Forecasting Ethereum Return Using Linear and Nonlinear Time Series Models

Brian Godwin Lim[1]

[1]Department of Mathematics, Ateneo de Manila University

Abstract

In the last decade, cryptocurrencies continue to play a bigger role in the financial markets. Its attractive return made retail investors and institutional players consider allocating parts of their portfolio to the new type of asset. However, cryptocurrencies often provide greater risk than traditional assets. This paper attempted to model the Ethereum return series with five linear and nonlinear models with the goal of forecasting future returns. The results show that the two-regime Self-Exciting Threshold Autoregressive (SETAR) model with -0.04297 threshold has the best forecasting performance with a MSE of 0.002956. This model also captures several key characteristics of the return series.

Keywords: Ethereum, Cryptocurrency, Self-Exciting Threshold Autoregressive Model

1 Introduction

Background. Blockchain technologies have been one of the largest trend in the last decade. At its core, it is a "decentralized ledger of all transactions across a peer-to-peer network. Using this technology, participants can confirm transactions without a need for a central clearing authority" (PricewaterhouseCoopers, n.d.).

Advancements in this technology gave rise to cryptocurrencies - virtual currencies secured by cryptography, making it virtually impossible to counterfeit or double-spend (Frankenfield, 2021a). Compared to traditional assets such as stocks and bonds, cryptocurrencies are generally not issued by any central authority, making them theoretically immune to government interventions.

As of January 2021, there are over 4,000 cryptocurrencies in existence (Conway, 2021). Popular cryptocurrencies such as Bitcoin, Ethereum, and Dogecoin among others have presented attractive returns for investors and traders alike. Moreover, new developments have also been made in the last decade including the increasingly popular play-to-earn NFT gaming wherein players can earn real cash from playing games. As a result, retail investors as well as institutional players are now considering cryptocurrencies as an alternative to traditional assets and have even considered cryptocurrencies as part of their portfolio.

Despite the attractive returns, investing in cryptocurrencies also comes with great risks. Historical data shows that compared to traditional assets, cryptocurrencies have greater risk and volatility. One can lose more than 30% of a position in just a day. Investors in cryptocurrencies can potentially lose their entire portfolio if proper risk management is not exercised.

Problem. As cryptocurrencies continually play a bigger role in the financial markets, time series analysis can be a useful tool to model the price actions and returns. Furthermore, it can also give useful insights into risk management especially when dealing with a volatile assets.

One of the most popular cryptocurrency today is the Ethereum (ETH). It was launched in July 2015 by a small group of blockchain enthusiasts including the company's current CEO Vitalik Buterin. As of May 2021, Ethereum's market capitalization was estimated at \$500 billion compared to \$1.080 trillion for Bitcoin (Frankenfield, 2021b). As one of the highly traded cryptocurrency, it captures the high volatility present in the cryptocurrency market with a maximum daily return of 41% and a minimum daily return of -130% since its inception.

This paper aimed to utilize five linear and nonlinear time series models to predict future returns of Ethereum. A comparison was made in terms of the forecasting performance, measured by the mean squared error (MSE), of each model. It is hoped that with a better model for the returns, investors can make informed decisions on risk management.

Related Literature. The few literature available today explores the statistical characteristics of cryptocurrencies. However, most of these studies are focused on studying the prices and returns of Bitcoin as it is the leading the cryptocurrency market.

Radovanov, Marcikić, and Gvozdenović (2018) explored the four major cryptocurrencies: Bitcoin (BTC), Ethereum (ETH), Ripple (XRP) and Litecoin (LTC). Using an AR(1) model with $\text{GARCH}(1,1), \text{GJRGARCH}(1,1), \text{EGARCH}(1,1)$, they found that $\alpha + \beta \approx 1$ indicating the persistence of volatility over time. Moreover, they also found that in the case of BTC and ETH, there is only a small level of volatility asymmetry for daily returns while the models on XRP and LTC revels a positive asymmetry which contradicts other financial time series.

