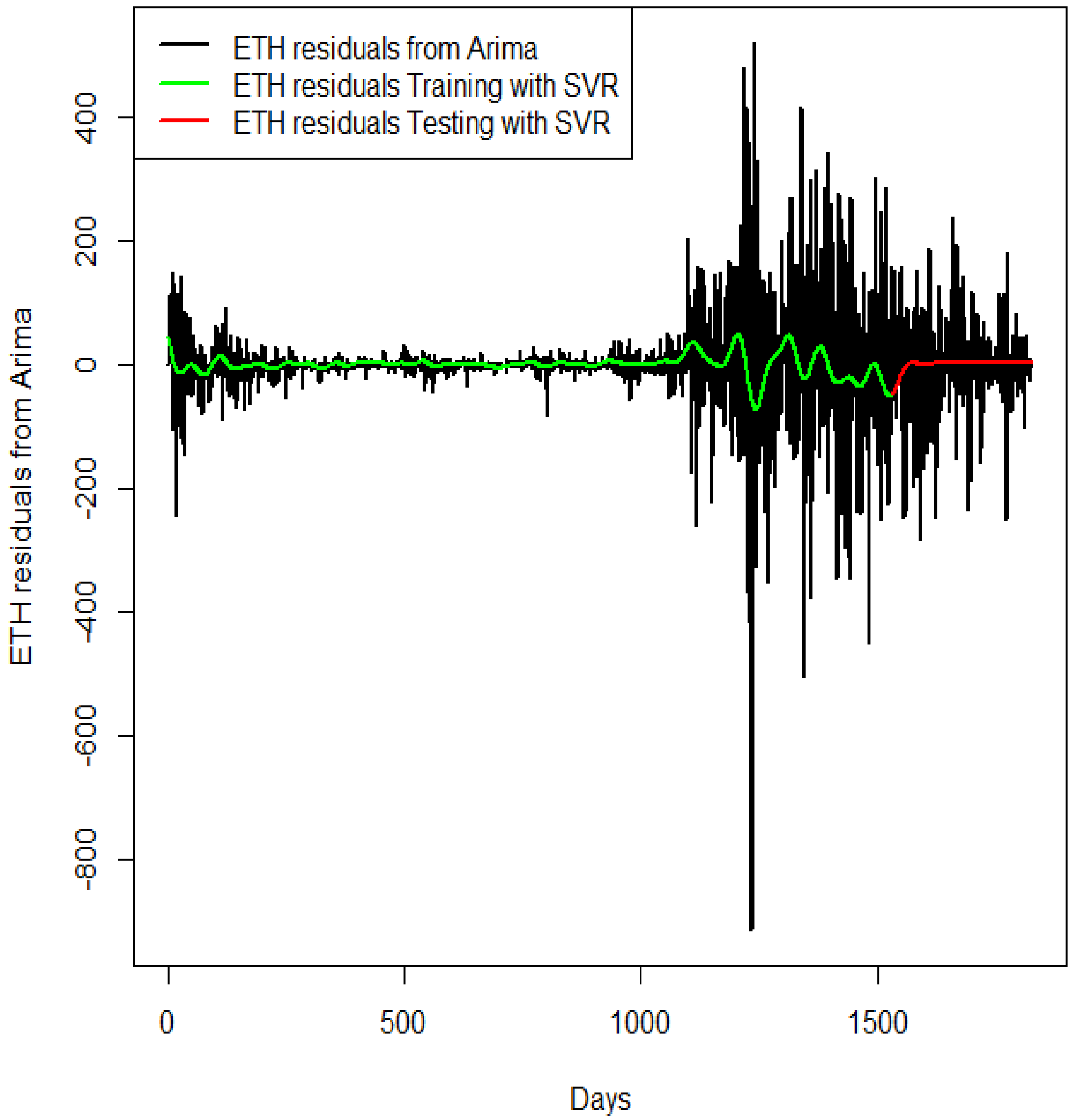


Model performance	R2 Training:	0.99679
	R2 Test:	0.98806
	MASE Training:	1.00848
	MASE Test:	1.1448

Figure 28 Prediction of BTC prices with Hybrid ARIMA-SVR model

Source: author's estimates in R using the data of Yahoo finance

As illustrated on Figure 29, in terms of R2 indicator, SVR has very low accuracy of forecasting ETH residuals from the ARIMA model:

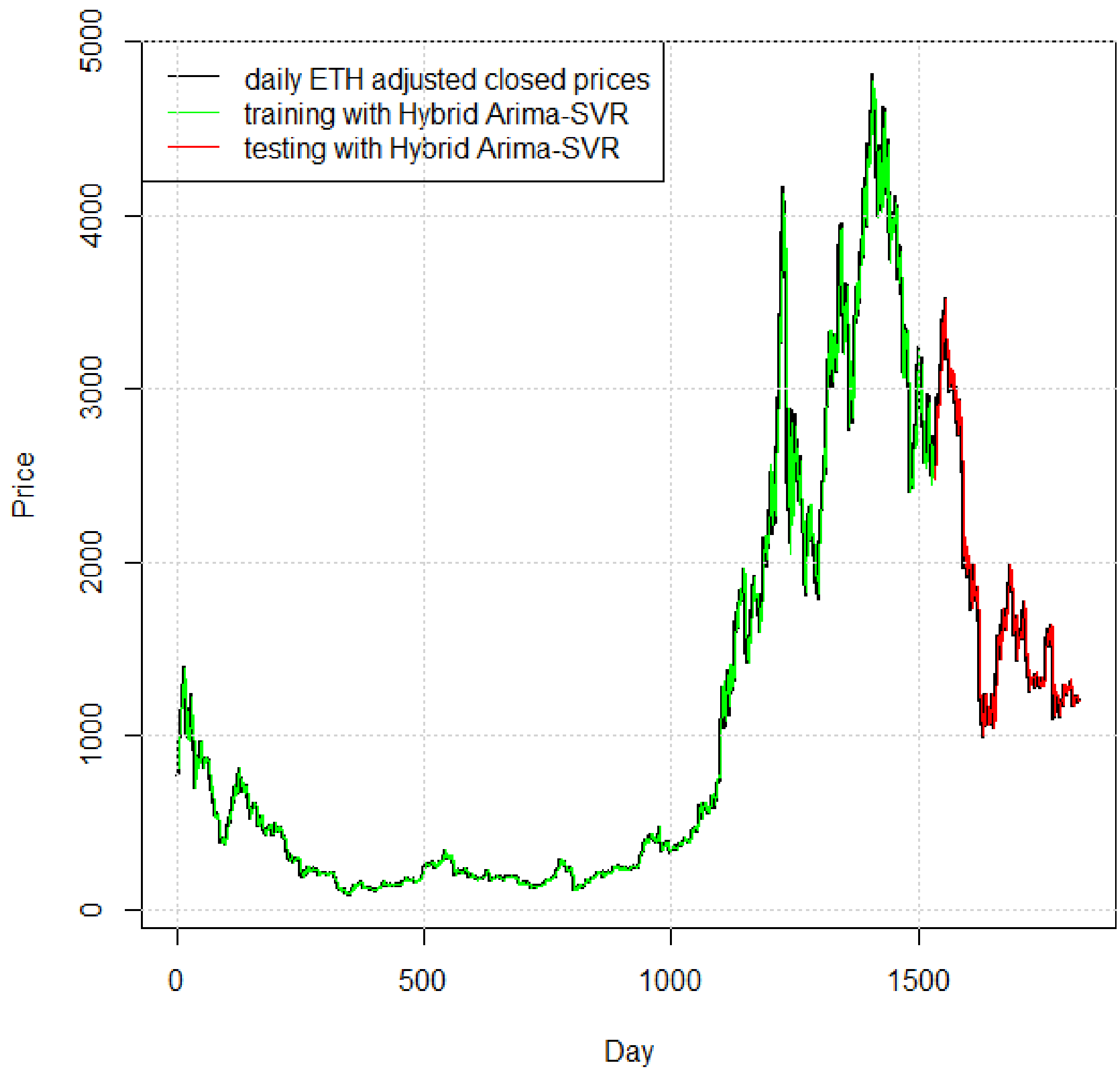


Model performance	R2 Training: 0.0301 R2 Test: 0.0280 MASE Training: 0.6761 MASE Test: 0.7244

Figure 29 Prediction of ETH residuals from ARIMA with the SVR model

Source: author's estimates in R using the data of Yahoo finance

Nonetheless, the Hybrid Arima-SVR model (Figure 30) has almost unitary accuracy of ETH price forecasting, in terms of R^2:

