Bitcoin spread log returns´ forecasting

Author: Lúcia Moreira
Date: 30/01/2020

Summary

Parametric ARMA(p, q) fitting with and without GARCH(p, q) residuals and exponential smoothing (ES) based non-parametric models were used for 21 days forecasting on Bitcoin spread (maximum minus the minimum of a day) based log returns (log(today/yesterday). The data set used for model development considers daily historical data from 2017 to 2019. Out-of-sample forecasts were compared with real market values extracted from January 1st to January 21st, 2020. Regarding the parametric modelling, a MA(2) process with an ARCH(1) model on the residuals provided the best fit to the dataset. A simple exponential smoothing with additive errors provided to be the best fitting to the Bitcoin log returns, when using non-parametric modelling, although fitting to the data was poor. The first-day ahead out-of-sample and in-sample forecast comparison with the spread log return values observed to those specific days were better predicted by the parametric model with GARCH. Moreover, when using the raw spread values for Bitcoin, the ES model fitting has improved substantially and also has improved the prediction on the first day ahead. However, the parametric model is able to correctly predict the market trend for two consecutive days ahead what can be a very strong advantage in the finance setting.

The dataset

The historic crypto currency market daily data for 2017 to 2019 was obtained using the *crypto_history()* function from the crypto package available in R. The function returns among other variables the Market date, Market open, Market high, Market low, Market close, Volume 24 hours, USD Market capitalization and the spread, defined as a volatility premium (high minus low for that day). The crypto coin herein considered is Bitcoin and particularly its *spread* value. Out-of-sample forecasts were performed for 10 days or 21 days ahead and are compared with the values extracted with the *crypto_history()* function from January 1st to January 21st, 2020.

Exploratory Data Analysis

Fig. 1 shows the daily history of the spread (maximum minus minimum of the day) variable for Bitcoin in US dollars; a total of 1096 days are considered. We can see a substantial volatility typical of cryptocurrencies; the highest spreads were observed by the end of 2017 beginning of 2018 and at middle 2019. This time series is visually not stationary and confirmed by the unit root test (ADF test- Ho: presence of unit root, p-value = 0.07 (fail to reject Ho for a 0.05 confidence level)). Moreover, visual inspection of the sample ACF indeed shows a very slow decay to zero (not shown).

Fig. 2 shows the spread log returns from that period of time; the plot evidences a similarly stationary time series. Table 1 shows that the Box Cox lambda coefficient (lambda minimizes the coefficient of variation for subseries of the time series) is 1, so no further transformation is indeed required and the number of differentiations required is actually zero, as expected. Analysis with the Augmented Dickey-Fuller test for the null that this time series has a unit root indicated a rejection of the null (p-value < 0.01) for a confidence level of 0.05, indicating that such log returns time series has indeed a stationary behavior (Table 1). This hypothesis was confirmed by the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for the null hypothesis that the time series is level or trend stationary that failed to reject the null (p-value > 0.1). These tests confirm that log returns introduce a stationary profile for non-seasonal behavior financial data.

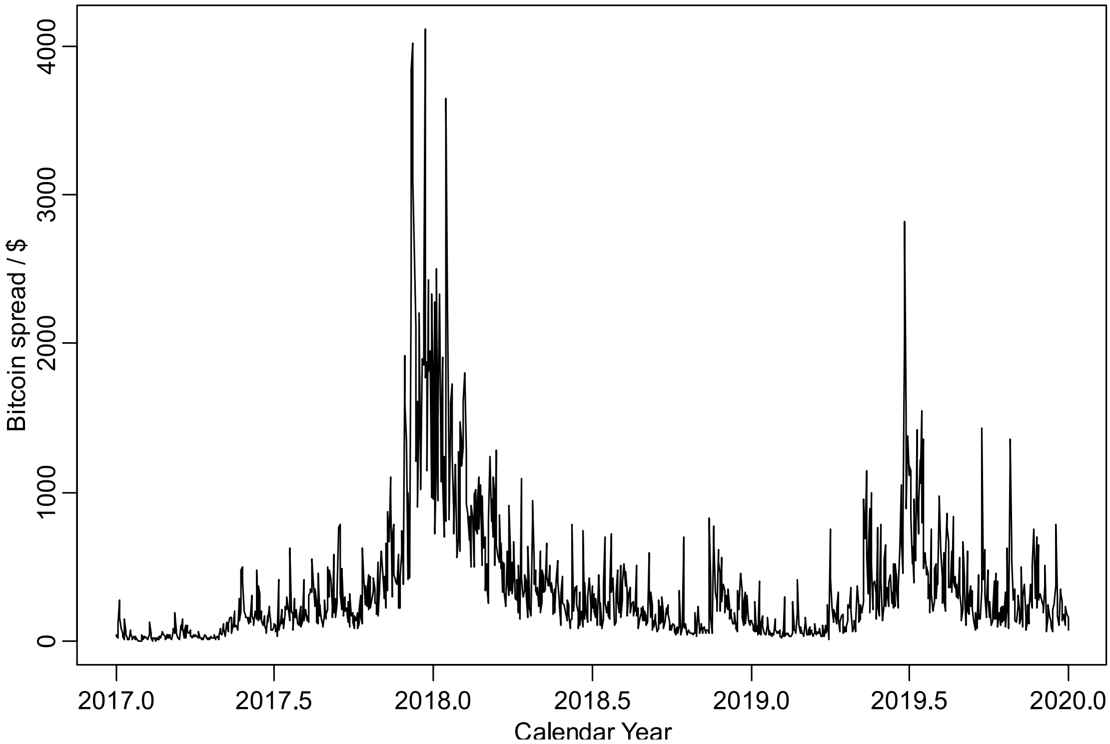


Fig. 1- Daily historical data for Bitcoin spread values from 2017 to 2019 (1096 days).

Analogously, departing from the original spread values and performing a Box Cox transformation would indeed result in a lambda value near zero (α = -0.02) thus indicating that a logarithmic transformation is the most adequate for this data set. Moreover, the further differentiation of the transformed data series would finally convert the data set into a stationary time series (see Table 1). So indeed log returns is the standard way to get a stationary time series in financial data, as previously documented.

Table 1 – Stationary data set analysis.