Similarly, Naimy and Hayek (2018) attempted to model Bitcoin volatility using GARCH models. They compared the $\text{GARCH}(1,1)$, EWMA, $\text{EGARCH}(1,1)$ models and found that the EGARCH outperforms the other two models in both in-sample and out-sample performance. However, the leverage coefficient γ was found to have a value 0.0113, a positive number contrary to what is expected, indicating that positive shocks are slightly more destabilizing.

Furthermore, Udom (2019) also attempted to model Bitcoin returns and volatility with a ARMA-GARCH model. The study found that $\text{ARIMA}(2,0,1) - \text{GARCH}(1,1)$ with normal distribution performs better compared to a pure $\text{ARIMA}(2,0,1)$ model as well as $\text{ARIMA}(2,0,1) - \text{GARCH}(1,1)$ model with t distribution and skewed t distribution. The researches obtained a minimum RMSE of 0.0372 when the out-sample data was from February 1, 2018 to July 31, 2019.

Likewise, in Gyamerah (2019), the researcher found that the return series of Bitcoin are leptokurtic. Additionally, the TGARCH model with a normal inverse Gaussian (NIG) distribution was found to be the most appropriate model for estimating Bitcoin returns compared to the t distribution and the generalized error distribution (GED).

Finally, Chappell (2018) applied the Markov regime-switching (MRS) models to Bitcoin returns. The researcher studied six cases of m-state MRS with $m \in \{2, \cdots, 7\}$. In the 2-state MRS, it was found that there is a presence of volatility clustering as indicated by the transition probability matrix where there is a high probability to remain in the same state in the next interval. However, the restricted 5-state model generated the optimal estimation for the sample in terms of Goodness-of-Fit test.

2 Methodology

This paper explored several linear and nonlinear time series models to be applied to the Ethereum return series with the goal of identifying the best model to forecast the returns of the cryptocurrency. The information criterion (IC), mean absolute error (MAE), and mean squared error (MSE) are the values investigated.

2.1 Linear Time Series Models

Augmented Dickey-Fuller Test. Before modeling the Ethereum return series using linear models, we first check for unit root in the series. The primary test used is the Augmented Dickey-Fuller (ADF) test. Suppose that

$$X_t = c_t + \beta X_{t-1} + \sum_{i=1}^{p-1} \phi_i \Delta X_{t-i} + a_t, \tag{1}$$

where $\Delta X_j = X_j - X_{j-1}$ and c_t is a function of time. The ADF test has a t-statistic given by

$$\text{ADF} = \frac{\hat{\beta} - 1}{\text{std}(\hat{\beta})}, \tag{2}$$

where H_0 : non-stationary and H_1 : stationary. Should the return series fail to reject the null hypothesis, we perform differencing on it and perform the test again. Otherwise, we proceed with fitting the series to the linear models.

ARMA-IGARCH Model. The first model to fit is the ARMA(p, q)-IGARCH$(1, 1)$ model which can be written as

$$\begin{aligned} r_t &= \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} + a_t - \sum_{j=1}^{q} \theta_j a_{t-j} \\ a_t &= \sigma_t \epsilon_t, \quad \sigma_t^2 = \alpha_0 + \beta_1 \sigma_{t-1}^2 + \alpha_1 a_{t-1}^2, \end{aligned} \tag{3}$$

where $\epsilon_t \sim N(0, 1)$, t distribution, or generalized error distribution (GED), $\alpha_1 = 1 - \beta_1$, and $0 < \beta_1 < 1$. This will be explored since it captures the persistence of volatility or past squared shocks $\eta_t = a_t^2 - \sigma_t^2$ in cryptocurrencies seen in Radovanov et al. (2018).