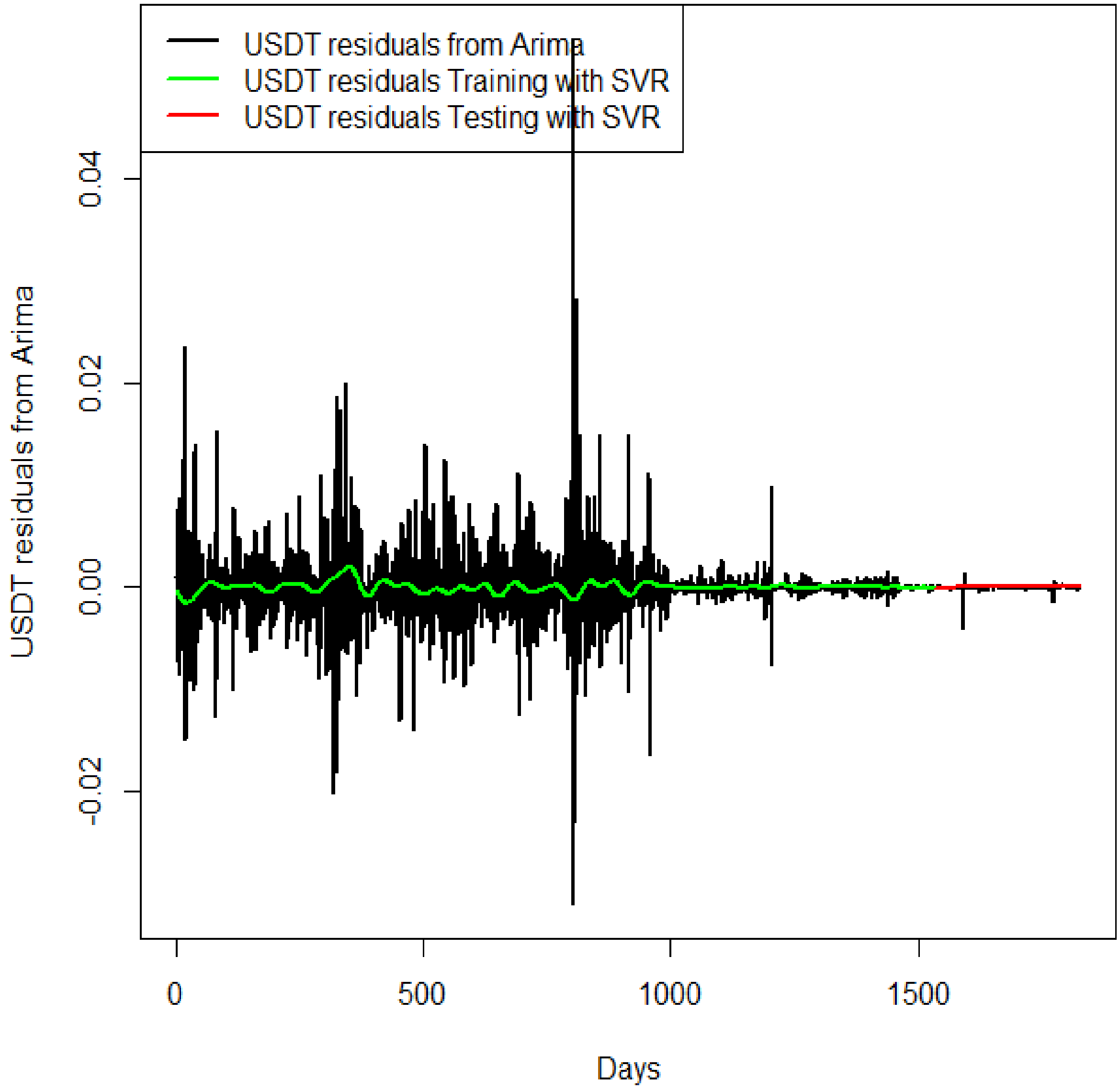


Model performance	R2 Training:	0.99594
	R2 Test:	0.98577
	MASE Training:	1.04912
	MASE Test:	1.10627

Figure 30 Prediction of ETH prices with Hybrid ARIMA-SVR model

Source: author's estimates in R using the data of Yahoo finance

When measured in terms of R^2, the SVR model has very low quality of forecasting of USDT residuals from ARIMA model, Figure 31:

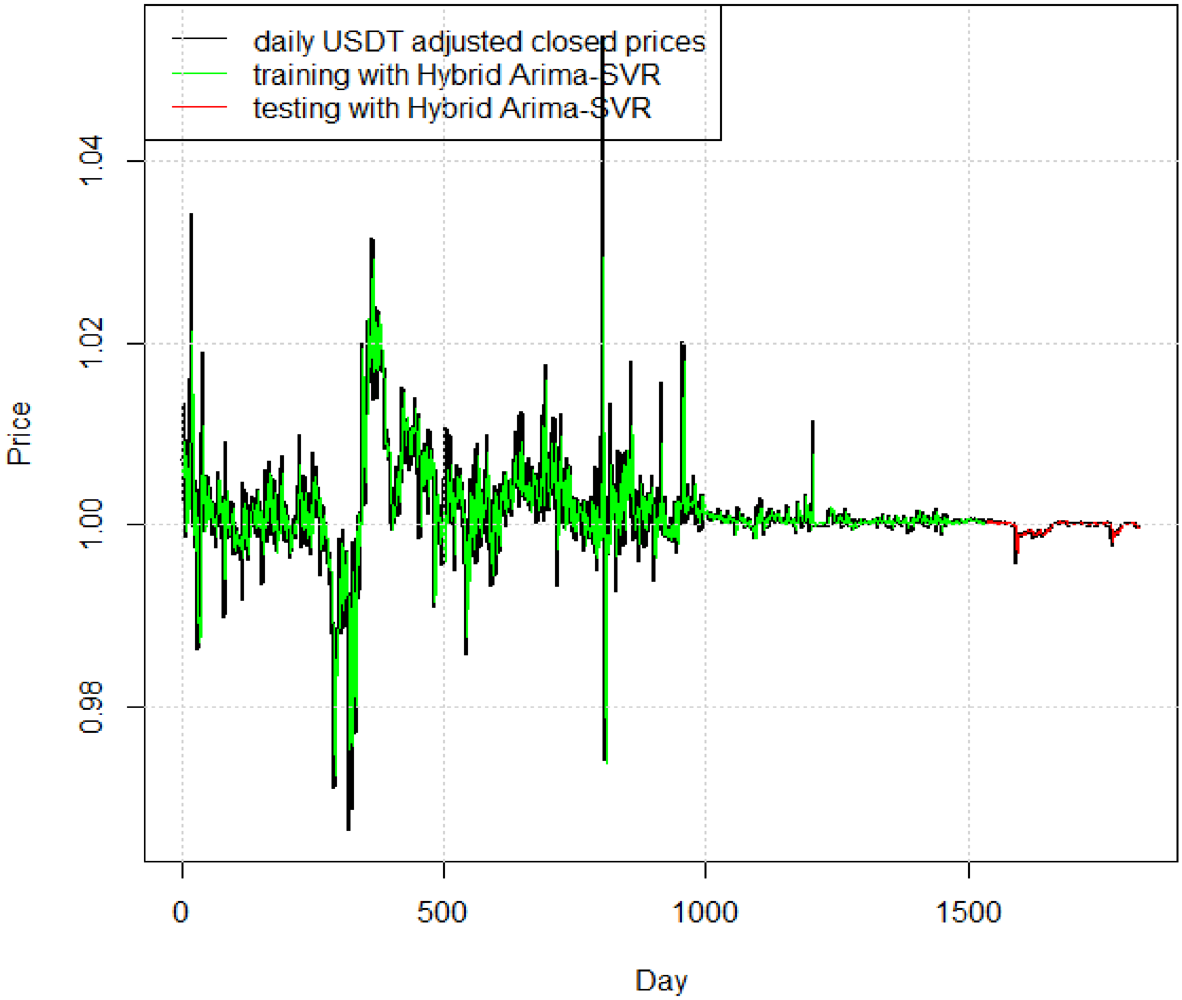


Model performance	R2 Training: 0.00239 R2 Test: 0.00007 MASE Training: 0.6745 MASE Test: 0.8958

Figure 31 Prediction of USDT residuals from ARIMA with the SVR model

Source: author's estimates in R using the data of Yahoo finance

Since both ARIMA and SVR model have difficulties with capturing the highly fluctuating data, it not surprising that the Hybrid ARIMA-SVR model has low performance in forecasting USDT prices, Figure 32:



Model performance	R2	Training:	0.41847
	R2	Test:	0.67374
	MASE	Training:	1.60582
	MASE	Test:	1.65656

Figure 32 Prediction of USDT prices with Hybrid ARIMA-SVR model

Source: author's estimates in R using the data of Yahoo finance

2.7 Prediction of Cryptocurrencies Prices with LSTM Models

LSTM models are known to tackle well the stationary and also the noisy, non-stationary and non-linear data, which is often the case with cryptocurrency prices, due to various external factors such as news events, economic indicators, market sentiment and others. Furthermore, these models can process fast and efficiently large amounts of data which is crucial for analyzing and predicting stock prices that are updated frequently, such as daily or intraday data. LSTMs are quite useful in the task of time series prediction that involve autocorrelation because of their ability to maintain state and recognize patterns over the length of the time series. LSTM models introduced by Hochreiter and Schmidhuber (1997) dominate recent literature of financial time series forecasting with deep learning (Sezer et al. 2020). The supremacy of LSTM in cryptocurrency price forecasting was demonstrated by Jiang (2019), Lahmiri & Bekiros (2019), and Ji et al.(2019), among the others.

A common LSTM unit is composed of a cell, input gate, output gate and forget gate.

The cell state transports information across the sequence chain. It works as a memory of the network. The cell state carries only the relevant information in the sequence since the information can be removed or added via the gates. The gates learn what information is pertinent to keep or forget during training. Therefore, the information of earlier stages has now impact on later stages in the sequence. LSTM has three types of gates: the input (i_t), forget (f_t) and output (o_t) gates, as shown in the equations below:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (17)$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_t) \quad (18)$$

$$\tilde{c}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \quad (19)$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (20)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (21)$$

$$h_t = o_t * \tanh(c_t) \quad (22)$$

The sigmoid function σ and the hyperbolic tangent function (tanh) are used to bound the output between 0 and 1, and between –1 and 1, respectively. W_x are the weights for the respective gate(x) neurons. $h_{t-1}$is the output of the previous LSTM block at *t-1*; x_t is the input at time t, b_x are the biases for the respective gates (x). The input gate decides what new information to store in the cell state. The forget gate decides what information must be erased from the cell state. The output gate provides activation to the final output of the LSTM block at time t. c_t is the cell state (memory) at time t; $\tilde{c}_t$ is the candidate for cell state at time t. Functional design of the LSTM cell is outlined on Figure 33:

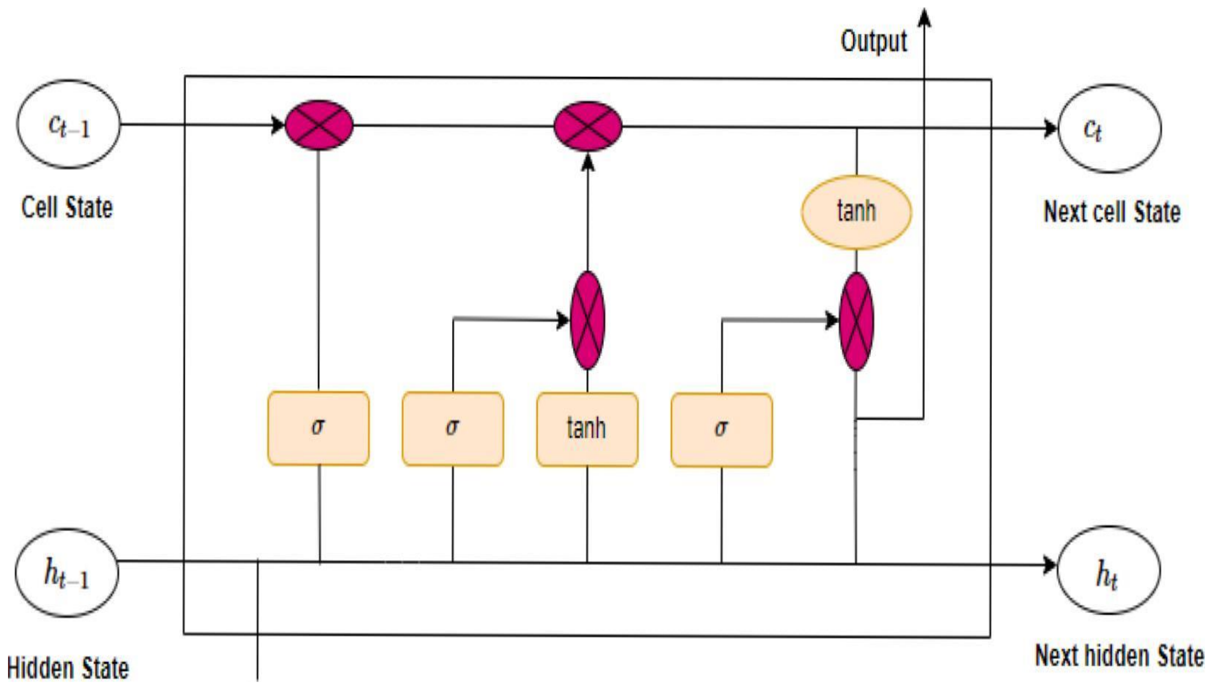


Figure 33 The Structure of a LSTM cell

Source: Seabe et al, 2023.

Various hyperparameters and their combinations can be used in LSTM models for time series forecasting. In LSTM models the values of target variable are predicted depending on the values of the lagged variables, depending on the selected number of time steps. For this study, the LSTM model was built with the following architecture, Table 18:

Table 18 Architecture of the LSTM model

Hyperparameter	Function	Value
Time steps	number of previous steps to predict the current step	8
Window size	prediction window	Time steps-1
Hidden layers	perform nonlinear transformations of the inputs entered into the network	2
Dropout layers	dropout is a regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network. This has the effect of reducing overfitting and improving model performance	2

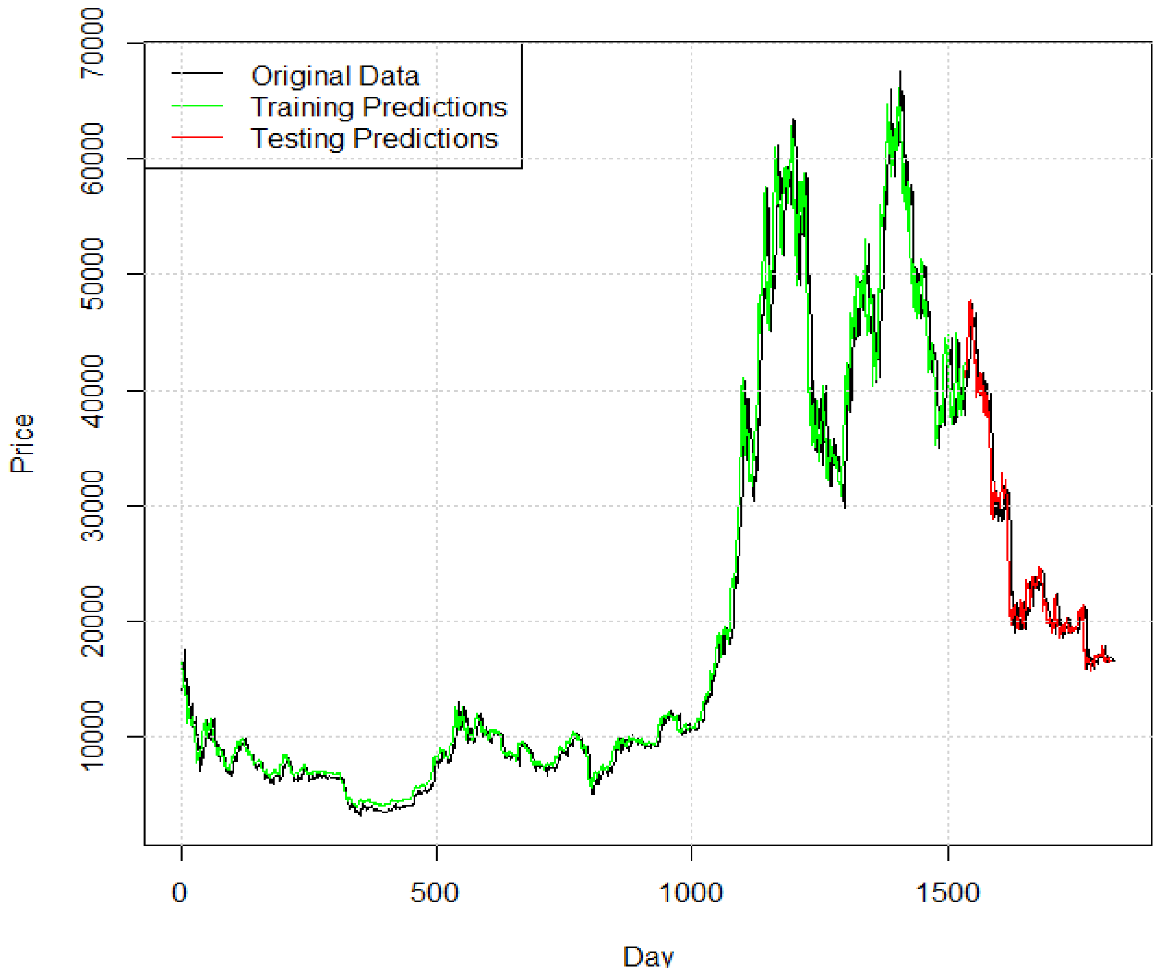
Number of units	dimension of the hidden state, determines the capability of the model to retain the memory for all the past information.	32 (BTC, USDT) 4(ETH)
Batch size	controls the number of training samples to work through before the model's internal parameters are updated	21
Epochs	number of forward and backward pass iterations. This hyperparameter controls the number of complete passes through the training dataset	200(BTC), 300(ETH) 600(USDT)
Testing set vs Training set split	split between a subset to test the trained model and a subset to train a model	19%
Optimizer	the algorithm used to optimize the weights in each level	Adam
Learning rate	the ratio of parameter update to gradient/momentum/velocity depending on the optimization algorithm used	0.01
Dropout rate	A technique to randomly drop units and their connections from the neural network during training in order to prevent overfitting	0.01(BTC) 0.0001 (ETH) 0.5 (USDT)
Activation Function	Performs a nonlinear transformation of the input data and thus enable the neurons to learn better; defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network	Sigmoid
Loss function	MSE	

In order to get the best model fit with the smallest MSE the LSTM model was tested for the following alternative settings:

- batch sizes (1,9, 21)[7];
- time steps (32,24,16,8);
- number of units (32,24,16,8);
- dropout rates (0.001, 0.1, 0.2 and 0.5);
- activation layer (Relu, TanH, Sigmoid);
- number of epochs (100, 200, 300 and 400,600,800) ;
- bidirectional layers;
- optimizer (Adam, AdaDelta, SGD, RMSProp).

[7] the batch size must be evenly divisible in both the training and testing lengths

As shown on Figure 34, the LSTM model has almost unitary accuracy of BTC price predictions:

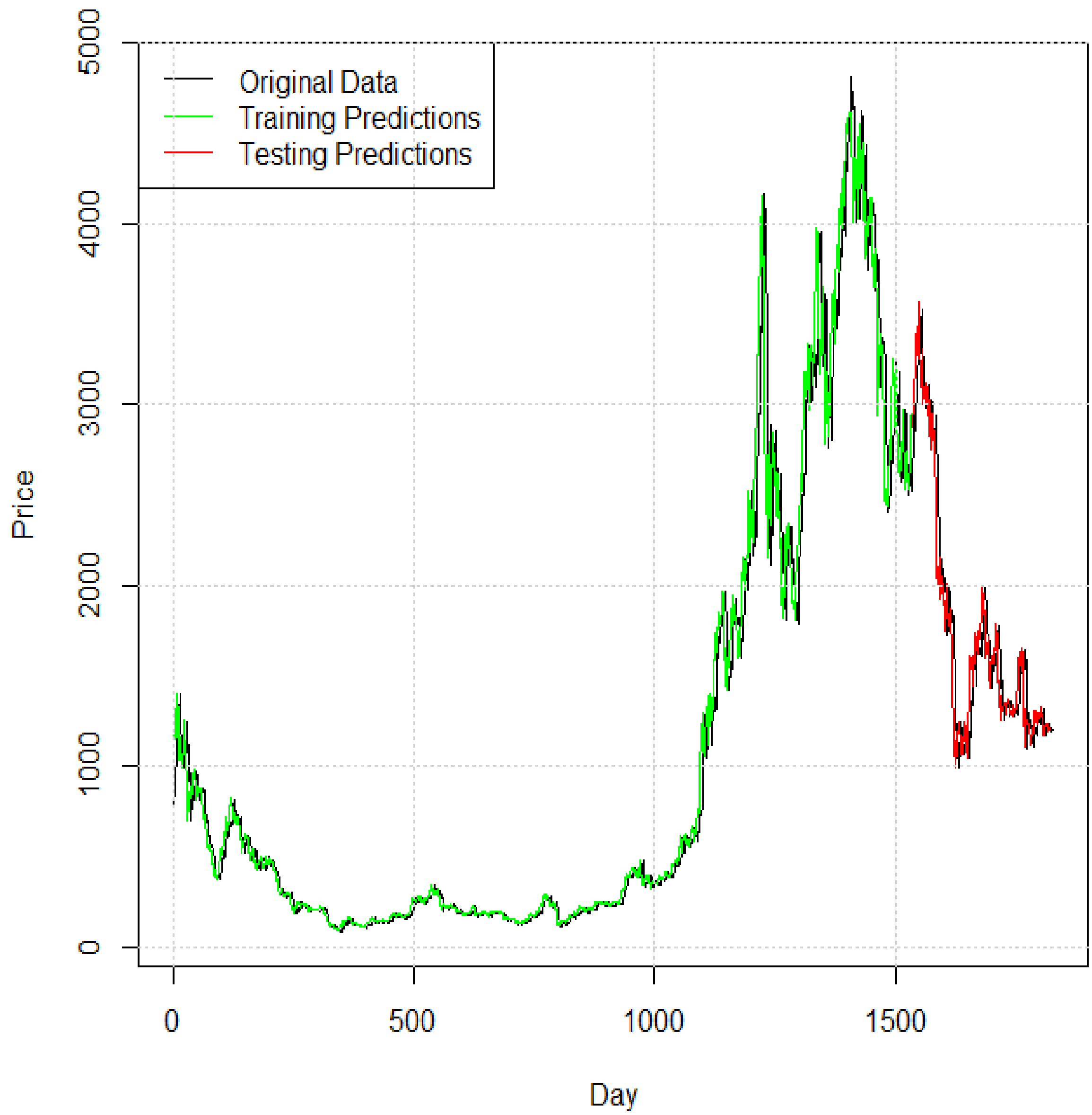


Model performance	R^2 Training:	0.997
	R^2 Test:	0.980
	MASE Training:	1.074
	MASE Test:	1.542