Time series	Box Cox (α)	ndiffs /-	ADF test p-value	KPSS test p-value
Original	-0.02	1	<0.01 (after transformation and differentiation)	> 0.1 (after transformation and differentiation)
Log returns	1.07	0	< 0.01	> 0.1

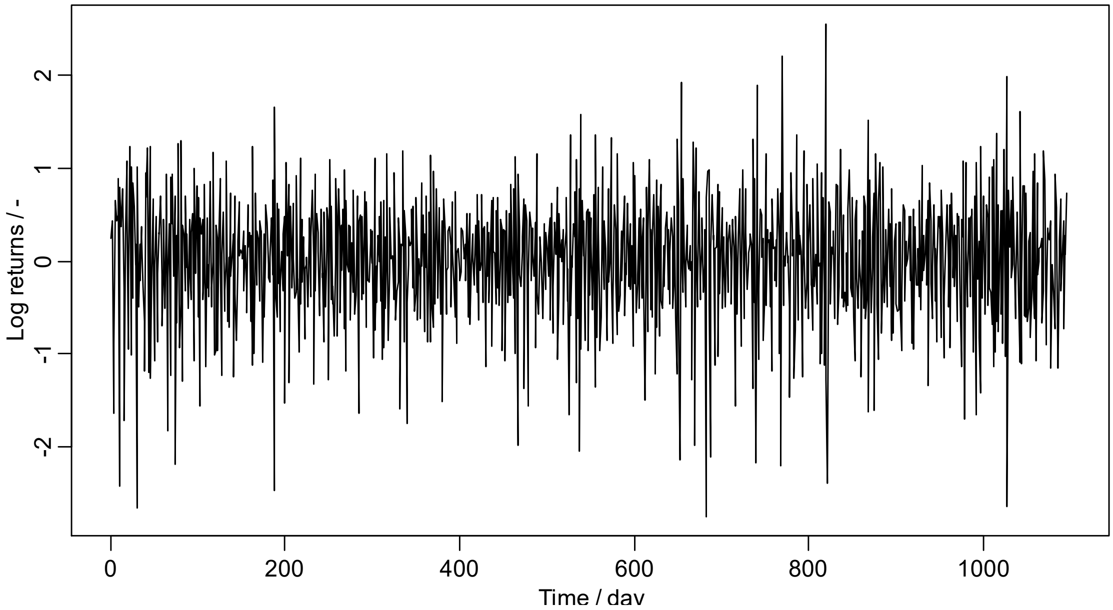


Fig. 2 - Daily log returns for Bitcoin from 2017 to 2019 (1096 days).

Fig. 3 shows the sample auto (ACF) and partial (PACF) correlation function from the log returns time series during 2 weeks. Sample ACF and PACF show that an ARMA(p,q) process may represent this data series, as both correlations decreases to zero and more particularly a MA(2) process once PACF is statistically zero for $q > 2$ for a 95 % confidence while PACF decays to zero.

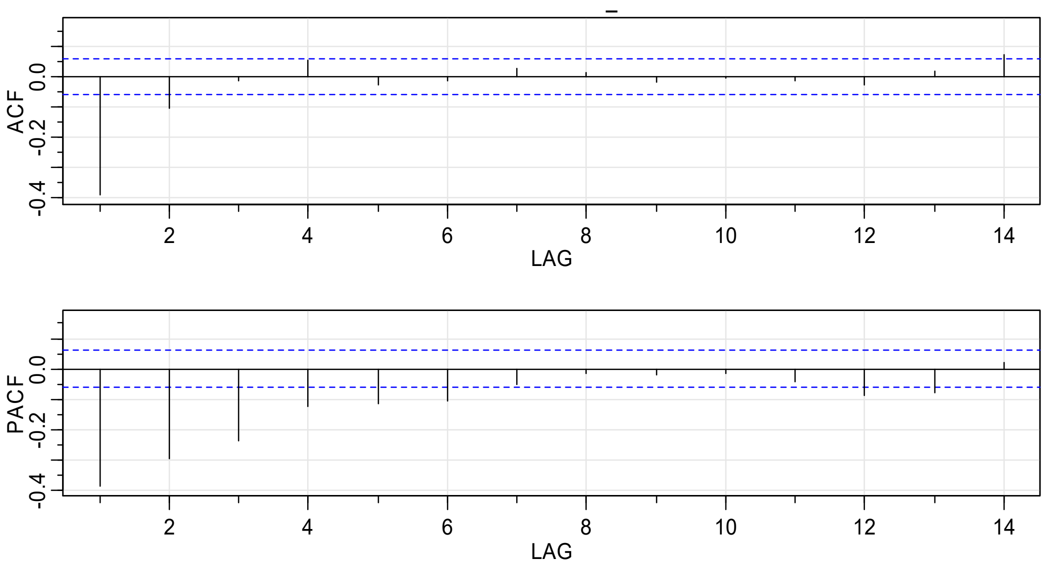


Fig. 3 – Sample autocorrelation and sample partial correlation.

Fig. 4 shows the auto-dependence plot when auto-correlations vanish. One can see that for lags higher than 1 autocorrelation is indeed very low while for lag = 1 some quite relevant negative correlation with lag 1 can be observed and in accordance to ACF. This behavior is indeed a characteristic of financial data.