ARMA-EGARCH Model. The other linear model to fit is the ARMA(p, q)-EGARCH$(1, 1)$ model which accounts for the asymmetric (leverage) effects between positive and negative asset returns as seen in Naimy and Hayek (2018), albeit with different results than traditional assets. We consider the weighted innovation

$$g(\epsilon_t) = \theta \epsilon_t + \gamma \left[|\epsilon_t| - \mathbb{E}(|\epsilon_t|) \right] = \begin{cases} (\theta + \gamma)\epsilon_t - \gamma \mathbb{E}(|\epsilon_t|), & \text{if } \epsilon_t \geq 0 \\ (\theta - \gamma)\epsilon_t - \gamma \mathbb{E}(|\epsilon_t|), & \text{if } \epsilon_t < 0, \end{cases} \tag{4}$$

where θ, γ are some constants. The model can then be written as

$$\begin{aligned} r_t &= \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} + a_t - \sum_{j=1}^{q} \theta_j a_{t-j} \\ a_t &= \sigma_t \epsilon_t, \quad \ln(\sigma_t^2) = \alpha_0 + g(\epsilon_{t-1}) + \beta_1 \ln(\sigma_{t-1}^2). \end{aligned} \tag{5}$$

2.2 Nonlinear Time Series Models

F Test. To check for nonlinearity in the Ethereum return series, we use the F test by Tsay (1986) which is considered an improvement to the Keenan's test (1985) and the RESET test (1969). In practice, the test is the partial F statistic for testing $\boldsymbol{\alpha} = \mathbf{0}$ in the linear least-square regression

$$x_t = \boldsymbol{X'_{t-1}}\boldsymbol{\phi} + \boldsymbol{M'_{t-1}}\boldsymbol{\alpha} + a_t, \tag{6}$$

where $\boldsymbol{X_{t-1}} = (1, x_{t-1}, \cdots, x_{t-p})'$, $\boldsymbol{\phi} = (\phi_0, \cdots, \phi_p)'$, $\boldsymbol{M_{t-1}} = \text{vech}(\boldsymbol{X_{t-1}}\boldsymbol{X'_{t-1}})$ [1].

Self-Exciting Threshold Autoregressive (SETAR) Model. As an improvement to the linear AR(p) model, we investigated the k-regime SETAR model with threshold variable x_{t-d}. In its essence, the SETAR model is a piecewise linear AR model in the threshold space which can be written as

$$x_t = \phi_0^{(j)} + \phi_1^{(j)} x_{t-1} + \cdots + \phi_p^{(j)} x_{t-p} + a_t^{(j)}, \qquad \text{if } \gamma_{j-1} < x_{t-1} < \gamma_j, \tag{7}$$

where $j \in \{1, \cdots, k\}$ and $\gamma_i \in \mathbb{R}$ such that $-\infty = \gamma_0 < \gamma_1 < \cdots < \gamma_{k-1} < \gamma_k = \infty$. The superscript (j) is used to indicate the regime, where $\{a_t^{(j)}\}$ are iid with mean 0 and variance σ_j^2 that are mutually independent for different j, and the γ_j are the thresholds. This model was chosen due to its ability to capture the leverage effect (when the threshold value is set close to 0). It also provides greater flexibility compared to linear models and can provide better estimates of the conditional mean.

Logistic Smooth Transition Autoregressive (LSTAR) Model. A further improvement to the SETAR model is the LSTAR model which addresses the discontinuity of the thresholds in the SETAR model. It makes use of a logistic smooth function $0 \leq F(\cdot) \leq 1$ to produce a weighted linear combination of two AR(p) models

$$\begin{aligned} \mu_{1,t} &= c_1 + \sum_{i=1}^{p} \phi_{1,i} x_{t-i} \\ \mu_{2,t} &= c_2 + \sum_{i=1}^{p} \phi_{2,i} x_{t-i}, \end{aligned} \tag{8}$$

The weights are determined by $F\left(\dfrac{x_{t-1} - \Delta}{s}\right)$ with Δ, s as parameters representing the location and scale of the model transition.