Figure 34 BTC price forecasting with the LSTM model

Source: author's estimates in R using the data of Yahoo finance

The LSTM model has almost unitary accuracy of ETH price forecasting, Figure 35:

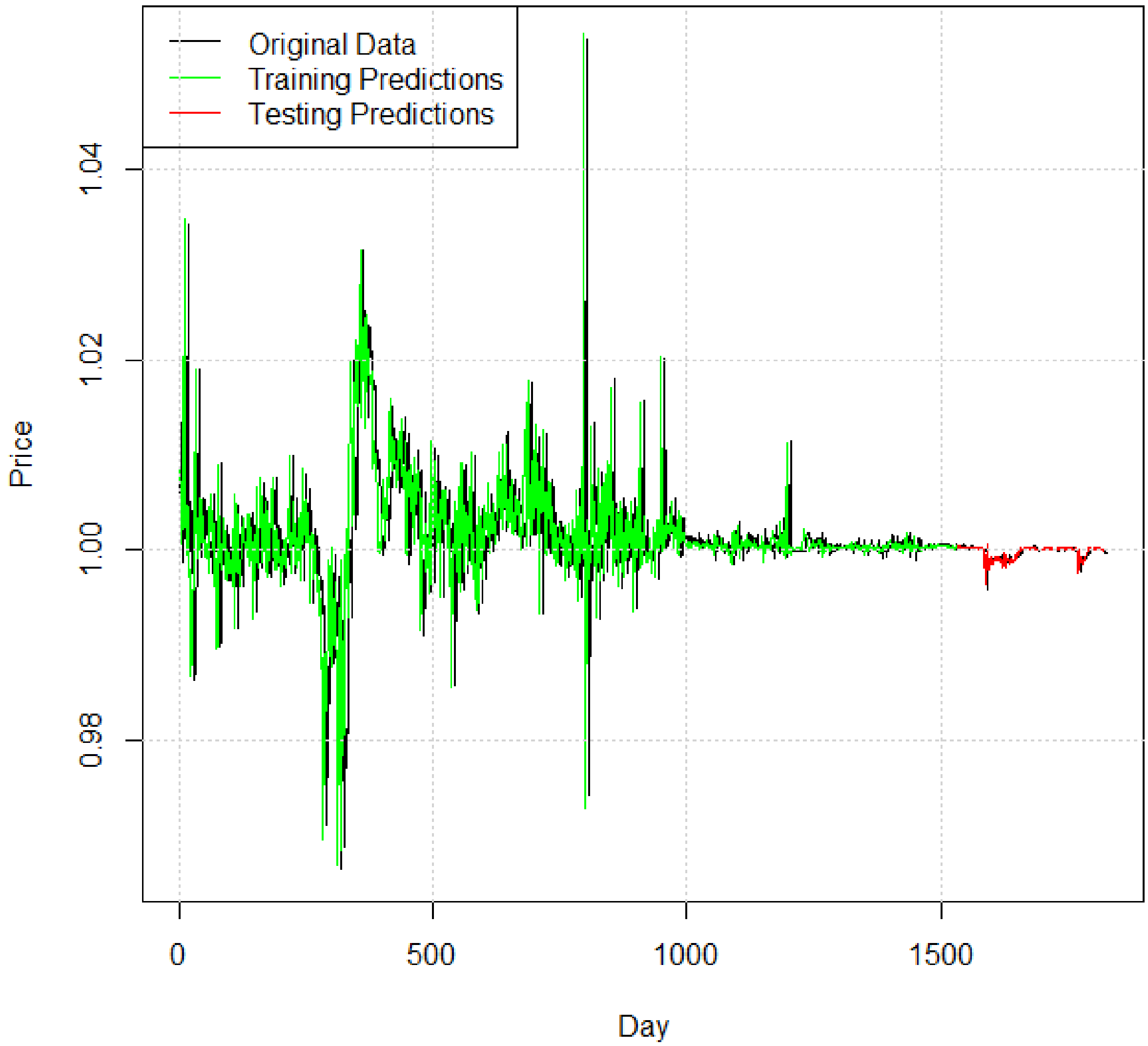


Model performance	R2	Training:	0.996
	R2	Test:	0.985
	MASE Training:		0.995
	MASE Test:		1.041

Figure 35 ETH Price Forecast with the LSTM model

Source: author's estimates in R using the data of Yahoo finance

The LSTM model have manifested a very high, almost unitary accuracy of USDT price forecasting, both on training and test data, Figure 35:



Model performance	R2	Training:	0.996
	R2	Test:	0.985
	MASE Training:		1.003
	MASE Test:		1.013

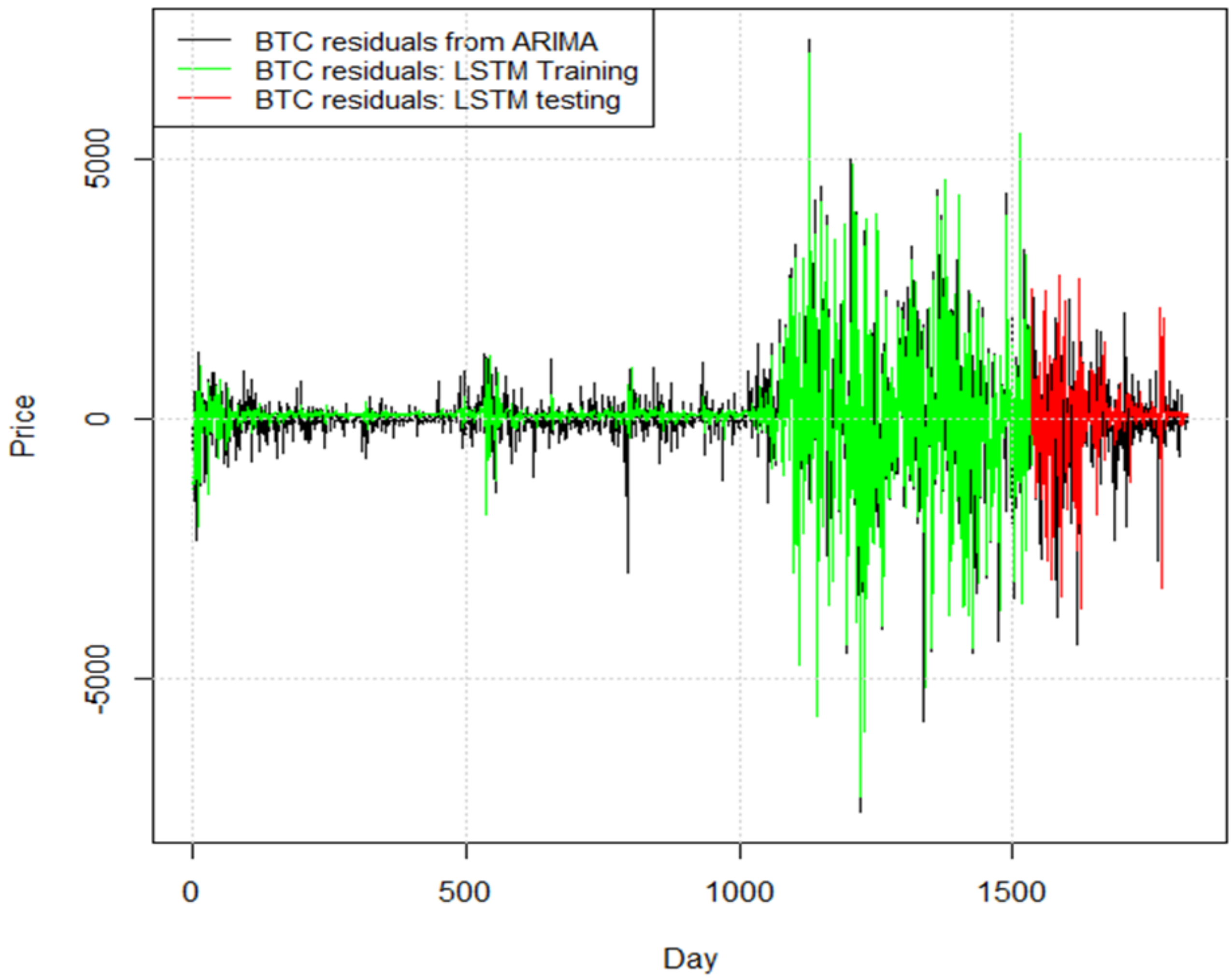
Figure 36 USDT Price Forecast with the LSTM model

Source: author's estimates in R using the data of Yahoo finance

Considering that ARIMA and SVR models had difficulties with capturing the highly oscillating USDT prices, the LSTM model clearly defended its reputation of having a good capability with non-linear data.