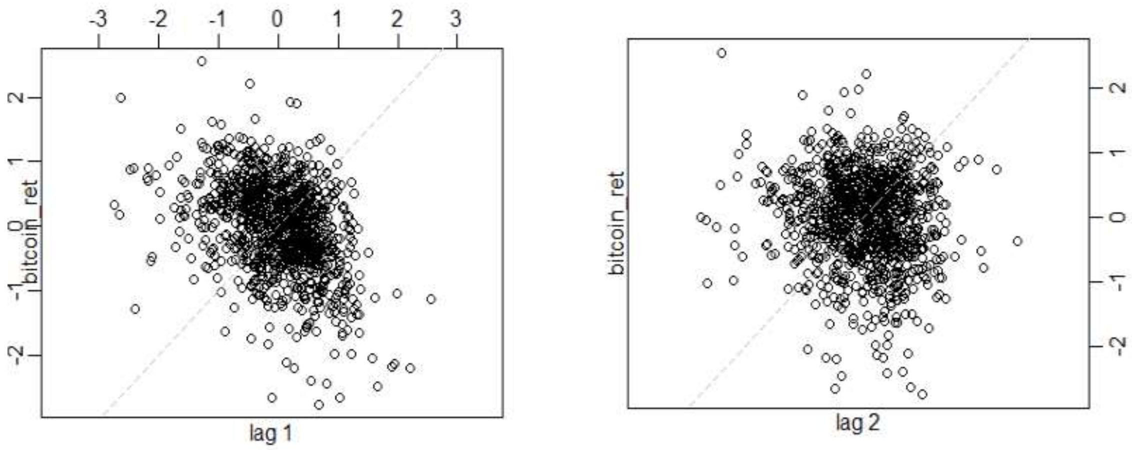


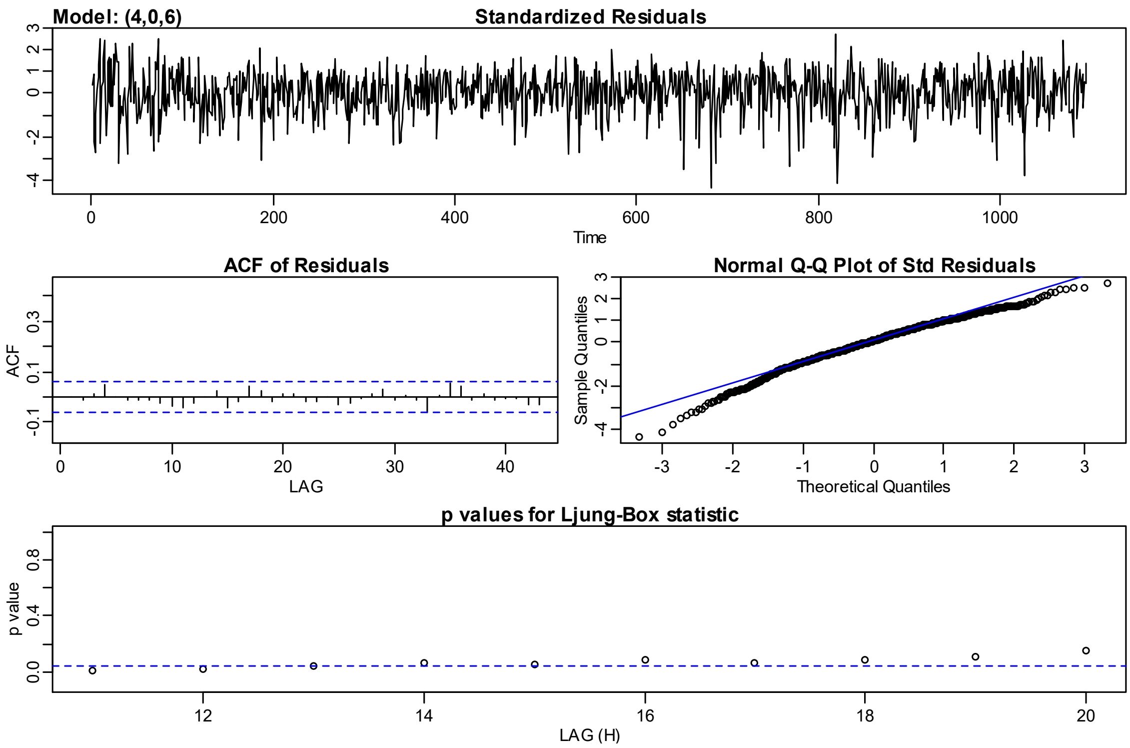
Fig. 4 - Lag plots up to lag = 2.

Parametric modelling

Based on the EDA performed on the previous section, the choice of the best ARMA model was performed for a short range of AR and MA orders using AIC and BIC criteria in order to confirm if the data set indeed points to a MA(2) process. For that, ARMA modelling included auto-regressive *p* orders from 0 to 9 simultaneously with moving average *q* orders also from 0 to 9, corresponding to the total fitting of 100 different ARMA models using *sarima()* function.

Fig. 5 shows the output (top) and summary (bottom) of the model when using AIC criterion. Based on AIC (lowest value) an ARMA(4,6) model is obtained. The estimated variance is 0.326 with all the coefficients statistically significant for a 95 % confidence. Also, the ACF of the residuals point to white noise (Barlett test: residuals are approximately iid) despite some of the p-values of the Ljung-Box statistic indeed show that some autocorrelations can be jointly significant. Also, normality distribution of the residuals fails at lower values and at the extremes.

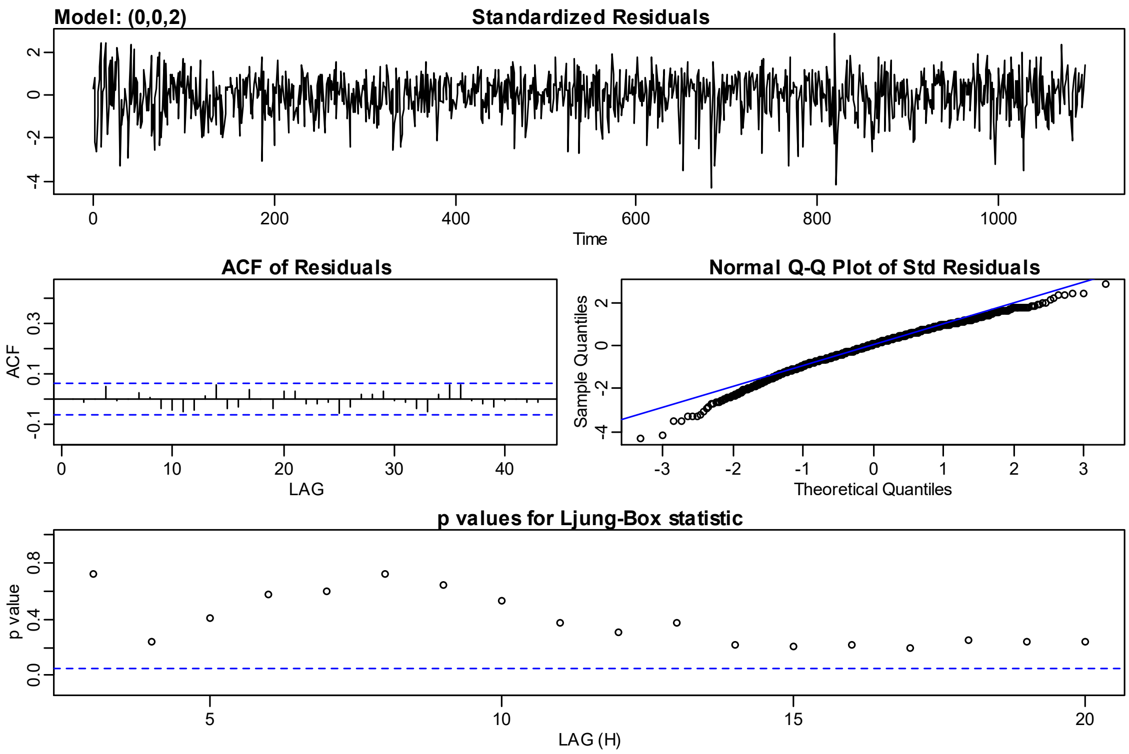
Regarding the BIC criterion, Fig. 6 shows the output (top) and summary (bottom) of the model obtained when using such criterion. Based on BIC (lowest value) a MA(2) is obtained. The estimated variance is 0.336 (slightly higher than with AIC) and all the coefficients are statistically significant. The ACF of the residuals point to white noise and all the p-values of the Ljung-Box statistic indeed show that autocorrelations are not jointly significant. Also, normality distribution of the residuals only fails at the extremes.




```
sigma^2 estimated as 0.326:  log likelihood = -943.19,  aic = 1910.39

$degrees_of_freedom
[1] 1084

$ttable
      Estimate     SE   t.value p.value
ar1     0.5380 0.0063   85.2549  0.0000
ar2    -1.1134 0.0109 -102.2714  0.0000
ar3     0.5282 0.0095   55.8608  0.0000
ar4    -0.9899 0.0102  -97.2353  0.0000
ma1    -1.1968 0.0567  -21.1000  0.0000
ma2     1.3612 0.0883   15.4245  0.0000
ma3    -1.2230 0.0807  -15.1591  0.0000
ma4     1.2482 0.0863   14.4612  0.0000
ma5    -0.6263 0.0822   -7.6217  0.0000
ma6    -0.0900 0.0436   -2.0675  0.0389
xmean  -0.0009 0.0040   -0.2238  0.8230

$AIC
[1] 1.744645

$AICc
[1] 1.744868

$BIC
[1] 1.799424
```

Fig. 5 - Output (top) and summary (bottom) of the ARMA(4,6) model obtained after using AIC criterion.