Markov Switching Autoregressive (MSA) Model. The last model we investigate is the 2-regime MSA model where the transition is driven by a hidden two-state Markov chain. The model can be written as

$$x_t = \begin{cases} c_1 + \sum_{i=1}^{p} \phi_{1,i} x_{t-i} + a_{1,t}, & \text{if } s_t = 1 \\ c_2 + \sum_{i=1}^{p} \phi_{2,i} x_{t-i} + a_{2,t}, & \text{if } s_t = 2, \end{cases} \tag{9}$$

where s_t is the state at time t that assumes values $\{1, 2\}$ and is governed by a transition matrix $\boldsymbol{M}$ which gives the probabilities of the next state given the current state. This model was chosen since it captures the volatility clustering in Chappell (2018).

[1] If $\boldsymbol{A} = [a_{i,j}]_{k \times k}$, then $\text{vech}(\boldsymbol{A}) = (\boldsymbol{a'_1}, \boldsymbol{a'_{2*}}, \cdots, \boldsymbol{a'_{k*}})'$, where $\boldsymbol{a_1}$ is the first column of $\boldsymbol{A}$, and $\boldsymbol{a_{i*}} = (a_{i,i}, \cdots, a_{k,i})$ is a $(k - i + 1)$-dimensional vector.

3 Data

Ethereum Price Data. The Ethereum (ETH) price data used in this paper was obtained from the Coinbase website. It is worth noting that unlike traditional assets, cryptocurrency markets are open 24/7. Therefore, daily price data are available. The price data obtained was from August 9, 2015 to July 25, 2021, a total of 2178 data points.

Figure 1: Candlestick Chart of ETH Daily Price in USD

Ethereum Return Series. We convert these price data into daily returns with

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right), \tag{10}$$

where P_t are the daily closing prices. With this conversion, we are left with 2177 data points.

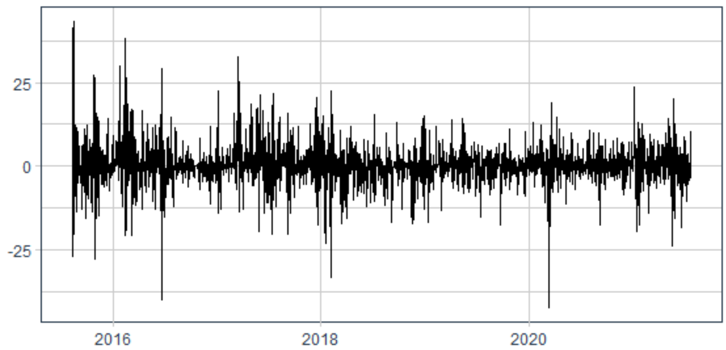

Figure 2: ETH Daily Log Return in Percentage

Inspecting the distribution of the return series, we obtain a kurtosis of 6.4172 indicating a leptokurtic distribution for the returns. Figure 3 shows the density plot of the return. This observation is in line with the results of Gyamerah (2019) wherein the return series of cryptocurrencies are leptokurtic. Table 14 also presents a summary of the return series.

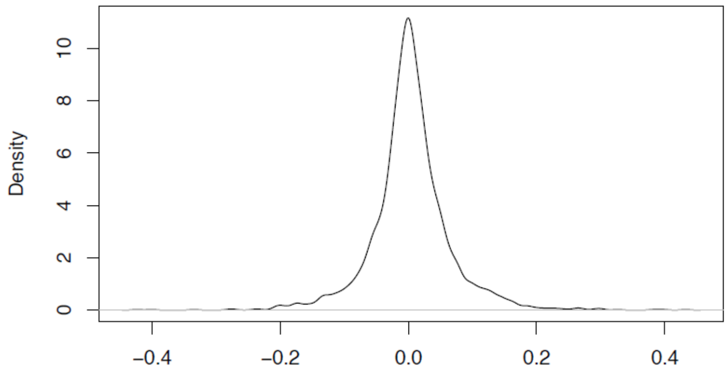

Figure 3: Density Plot of Return

Mean	Median	Max	Min	Std. Dev.
0.36%	0.14%	43.27%	-42.36%	0.0637

Table 1: Summary Statistics on Ethereum Return

4 Results and Discussion

Augmented Dickey-Fuller Test. Before fitting linear time series models, we performed the Augmented Dickey-Fuller test on the return series. The results are as follows:

lag	test statistic	p-value
5	-18.5287	0.01
7	-15.9762	0.01

Table 2: Results of the Augmented Dickey-Fuller Test

Therefore, at $\alpha = 0.05$, we reject the null hypothesis and the return series is stationary. Thus, we can proceed with fitting to linear models.