2.8 Prediction of Cryptocurrencies Prices with Hybrid ARIMA-LSTM Models

Results of computer simulations reveal that LSTM model has high accuracy of ARIMA residuals prediction in the training set of data, and quite low accuracy of predictions with test data reflected in R^2 metrics, although the MASE metrics confirms its high predictive capacity on both sets of data (Figure 37):

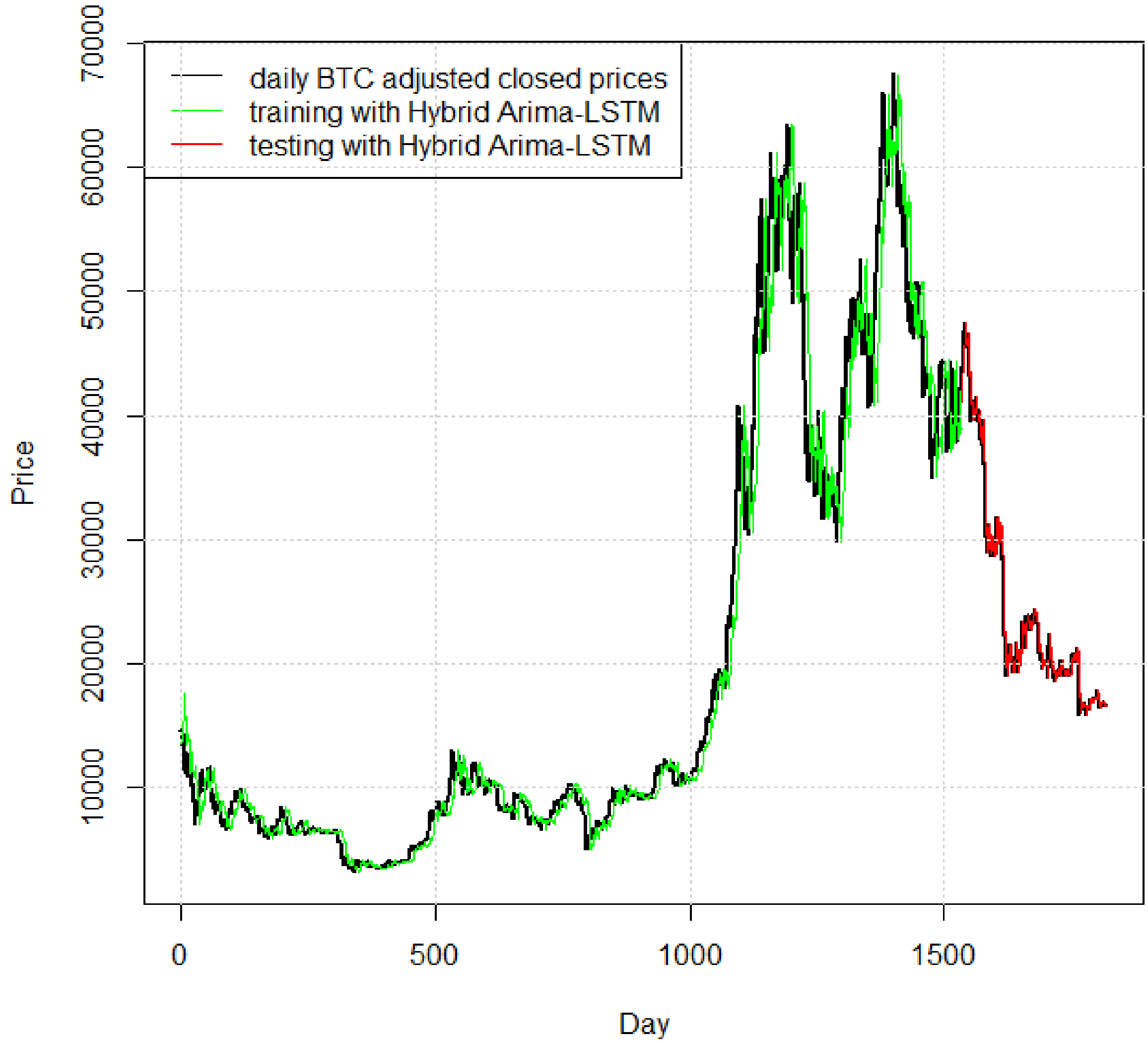


Model performance	R2 Training:	0.93853
	R2 Test:	0.01278
	MASE Training:	0.2154
	MASE Test:	0.91067

Figure 37 Prediction of BTC residuals from ARIMA with the LSTM model

Source: author's estimates in R using the data of Yahoo finance

The Hybrid Arima-LSTM model manifests almost unitary accuracy of predictions on the training set of data (R^2 measure), and a bit lower 98% accuracy of predictions on test data (R2 measure), Figure 38:

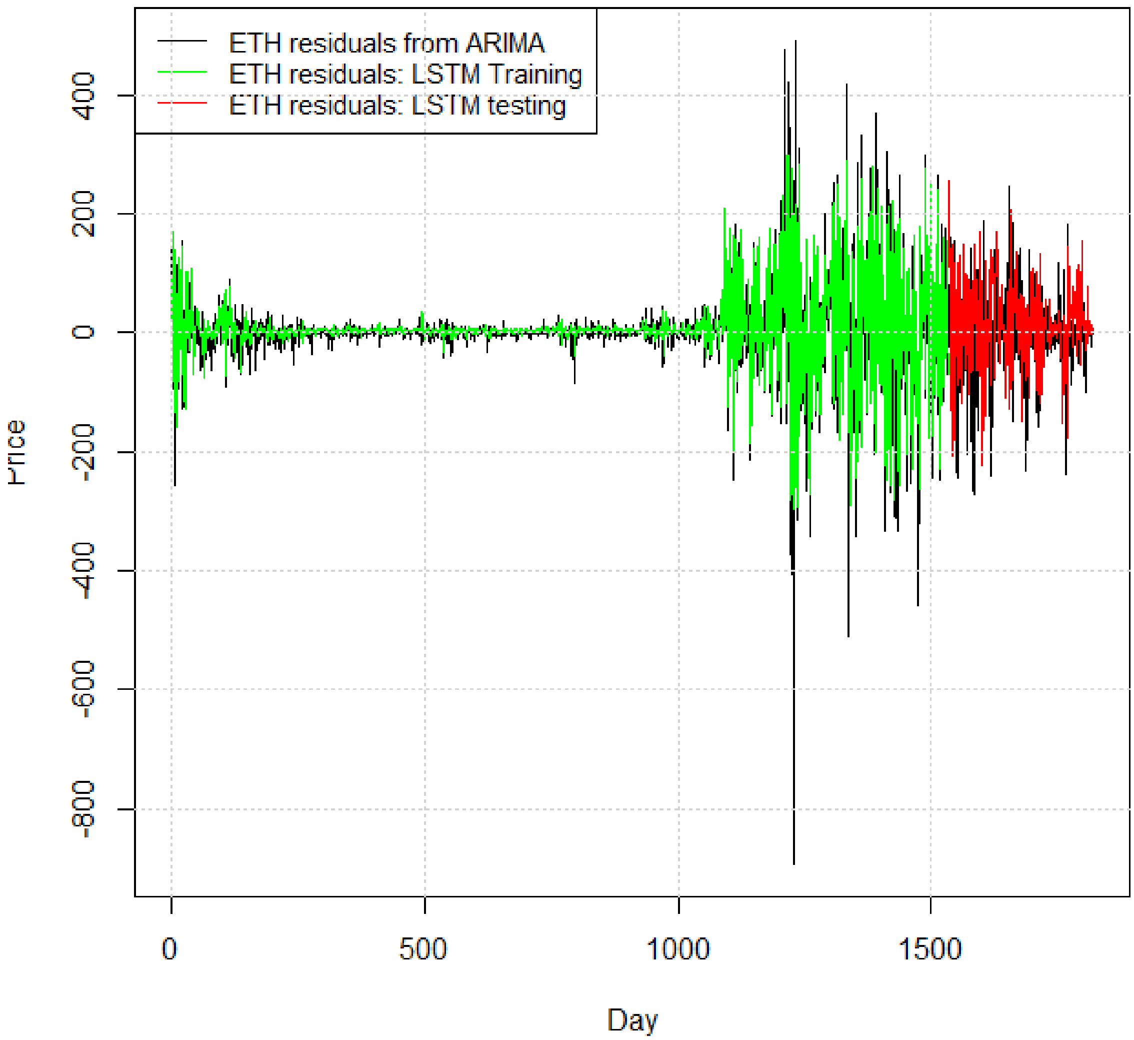


Model performance	R2	Training:	0.99671
	R2	Test:	0.97916
	MASE Training:		1.02906
	MASE Test:		0.83103

Figure 38 Prediction of BTC Adjusted Close Prices with the Hybrid ARIMA-LSTM model

Source: author's estimates in R using the data of Yahoo finance

The LSTM model has quite high accuracy of predicting the ETH residuals with train data and very low accuracy with test data (R2 measure), although the MASE metrics confirms its high predictive capacity on both sets of data , Figure 39:

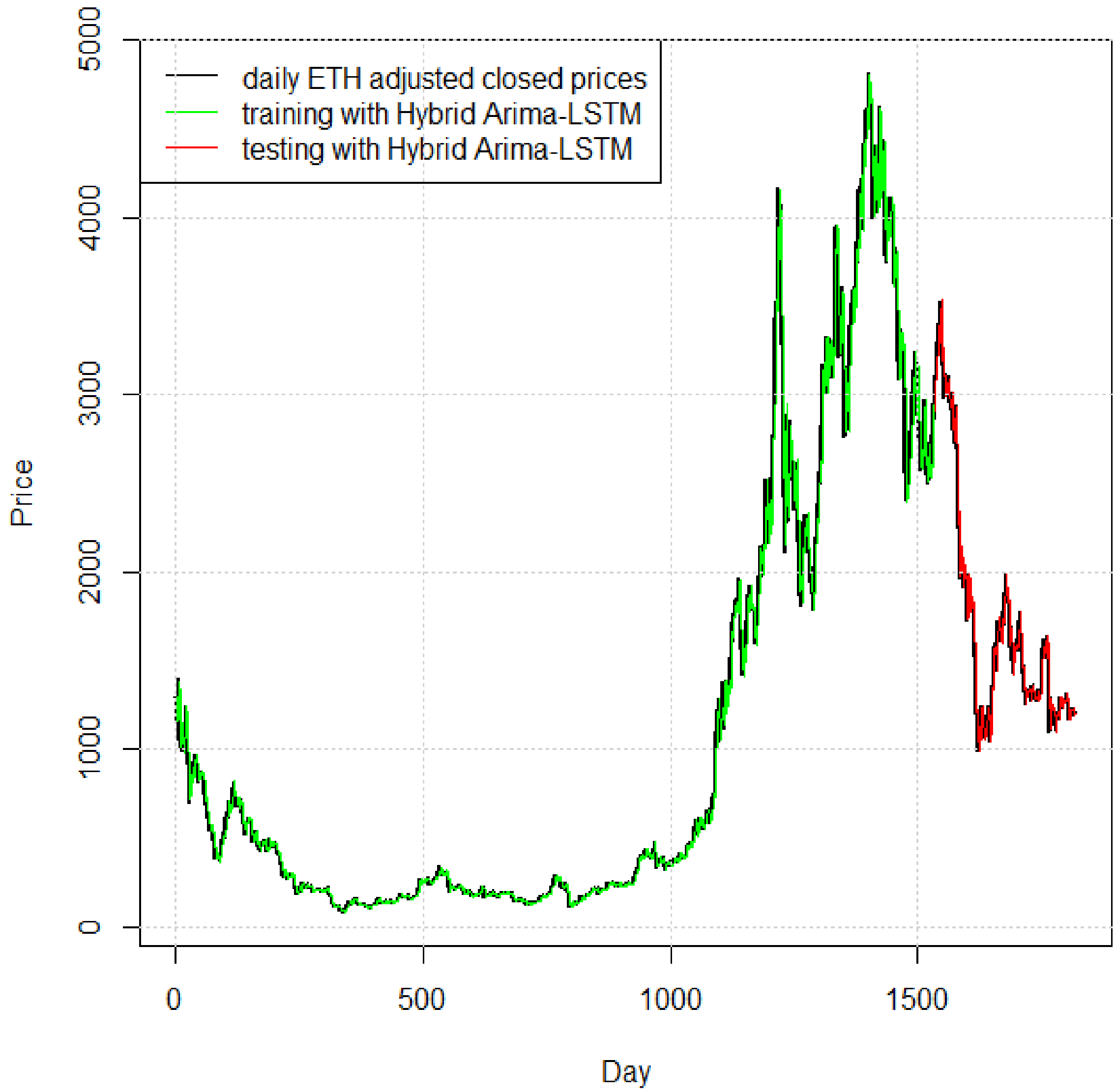


Model performance	R2 Training: 0.94788
	R2 Test: 0.01971
	MASE Training: 0.30379
	MASE Test: 0.92017

Figure 39 Prediction of ETH Residuals from ARIMA with the LSTM model

Source: author's estimates in R using the data of Yahoo finance

The Hybrid Arima-LSTM model has almost unitary accuracy of ETH price predictions with train data and test data, Figure 40:

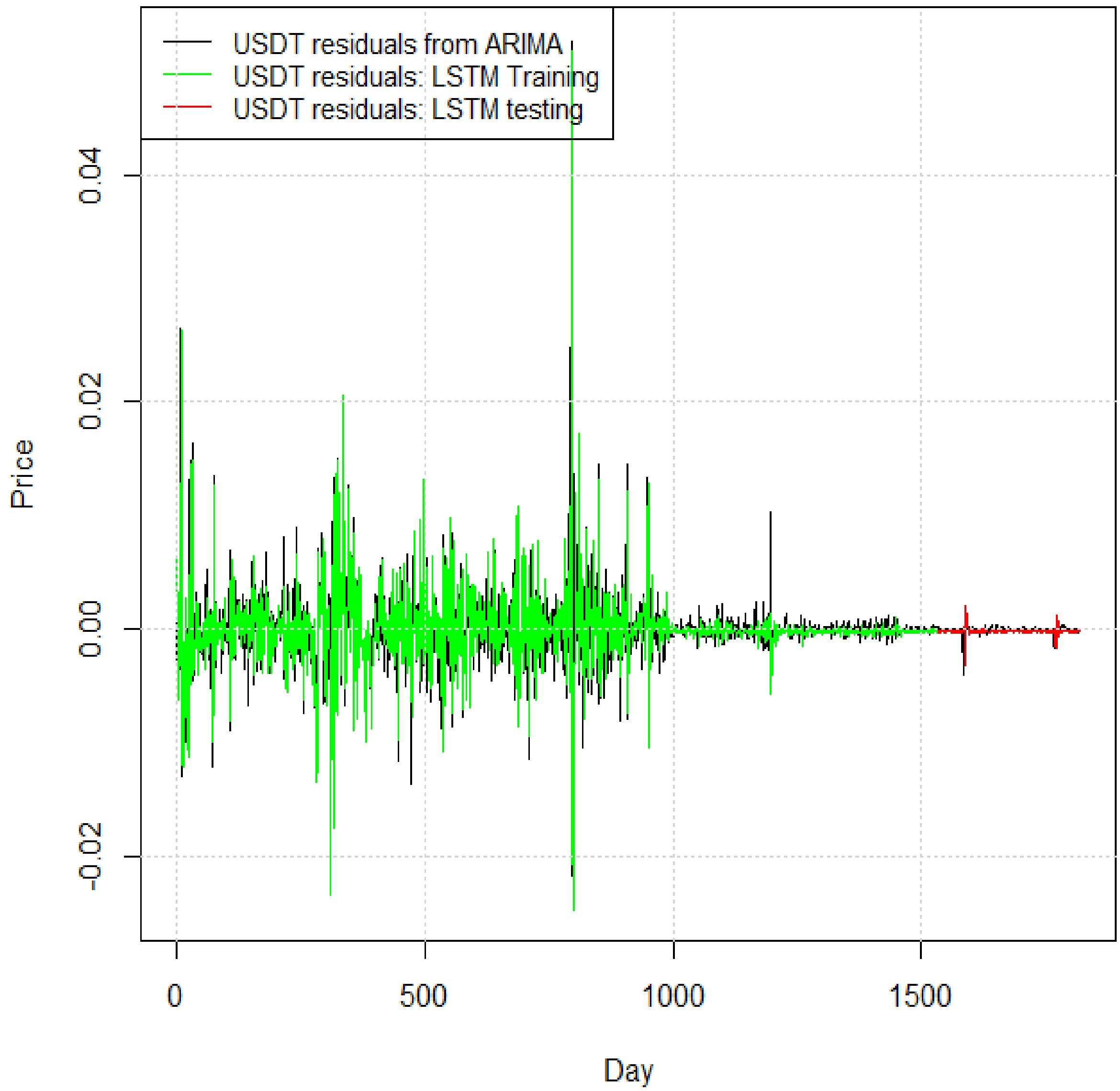


Model performance	R2	Training:	0.99592
	R2	Test:	0.98624
	MASE	Training:	1.04562
	MASE	Test:	1.03724

Figure 40 Prediction of ETH Adjusted Close Prices with the Hybrid ARIMA-LSTM model

Source: author's estimates in R using the data of Yahoo finance

The LSTM model has high accuracy of USDT residuals forecasting on train data and very low accuracy on test data, reflected in both R^2 and MASE metrics, Figure 41:

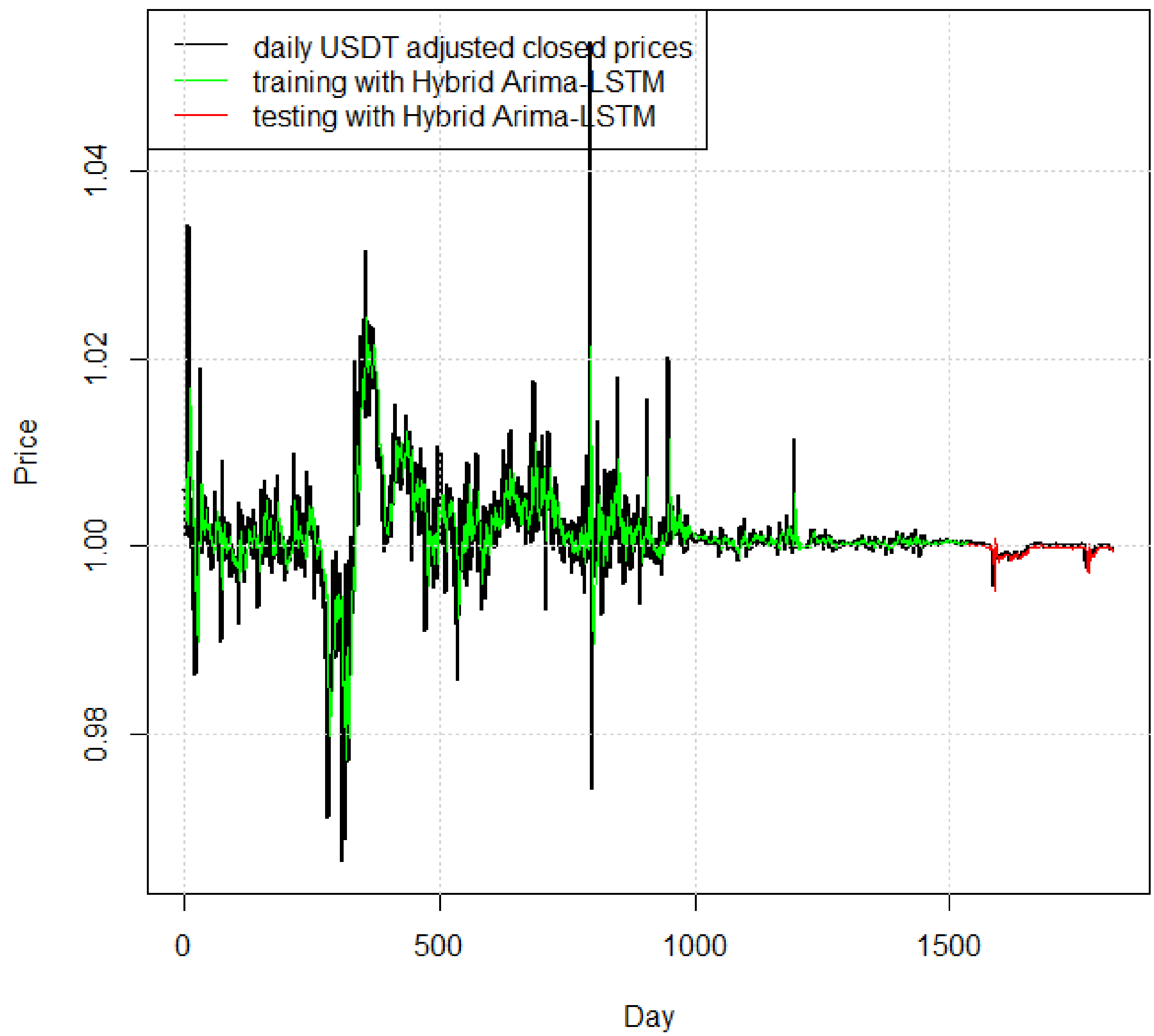


Model performance	R2 Training:	0.91631
	R2 Test:	0.00203
	MASE Training:	0.26812
	MASE Test:	1.87017