Owing to these results, the EDA previously performed and taking into consideration possible overfitting problems, the MA(2) model based on BIC was chosen as the best fitting to the spread log returns for Bitcoin.




```
sigma^2 estimated as 0.3356:  log likelihood = -956.36,  aic = 1920.72

$degrees_of_freedom
[1] 1092

$ttable
      Estimate     SE  t.value p.value
ma1    -0.6504 0.0302 -21.5474  0.0000
ma2    -0.1276 0.0308  -4.1474  0.0000
xmean  -0.0010 0.0039  -0.2460  0.8057

$AIC
[1] 1.75408

$AICc
[1] 1.7541

$BIC
[1] 1.772339
```

Fig. 6 - Output (top) and summary (bottom) of the MA(2) model obtained after using BIC criterion.

Once this type of time series is prone to present some non-constant volatility, a GARCH model of the MA(2) residuals was also considered to evaluate if there was any improvement on prediction. ACF and PACF analysis of the squared MA(2) model residuals (right) and log returns density compared to a normal distribution (left) are shown on Fig. 7. Log returns do not follow a normal distribution; moreover, log returns distribution shows that returns are more often positives than negatives. Undeniably, people can make money if would invest on this crypto coin but its distribution also shows that very big losses are denser than very big profits! Actually, financial data display evidence of asymmetric distributions (i.e., outcomes below or above the mean may carry different overall probabilities). This type of distribution can be an indication that GARCH modelling can be quite applicable.

Moreover, PACF and ACF of MA(2) model squared residuals show that some correlation at lag 1 may have not been completely explained by the model, although showing that only ca. 0.14 % of that correlation is not explained what may not be significant. However, and owing to the pedagogical character of this project, an analysis including a GARCH model was also herein considered.

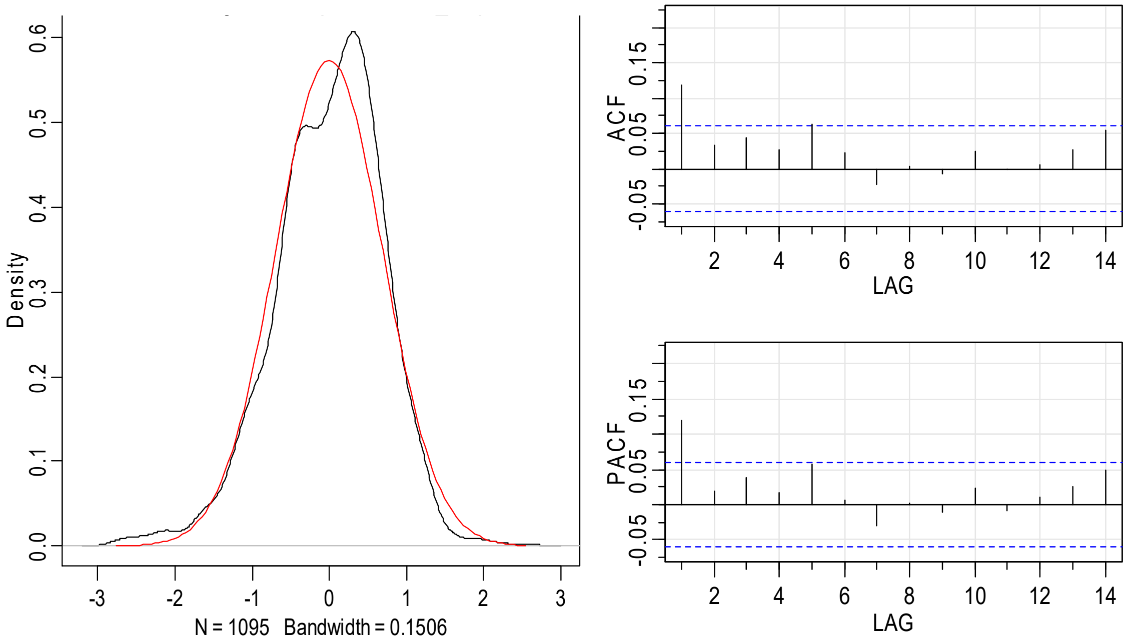


Fig. 7 - ACF and PACF analysis of the squared MA(2) model residuals (right) and log returns density compared to a normal distribution (left).

<u>GARCH modelling</u>

GARCH models may be suggested by an "ARMA-type" model by looking into the ACF and PACF. From Fig. 7 (right) a GARCH(1,1) for the squared MA(2) residuals may fit the observations (PACF indicates a AR(1) while ACF may suggest a MA(1)). However, the choice of the GARCH model to the residues was performed for a short range of orders using BIC criteria in order to confirm if the residues indeed may be pointing to a GARCH(1,1) process. For that, GARCH modelling included "auto-regressive"-type *p* orders from 1 to 5 simultaneously with "moving average"-type *q* orders from 0 to 5, corresponding to the total fitting of 30 different GARCH models using *garch()* function. The negative log-likelihood output from this function was suitably converted to BIC.

BIC criterion indicates that an ARCH(1) model is the best fit for the MA(2) squared residuals. Fig. 8 shows the output summary of the garchFit() function when using the MA(2) model with the selected ARCH(1). The alpha1 coefficient is statistically relevant and the BIC value (1.765) is slightly lower when comparing to the BIC (1.772) obtained from the MA(2) fitting only.

```
Error Analysis:
        Estimate  Std. Error  t value              Pr(>|t|)
mu      0.0002227   0.0034554    0.064             0.948623
ma1    -0.6593232   0.0339511  -19.420 < 0.0000000000000002 ***
ma2    -0.1394603   0.0333177   -4.186            0.0000284 ***
omega   0.2911671   0.0162450   17.924 < 0.0000000000000002 ***
alpha1  0.1371877   0.0403961    3.396             0.000684 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 -948.9907    normalized:  -0.8666581

Description:
 Tue Jan 28 14:59:25 2020 by user: Casa


Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  91.5615   0
 Shapiro-Wilk Test  R    W      0.9820281 0.0000000002186672
 Ljung-Box Test     R    Q(10)  8.681394  0.5625836
 Ljung-Box Test     R    Q(15)  18.31233  0.2466016
 Ljung-Box Test     R    Q(20)  24.31004  0.2291111
 Ljung-Box Test     R^2  Q(10)  11.01753  0.3561538
 Ljung-Box Test     R^2  Q(15)  15.66117  0.4049208
 Ljung-Box Test     R^2  Q(20)  20.99852  0.39722
 LM Arch Test       R    TR^2   12.07818  0.4394214

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
1.742449 1.765273 1.742407 1.751085
```

Fig. 8 - Summary of the MA(2) + ARCH(1) model obtained after using BIC criterion.