ARMA Order Determination. To determine the order of the ARMA component, we use the Box–Jenkins method. We plot the autocorrelation function (ACF), partial autocorrelation function (PACF) in Figure 5, and the extended autocorrelation function (EACF) in Figure 4

```
AR/MA
  0 1 2 3 4 5 6 7 8
0 o o x o o o o o o
1 x o x o o o o o o
2 o o x o o o o o o
3 x x x o o o o o o
4 x o x x o o o o o
5 x x x x o o o o o
```

Figure 4: EACF on Ethereum Return

Therefore, the possible models are ARMA$(0, 0)$, AR(3), and MA(3).

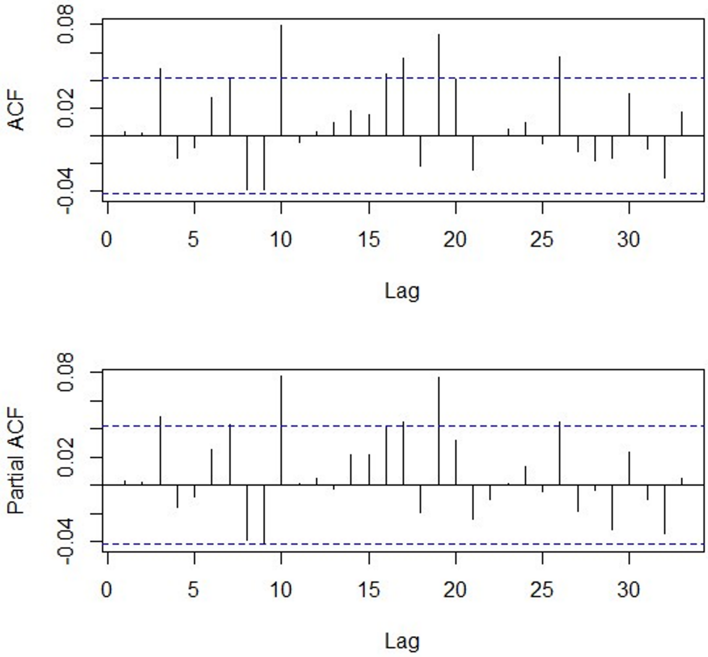

Figure 5: ACF and PACF on Ethereum Return

ARMA-IGARCH Model. Using the volatility model IGARCH$(1, 1)$ with $\epsilon_t \sim t$ distribution (due to leptokurtic distribution), along with the three possible models in the previous page, we obtain the following results:

Model	AIC	HQIC	MSE	MAE
ARMA(0,0)-IGARCH(1,1)	-3.1031	-3.0992	0.002963	0.040250
AR(3)-IGARCH(1,1)	-3.1012	-3.0945	0.002966	0.040240
MA(3)-IGARCH(1,1)	-3.1008	-3.0941	0.002966	0.040240

Table 3: Information Criterion and Forecast Errors on the ARMA-IGARCH Models

Note that the out-sample results was obtained when refitting every 7 days with a recursive window and a forecast length of 365 days.

In the in-sample result, for both information criterion, the ARMA$(0, 0)$-IGARCH(1,1) model produced the least value followed by the AR(3)-IGARCH(1,1) model. In the out-sample result, the same model obtained the least MSE. Therefore, ARMA$(0, 0)$-IGARCH$(1, 1)$ is the best model out of the three. The optimal parameters for the model are as follows:

Parameter	Estimate	Std.Error	p-value
ϕ_0	0.001386	0.000815	0.089053
α_0	0.000226	0.000053	0.000019
α_1	0.250065	0.031720	0.000000
β_1	0.749935	-	-

Table 4: Optimal Parameters for ARMA(0,0)-IGARCH(1,1)

The parameters in Table 4 indicate the persistence of volatility in the return series with $\alpha_1 + \beta_1 = 1$ and a statistically significant value for α_1. We also obtain a slightly positive mean ϕ_0 although not statistically significant. However, we do get a statistically significant value for α_0.