Figure 41 Prediction of USDT residuals from ARIMA with the LSTM model

Source: author's estimates in R using the data of Yahoo finance

The Hybrid Arima-LSTM model has low accuracy of USDT price prediction in terms of R^2 and MASE measures, Figure 42:



Model performance	R2	Training:	0.37409
	R2	Test:	0.36898
	MASE	Training:	2.16038
	MASE	Test:	1.50476

Figure 42 Prediction of USDT Adjusted Close Prices with the Hybrid ARIMA-LSTM model

Source: author's estimates in R using the data of Yahoo finance

Summing up this chapter, the performance of the alternative models that were tested in this study can be summarized in Table 19:

Table 19 Comparison of the alternative models' performance in BTC, ETH and USDT price prediction

Model	Data set	R2	MASE	R2	MASE	R2	MASE
		BTC		ETH		USDT	
ARIMA	Training	**0.997**	**0.998**	**0.996**	**0.987**	**0.556**	**0.873**
	Test	**0.991**	**1.006**	**0.987**	**1.004**	**0.716**	**1.117**
Random Forest	Training	1.000	0.201	1.000	0.191	0.990	0.108
	Test	0.926	3.292	0.990	0.955	0.675	4.493
Hybrid ARIMA-Random Forest	Training	1.000	0.094	1.000	0.096	0.989	0.093
	Test	0.999	0.233	0.999	0.222	0.692	1.991
SVR	Training	0.990	2.472	0.988	2.746	0.565	0.814
	Test	0.863	4.454	0.646	6.182	0.124	14.869
Hybrid ARIMA-SVR	Training	0.997	1.008	0.996	1.049	0.418	1.606
	Test	0.988	1.145	0.986	1.106	0.674	1.657
LSTM	Training	0.997	1.074	0.996	0.995	0.996	1.003
	Test	0.980	1.542	0.985	1.041	0.985	1.013
Hybrid ARIMA-LSTM	Training	0.997	1.029	0.996	1.046	0.374	2.160
	Test	0.979	0.831	0.986	1.037	0.369	1.505

Comparison of models' performance in cryptocurrency forecasting revealed that the forecasting accuracy differs much across the tested models and across the selected cryptocurrencies, see Table 19.

The least predictable cryptocurrency is USDT. The forecasting difficulties are likely to be caused by USDT price trajectory that is characterized by a high frequency of oscillations of different amplitude around the mean. The best predictive capacity of USDT prices had LSTM model with a 99% accuracy of predictions on test data. Second best almost 70% accuracy on test data of were provided by Random Forest and Hybrid Arima-Random Forest models.

The most predictable cryptocurrency is Ether. ARIMA, Random Forest, Hybrid Arima-Random Forest, Hybrid Arima-SVR, LSTM and hybrid ARIMA-LSTM models ensured almost 100% accuracy of predictions on training and test data.

As for Bitcoin, ARIMA, Random Forest, Hybrid Arima-Random Forest, Hybrid Arima-SVR, LSTM and Hybrid Arima-LSTM have accuracy of predictions exceeding 97%.

A comparison among the models permits to conclude that the LSTM model has the highest predictive capacity for all three cryptocurrencies, followed by the Random Forest and Hybrid Arima-Random Forest models.

ARIMA model has good predictive capacity on Bitcoin and Ether, however it did not capture well the oscillating trajectory of Tether prices, which is quite expected result for a linear model.

SVR and LSTM models had difficulties with predicting the residuals from ARIMA model, that are characterized by a high frequency of oscillations around the mean with rapidly changing amplitude along the model horizon. For this reason, the Hybrid ARIMA-LSTM and the Hybrid Arima-SVR framework did not improve much the quality of ARIMA predictions, especially on test data.

Random Forest were the only model with a good predictive capacity of ARIMA residuals, especially for ETH and BTC. For these cryptocurrencies the Hybrid ARIMA-Random Forest framework clearly outperformed both ARIMA and the Random Forest models.

For BTC and ETH the Hybrid Arima-LSTM model had better fit than ARIMA on training data, however performed slightly worse than ARIMA on test data.

Conclusions

This study contributes to the literature on financial time series forecasting by investigating the interdependencies and interactions among Bitcoin, Ether and Tether, and by identifying the most adequate method for forecasting prices of these cryptocurrencies.

Statistical tests conducted in R demonstrated that these three cryptocurrencies are highly non-linear, non-stationary with high degrees of autocorrelation, long memory, structural breaks and the first order of integration. Seasonal effects were not detected in the daily data of selected cryptocurrencies. Bitcoin, Ether and Tether are cointegrated. Volatility of these cryptocurrencies exhibits clustering and has varied considerably over time.

Being a stable coin, price trajectory of USDT is characterized by a high frequency of oscillations of different amplitude around the mean. Although BTC and ETH were created in different time and in different manner, their prices follow quite similar predominantly rising trend of different magnitude.

In accordance with the Granger Causality test, past values of Ether log returns have predictive capacity for BTC and USDT, whereas that past values of USDT log returns can be used in prediction of Ether and Bitcoin.

The forecast error variance decomposition applied to the VAR model gave evidence that the stationarized Bitcoin prices are predominantly influenced by their past values and are largely insensitive to the variations in Ether or Tether. In contrast, through the whole horizon of forecasting, about 40% of the variation in stationarized ETH prices came from the shocks to ETH itself and the remaining 60% originated from

BTC perturbations. In a similar fashion to BTC, the stationarized USDT prices are affected only by the shocks to itself and are irresponsive to the shocks to BTC or ETH. All the shock impacts described above were stable during the short run forecast period.

The long-run dynamic volatility spillovers investigated with the multivariate GARCH-DCC models revealed that high historical correlations between Bitcoin and Ether make them good candidates for pairs trading; in both pairs USDT can be used as a base currency that indicates the value of the crypto compared to USD. The fact that USDT reacted differently to ETH and BTC to external shocks, like market crash in March 2020 proves its capacity for diversifying a portfolio consisting of BTC or ETH cryptocurrencies.

Johansen Cointegration test conducted in this study permitted to devise a stationary dynamic portfolio consisting of a certain number of Bitcoin, Ether and Tether shares suggested by a cointegrating vector. Investors can profit from purchasing this stationary portfolio when price is low and get a profit when its price returns to the mean or crosses above a certain level. Similarly, an investor can short sell the spread when its price is high and get a profit when price reverts to the mean or crosses below certain threshold.

In case of trading the individual cryptocurrencies, pairs trading or portfolio trading profits depend on the correctness of forecasted price moves. Considering high volatility of cryptocurrency markets, even small increases in model's precision can generate large profits to investors.

In order to determine the most accurate price forecasting method for each cryptocurrency the performance of the traditional linear Arima model was compared with machine learning models such as Random Forest, SVR and LSTM. To operationalize the non-linear residuals from ARIMA with machine learning models that have high predictive capacity on nonlinear data, Hybrid ARIMA-LSTM, Hybrid ARIMA-SVR and Hybrid ARIMA-Random Forest models were proposed and implemented.

The results of computer simulations revealed that the forecasting accuracy differs much across the tested models and across the selected cryptocurrencies.

The least predictable cryptocurrency is USDT. The forecasting difficulties are likely to be caused by USDT price trajectory that is characterized by a high frequency of oscillations of different amplitude around the mean. The best predictive capacity of USDT prices had LSTM model with a 99% accuracy of predictions on test data. Second best almost 70% accuracy on test data of were provided by Random Forest and Hybrid Arima-Random Forest models.

The most predictable cryptocurrency is Ether. ARIMA, Random Forest, Hybrid Arima-Random Forest, Hybrid Arima-SVR, LSTM and hybrid ARIMA-LSTM models ensured at least 98% accuracy of predictions on training and test data. The Hybrid Arima-Random Forest models had the highest almost unitary accuracy.

As for Bitcoin, ARIMA, Random Forest, Hybrid Arima-Random Forest, Hybrid Arima-SVR, LSTM and Hybrid Arima-LSTM have accuracy of predictions exceeding 97%. Overall, BTC was the best predicted with the Hybrid Arima-Random Forest model, that ensured almost unitary accuracy.

A comparison among the models permits to conclude that the LSTM model has the highest predictive capacity for all three cryptocurrencies, followed by the Random Forest and Hybrid Arima-Random Forest models.

ARIMA model has good predictive capacity on Bitcoin and Ether, however it did not capture well the oscillating trajectory of Tether prices.

SVR and LSTM models had difficulties with predicting the residuals from ARIMA model, that are characterized by a high frequency of oscillations around the mean with rapidly changing amplitude along the model horizon. For this reason, the Hybrid ARIMA-LSTM and the Hybrid Arima-SVR framework did not improve much quality of ARIMA predictions, especially on test data.

Random Forest were the only model with a good predictive capacity of ARIMA residuals, especially for ETH and BTC. For this reason the Hybrid ARIMA-Random Forest framework clearly outperformed ARIMA.

For BTC and ETH the Hybrid Arima-LSTM model had better fit than ARIMA on training data, however performed slightly worse than ARIMA on test data.

The result of this study is especially relevant for finance practitioners and cryptocurrency traders.

First, the causality and spillover relationships between cryptocurrency pairs that were identified in this study allow to determine the trading pairs and to decide on which positions to take in pairs trading.

Second, a dynamic stationary portfolio of the three cryptocurrencies was constructed, where a number of Bitcoin, Ether and Tether shares is suggested by a Johansen cointegrating vector.