Non-parametric modelling

ES models were estimated using the ets() function from the forecast package using a three-letter code. The possible inputs are “N” for none, “A” for additive, “M” for multiplicative, or “Z” for automatic selection. So, the possibilities for each component are: Error ={={A,M}}, Trend ={={N,A,Ad}d} and Seasonal ={={N,A,M}}.

The simple exponential smoothing was the best fit obtained for this dataset when using automatic selection for the parameters involved. Fig. 9 (top) shows the summary of the ets() function for data with no clear trend or seasonal pattern. The alpha coefficient is nearly zero. Fig. 9 (bottom) shows the fitting (in red) comparison with the data, where one can see that the fitting is quite poor.

However, the previous exercise on ES (homework #4), that was also performed on a quite similar data set, showed a quite good fitting to the spread raw values (please see HW4 report). This way, an ES fitting to Bitcoin raw spread values was also herein performed. Fig. 10 shows the respective summary and fitting, indicating now that a multiplicative error with no trend and no seasonal pattern was the best fit for the dataset (as observed in HW4 report). The alpha value obtained is 0.17. Fig. 10 (bottom) also shows that a substantially better fitting is observed. As a non-parametric approach is used and less assumptions have to be done, the raw non-stationary data set may be better fitted. However, and unfortunately, this possible conclusion is out of the range of author´s knowledge.

```
ETS(A,N,N)

Call:
 ets(y = bitcoin_ret, model = "ZZZ", damped = NULL, alpha = NULL,

 Call:
     beta = NULL, gamma = NULL, phi = NULL, additive.only = FALSE,

 Call:
     lambda = NULL, biasadj = FALSE, restrict = TRUE, allow.multiplicative.trend = FALSE)

  Smoothing parameters:
    alpha = 0.0001

  Initial states:
    l = -0.0005

  sigma:  0.6967

     AIC     AICc      BIC
6875.904 6875.926 6890.900

Training set error measures:
                        ME      RMSE       MAE      MPE     MAPE      MASE       ACF1
Training set 0.00009040511 0.6960651 0.5477826 100.1522 100.1522 0.6073791 -0.3895603
```

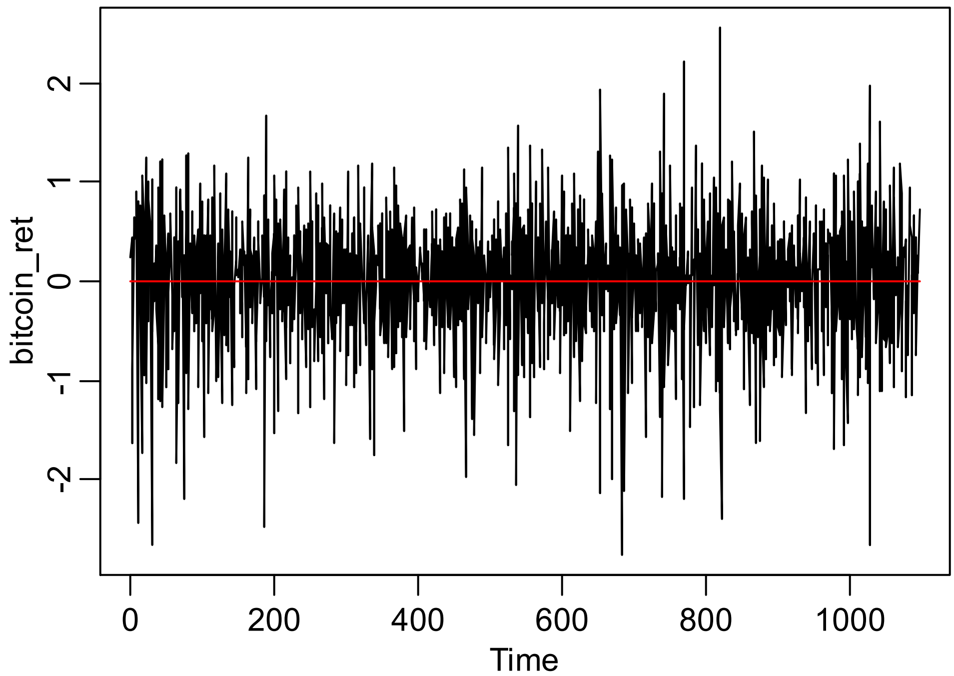


Fig. 9 – Summary (top) of exponential smoothing fitting to the Bitcoin log returns. Comparison of the fitted values (red) to the dataset (bottom).

```
ETS(M,N,N)

Call:
 ets(y = bitcoin, model = "ZZZ", damped = NULL, alpha = NULL,

 Call:
     beta = NULL, gamma = NULL, phi = NULL, additive.only = FALSE,

 Call:
     lambda = NULL, biasadj = FALSE, restrict = TRUE, allow.multiplicative.trend = FALSE)

  Smoothing parameters:
    alpha = 0.1732

  Initial states:
    l = 117.6399

  sigma:  0.8279

     AIC     AICc      BIC
19231.02 19231.04 19246.02

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set 0.3354896 285.9233 154.9277 -35.50485 59.40334 0.8599284 0.2822538
```

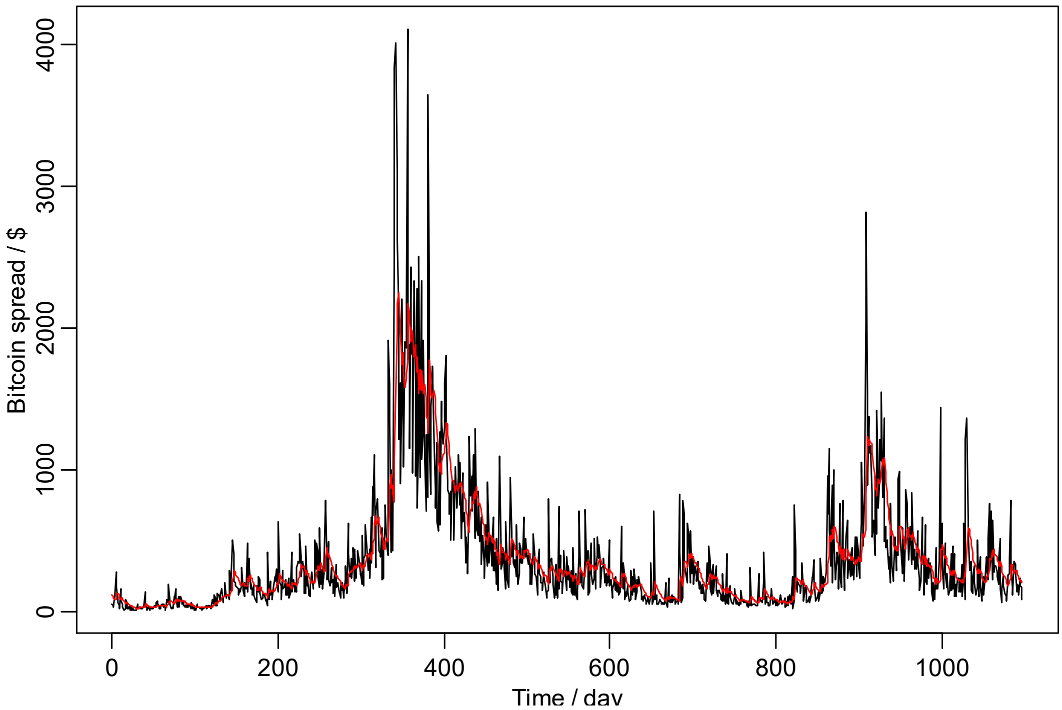


Fig. 10 – Summary (top) of exponential smoothing fitting to the Bitcoin spread. Comparison of the fitted values (red) to the dataset (bottom).

Forecasting in-sample and out-of-sample

Summary of the in-sample forecast predictions are shown in Table 2 for the first day ahead of the test set (last year of data) and the different models considered. In its turn, Table 3 shows the summary for the out-of-sample forecasts (21 days ahead). One year ahead forecasts, as requested, does not make sense on this type of dataset. Discussion in the following sections will be focused on out-of-sample forecasts because in this case data is also available for that period and because of that in both cases the methodology is very similar. When comparing in sample forecasts (Table 2) with out-of-sample forecasts (Table 3), ES model performs the best for in-sample forecast while Table 3 shows that for the out-f sample the MA(2) with GARCH turns out to be the best. This is related to the case that the actual log return value is near zero for the in-sample forecast and the ES indeed predict a nearly zero return for any day ahead. Please, see the detailed discussion on out-of-sample below.

Table 2 – In-sample root mean squared error (RMSE) for two days ahead and first day ahead forecast value for the different models tested.

Model	In-sample Forecast Day 1	In-sample Day 1	Out of sample RMSE
MA(2)	-0.559	0.10	0.51
MA(2) + ARCH(1)	-0.14	0.10	0.22
ETS (A,N,N)	0.00	0.10	0.14

Table 3 – Out of sample root mean squared error (RMSE) for two days ahead and first day ahead forecast value for the different models tested.

Model	Out of sample Forecast Day 1	Out of sample Day 1	Out of sample RMSE
MA(2)	-0.564	-1.25	0.59
MA(2) + ARCH(1)	-0.57	-1.25	0.58
ETS (A,N,N)	0.00	-1.25	0.98

Parametric modelling

Focusing on the out-of-sample case, forecasts for the MA(2) model were performed for 21 days ahead using sarima.for() function and were are compared with the values extracted with the crypto_history() function from January 1st to January 21st, 2020.

Fig. 11 (top) shows the plot obtained from sarima.for() output while Fig. 11 (bottom) shows the out-of-sample forecasts (red) with the 95 % confidence interval (blue) as well as the log returns for the test set (black). One can see that due to the type of model used (MA(2)) only the first two days ahead forecast show values different from the mean. The root mean squared error (RMSE) for 2 days ahead obtained for this model is 0.59 (Table 3). Table 3 also shows the first day ahead forecast value and the log return for the first day in the out-of-sample data set.

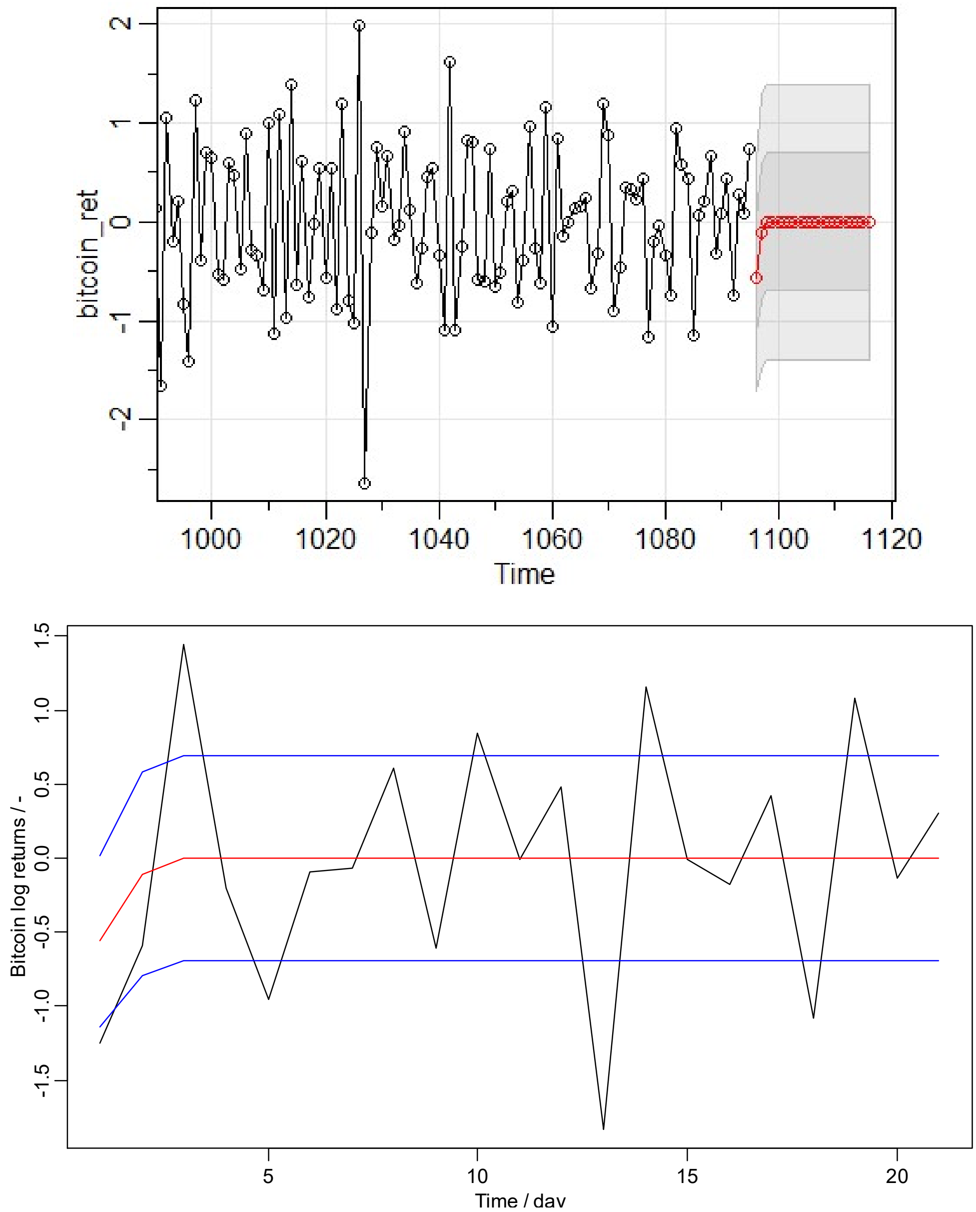


Fig. 11 – Plot output from sarima.for() (top); Out-of-sample forecasts (red) and the 95 % interval (blue) for the MA(2) model and 21 days ahead. Log returns for the out-of-sample data set (black) (bottom).