Third, for each cryptocurrency were found the most accurate forecasting method, which can be used in individual, pairs or portfolio trading. Novel hybridization methods were tested on forecasting cryptocurrencies prices.

This research can be extended by testing trading strategies that associate profits or losses with models' predictive capacity. This exercise would provide additional real-word validation for the constructed models.

In addition, this study can be tested on more cryptocurrencies and with more machine learning tools for cryptocurrency price forecasting.

References

Alahmari, S. A. (2019). Using Machine Learning ARIMA to Predict the Price of Cryptocurrencies. ISeCure-The ISC International Journal of Information Security, 11(3), 139-144.

Alexiadou M., Emmanouil S., Periklis G., and Theophilos P. 2023. Cryptocurrencies and Long-Range Trends. International Journal of Financial Studies 11: 40.

Bação P. & Duarte A.P. & Sebastião H. & Redzepagic S. (2018). Information Transmission Between Cryptocurrencies: Does Bitcoin Rule the Cryptocurrency World?, Scientific Annals of Economics and Business, Sciendo, vol. 65(2), pp 97-117.

Behrendt S., DimpflT., PeterF. J., Zimmermann D. J. (2019). RTransferEntropy — Quantifying information flow between different time series using effective transfer entropy, SoftwareX.

Best R. (2022). Number of cryptocurrencies worldwide from 2013 to November 2022, https://www.statista.com/statistics/863917/number-crypto-coi ns-tokens/

Blau B. (2018). Price dynamics and speculative trading in bitcoin. Res. Int. Bus. Financ. 43, 15– 21.

Bollerslev T.(1986). General Autoregressive Conditional Heteroscedasticity. Journal of Econometrics, 31, 307-327.

Box G., Jenkins G., and Reinsel G. (2008). Time Series Analysis, Wiley, Hoboken, NJ, USA, 4th edition.

Cheah E., Fry J. (2015). Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of bitcoin. Econ. Lett. 130, 32–36.

Chen X., Wei X., Su Z. Portfolio Investment Algorithm Based on LSTM-ARIMA Prediction Model. Automation and Machine Learning (2022) Vol. 3: 22-32.

Chuen K., David L., Guo L., & Wang Y. (2017). Cryptocurrency: A new investment opportunity? The Journal of Alternative Investments, 20(3), 16-40.

CNBC (2019). Top Economists Stiglitz, Roubini and Rogoff Renew Bitcoin Doom Scenarios, https://www.cnbc.com/2018/07/09/nobel-prize-winning-economist-joseph-stiglitzcriticizes-bitcoin.html (2018).

Corbet S., Meegan A., Larkin C., Lucey B., & Yarovaya L. (2018). Exploring the dynamic relationships between cryptocurrencies and other financial assets. Economics Letters, 165, 28-34.

Deng Y., Fan H. & Wu S. (2023). A hybrid ARIMA-LSTM model optimized by BP in the forecast of outpatient visits. J Ambient Intell Human Comput 14, 5517–5527 .

Engle R. 2002. Dynamic Conditional Correlation: A Simple Class of Multivariate Generalized Autoregressive Conditional Heteroscedasticity Models. Journal of Business & Economic Statistics. vol. 20. no. 3. pp. 339-350.

Engle R.F. and Sheppard, K. 2001. Correlation Multivariate GARCH: Theoretical and Empirical properties of Dynamic Conditional, NBER Working Paper no. 8554

Forbes (2023). January Crypto Market Outlook, https://www.forbes.com/advisor/

Gandal N. and Halaburda H. (2016). Can we predict the winner in a market with network effects? Competition in cryptocurrency market, Games 7.3, p. 16.

Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. Econometrica. 37 (3): 424–438.

Gu Y. (1926) Why do we sometimes get nonsense-correlations between time-series? A study in sampling and the nature of time-series. J R Stat Soc 89:1–63.

Hassler, U., Hosseinkouchack, M. (2022). Understanding nonsense correlation between (independent) random walks in finite samples. Stat Papers 63, 181–195.

Hendry DF (2000). On detectable and non-detectable structural change. Struct Chang Econ Dyn 11:45–65.

Hendry D.F., Clements M.P. (2003) .Economic forecasting: some lessons from recent research. Econ Model 20:301–329

Ho, T.K. (1995). Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

Hochreiter S., & Schmidhuber J. (1997). Long short-term memory. Neural Computation, 9 (8), 1735–1780.

Hyndman R. J. and Koehler A. B. (2006), Another look at measures of forecast accuracy, International Journal of Forecasting, 22, 679-688.

Hyndman R.J. 2014. Variations on Rolling Forecasts, https://robjhyndman.com/hyndsight/rolling-forecasts/

Hyndman R.J., Athanasopoulos G. 2014. Forecasting: Principles and Practice, OTexts.

Hyndman, R.J. and A.B. Koehler (2006). Another look seven groups: absolute forecasting error, percentage at measures of forecast accuracy. International forecasting error, symmetrical forecasting error, measures Journal of Forecasting, 22(4): 679-688.

Jang, H. and Lee, J. (2018) An Empirical Study on Modeling and Prediction of Bitcoin Prices with Bayesian Neural Networks Based on Blockchain Information. IEEE Access, 6, 5427-5437.

Ji S, Kim J, Im H (2019b) A comparative study of bitcoin price prediction using deep learning. Mathematics 7(10):898.

Johansen S. (1991). Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models. Econometrica. 59 (6): 1551–1580.

Kinateder, H. and Papavassiliou, V. G. (2019). Calendar Effects in Bitcoin Returns and Volatility. Finance Research Letters, Vol. 38, 101420, 2021.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. Physical review E 69(6): 066138.

Kumar, Manish and Thenmozhi, M., (2014), Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models, International Journal of Banking, Accounting and Finance, 5, issue 3, p. 284-308.

Kumar, S. (2019). Forecasting Cryptocurrency Prices using ARIMA and Neural Network: A Comparative Study. Journal of Prediction Markets, 13(2).

Lahajnar S.& Rozanec A. (2020). The correlation strength of the most important cryptocurrencies in the bull and bear market. Investment Management and Financial Innovations. 17. 67-81. 10.21511/imfi.17(3).2020.06.

Lahmiri S. and Bekiros S. (2019).Cryptocurrency forecasting with deep learning chaotic neural networks, Chaos, Solitons & Fractals , vol. 118, issue C, 35-40.

Liu Y., Tsyvinski A. (2021). Risks and Returns of Cryptocurrency. Technical report. National Bureau of Economic Research, https://www.nber.org/papers/w24877

Lütkepohl, H. (2010). Impulse response function. In: Durlauf, S.N., Blume, L.E. (eds) Macroeconometrics and Time Series Analysis. The New Palgrave Economics Collection. Palgrave Macmillan, London.

McNally, S., Roche, J. and Caton, S. (2018) Predicting the Price of Bitcoin Using Machine Learning. 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Cambridge, 21-23 March 2018, 339.

Moon H.R. and Perron, B. (2012). Beyond panel unit root tests: Using multiple testing to determine the non stationarity properties of individual series in a panel. Journal of Econometrics, 169(1), 29-33.

Obeng, C.,& Attor, C. (2022). Interconnection among cryptocurrencies: Using Vector Error Correction Model. International Journal of Entrepreneurial Knowledge, 10(2), 24-41. doi: 10.37335/ijek.v10i2.157.

Rebane J., Karlsson I., Denic S., & Papapetrou P. (2018). Seq2seq Rnns and Arima models for cryptocurrency prediction: A comparative study, SIGKDD Fintech, 18,2-6.

Romano, J.P., Shaikh, A.M., and Wolf, M. (2008). Control of the false discovery rate under dependence using the bootstrap and subsampling. Test, 17(3), 417.

Schreiber, T. (2000). Measuring Information Transfer. Physical Review Letters 85 (2): 461–64.

Sebastião, H., Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. Financ Innov 7, 3.

Selmi R., Tiwari A., Hammoudeh S. (2018). Efficiency or speculation? A dynamic analysis of the Bitcoin market. Econ. Bull. 38(4), 2037-2046.

Sezer O.B., Gudelek M.U., Ozbayoglu A.M.(2020). Financial time series forecasting with deep learning : A systematic literature review: 2005–2019, Applied Soft Computing, Volume 90,2020, 106181.

Shannon, C. E. (1948). A Mathematical Theory of Communication. Bell System Technical Journal. 27 (3): 379–423.

Siami-Namini, S., & Namin, A.S. (2018). Forecasting Economics and Financial Time Series: ARIMA vs. LSTM. ArXiv, abs/1803.06386.

Sims, C. (1980). Macroeconomics and Reality. *Econometrica*, 48(1), 1–48.

Sims, C. A. (1980), Macroeconomics and Reality, Econometrica 48, 1–48

Sungil K., Heeyoung K.(2016). A new metric of absolute percentage error for intermittent demand forecasts. International Journal of Forecasting. 32 (3): 669–679.

Swan, M. (2015) Blockchain: Blueprint for a New Economy. O'Reilly Media, Inc., Sebastopol, California.

Tofallis (2015). A Better Measure of Relative Prediction Accuracy for Model Selection and Model Estimation, Journal of the Operational Research Society, 66(8):1352-1362.

Tripathi M. and Tripathi B. (2022). Predictive Power of An Ensemble Model for Cryptocurrency Forecasting: Cryptocurrency Forecasting using Ensemble Modeling, The Journal of Prediction Markets 16(1)

Vapnik. V. (1995). The Nature of Statistical Learning Theory. Springer, New York

Verma, R., Sharma, D., & Sam, S. (2022). Testing of Random Walk Hypothesis in the Cryptocurrency Market. FIIB Business Review, 0(0), Volume 10.

Yule, G. U. (1926). Why Do We Sometimes Get Nonsense-Correlations between Time Series A Study in Sampling and the Nature of Time Series. Journal of Royal Statistical Society, 89, 1-64.

Zhang G. P. (2003).Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing, vol. 50, pp. 159–175

Zivot, E. and Andrews, K. (1992). Further Evidence On The Great Crash, The Oil Price Shock, and The Unit Root Hypothesis, Journal of Business and Economic Statistics, 10 (10), pp. 251–70.