Fig. 12 (top) shows the MA(2) model out-of-sample forecast output with ARCH(1) while Fig. 12 (bottom) shows a more detailed comparison for the 21 days ahead. Table 3 shows the first day forecast value and the log return of the first day in the out-of-sample data set together with the RMSE obtained for the model that is slightly lower than the RMSE obtained from the MA(2) model (Table 3).

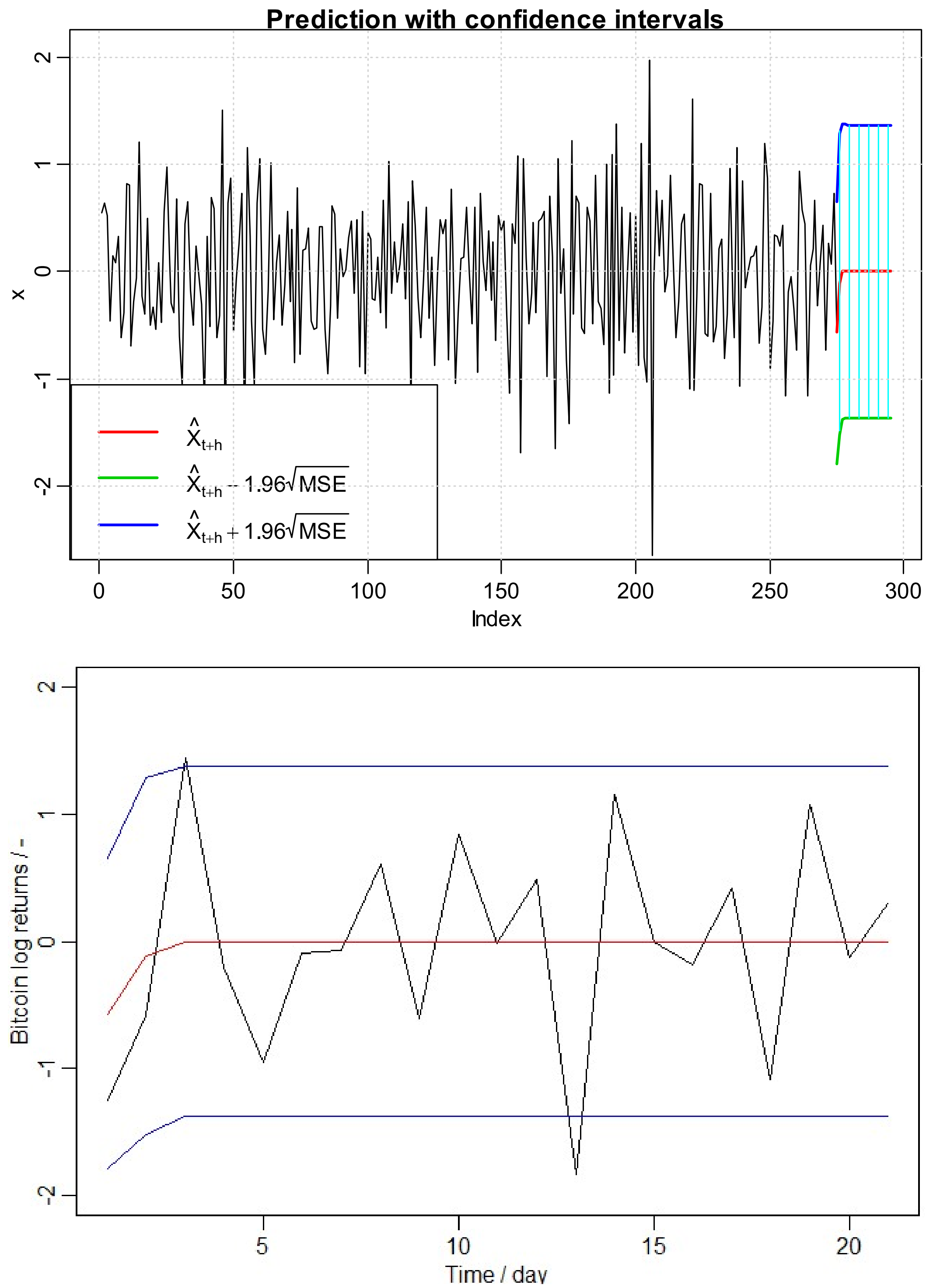


Fig. 12– Plot output from predict() function (top); Out-of-sample forecasts (red) and the 95% interval (blue) for the MA(2) model with ARCH(1) and 21 days ahead. Log returns for the out-of-sample data set (black) (bottom).

Non-parametric modelling

Fig. 13 (top) shows the ETS(A,N,N) model out-of-sample forecast output while Fig. 13 (bottom) shows a more detailed comparison for the 10 days ahead. Table 3 shows the first day ahead forecast value and the log return of the first day in the out-of-sample data set together with the RMSE obtained for the model that is the highest RMSE obtained (Table 3) indicating that ES non-parametric modelling performed the worse forecasts to Bitcoin log returns, due to the very poor fitting of the model to this data set. Moreover, Fig. 13 (bottom) shows that the first day ahead out-of-sample forecast is clearly not well predicted.

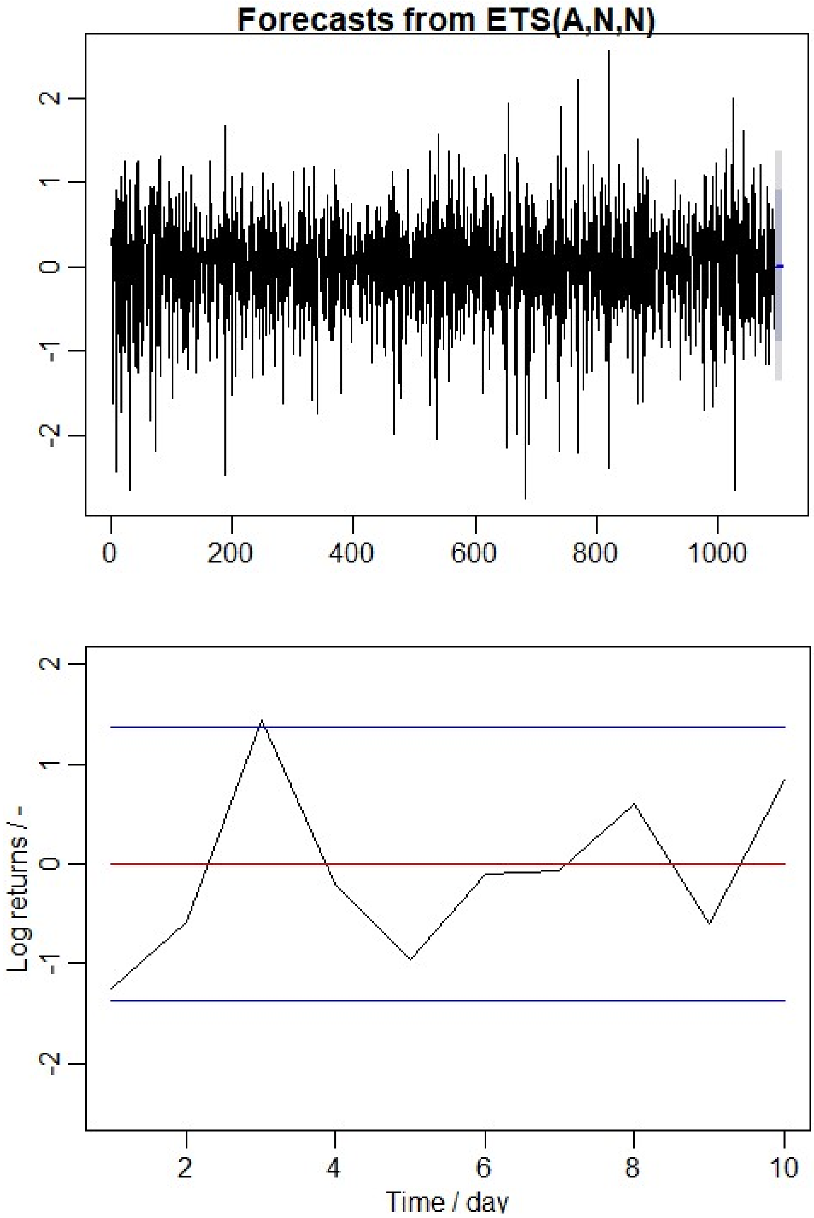


Fig. 13– Plot output from forecast() function (top); Out-of-sample forecasts (red) and the 95% interval (blue) for the ETS(A,N,N) model and 10 days ahead. Log returns for the out-of-sample data set (black) (bottom).

Finally, Fig. 14 (top) shows the ETS(M,N,N) model out-of-sample forecast output for the raw spread values while Fig. 14 (middle and bottom bottom) shows a more detailed comparison for the 10 days ahead for both the out-of-sample and in-sample forecasts, respectively. RMSE is not comparable with the values in Tables 2 and 3, so it is not shown. However, the first day ahead for both forecast types has improved, owing to the better ES fitting to the raw spread time series.

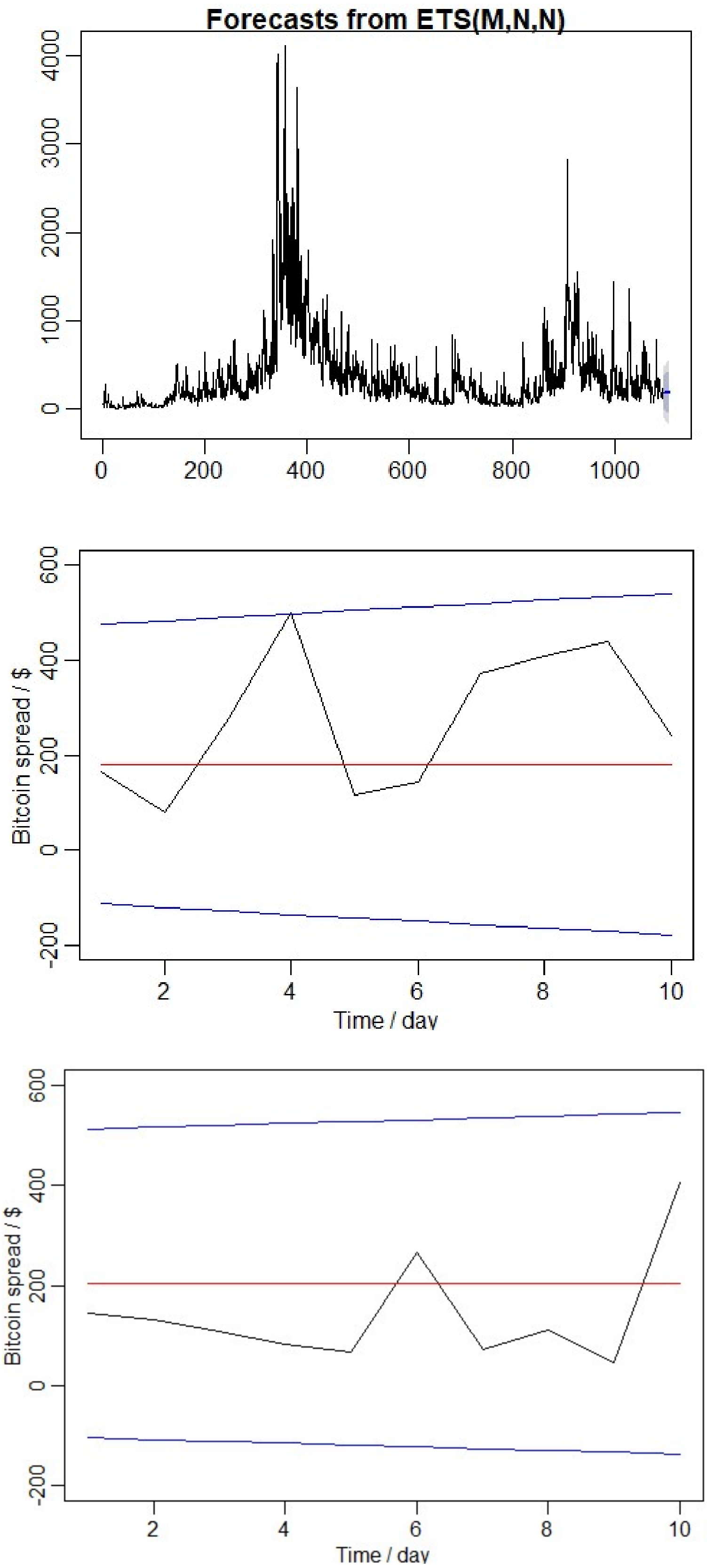


Fig. 14– Plot output from forecast() function (top); Out-of-sample forecasts (red) and the 95 % interval (blue) for the ETS(M,N,N) model and 10 days ahead. Bitcoin spread for the out-of-sample data set (black) (middle). In-sample forecasts (red) and the 95 % interval (blue) for the ETS(M,N,N) model and 10 days ahead. Bitcoin spread for the in-sample test set (black) (bottom).

Conclusions

Forecasting and modelling of the Bitcoin crypto currency was performed by using both parametric modelling and a non-parametric exponential smoothing approach. The parametric modelling indicated that a MA(2) process with an ARCH(1) model on the residuals provided the best fit to the spread log returns, while a simple exponential smoothing with additive errors provided to be the best fitting to the Bitcoin log returns, when using non-parametric modelling, although with a poor fitting to the data. The first-day ahead out-of-sample forecast comparison with the spread log return values observed to that specific day were better predicted by the parametric model. When using the raw spread values on the ES model the fitting has improved substantially and also improved the prediction on the out-of-sample first day ahead. However, the parametric model is able to better predict the market trend for two consecutive days ahead what can be a very strong advantage in the finance setting.