ARMA-EGARCH Model. Using the volatility model EGARCH$(1,1)$ with $\epsilon_t \sim t$ distribution (due to leptokurtic distribution), along with the three possible ARMA models, we obtain the following results:

Model	AIC	HQIC	MSE	MAE
ARMA(0,0)-EGARCH(1,1)	-3.1053	-3.0996	0.002965	0.040270
AR(3)-EGARCH(1,1)	-3.1042	-3.0956	0.002969	0.040260
MA(3)-EGARCH(1,1)	-3.1038	-3.0952	0.002969	0.040270

Table 5: Information Criterion and Forecast Errors on the ARMA-EGARCH Models

Note that the out-sample results was obtained when refitting every 7 days with a recursive window and a forecast length of 365 days.

Similarly, in the in-sample result, for both information criterion, the ARMA$(0,0)$-EGARCH(1,1) model produced the least value. In the out-sample result, the same model obtained the least MSE. Therefore, ARMA$(0,0)$-EGARCH$(1,1)$ is the best model. The optimal parameters for the model are as follows:

Parameter	Estimate	Std.Error	p-value
ϕ_0	0.001176	0.000831	0.156932
α_0	-0.328102	0.075633	0.000014
β_1	0.940888	0.013297	0.000000
θ	0.028942	0.023765	0.223287
γ	0.426877	0.050487	0.000000

Table 6: Optimal Parameters for ARMA(0,0)-EGARCH(1,1)

Table 6 shows that the model obtained a statistically significant positive value for γ indicating that positive news have greater impact on the returns, similar with Naimy and Hayek (2018). Additionally, it also shows a slightly positive value for the mean ϕ_0 although not statistically significant similar to the last model. However, we obtained a statistically significant negative value for α_0 which violates the constraints of the model.

In both linear models, the return series better fits a white noise series compared to an ARMA process. The two models produced a mean slightly greater than zero although not statistically significant. The volatility modeling component shows an almost similar forecasting performance. However, the IGARCH(1,1) model is preferred since it does not violate any constraints on the parameters.

F Test. After fitting linear time series models, we now fit the Ethereum daily return with nonlinear time series models. First, we perform the F test to check against the null hypothesis that the return series follows some AR process. The results are as follows:

AR order	test statistic	p-value
20	1.8672	0.01

Table 7: Results of the F test

Therefore, at $\alpha = 0.05$, we reject the null hypothesis and the return series does not follow a linear AR process. Thus, we can proceed with fitting nonlinear models to the series.

Self-Exciting Threshold Autoregressive Model (SETAR). As shown in Figure 5, the return series can be an AR(3) model. We compare two two-regime SETAR models: one with 0 threshold and the other with nonzero threshold. We obtain the following results with bootstrap method for 365 days forecasting:

Threshold	AIC	MSE	MAE
0	-12030	0.002964	0.040230
-0.04297	-12063	0.002956	0.040210

Table 8: Information Criterion and Forecast Errors on the SETAR Models

In both in-sample and out-sample results, the second model with nonzero threshold performed better than the first model with fixed 0 threshold. The optimal parameters for this model are as follows:

Parameter	Estimate	Std.Error	p-value
$\phi_{1,0}$	-0.013778	0.006868	0.044971
$\phi_{1,1}$	-0.201899	0.067079	0.002644
$\phi_{1,2}$	-0.235735	0.041989	0.000000
$\phi_{1,3}$	0.011203	0.043507	0.796810
$\phi_{2,0}$	0.002725	0.001580	0.084733
$\phi_{2,1}$	0.030573	0.029461	0.299506
$\phi_{2,2}$	0.091036	0.024440	0.000200
$\phi_{3,3}$	0.053924	0.024025	0.024901

Table 9: Optimal Parameters for SETAR Model with Nonzero Threshold

The parameter values in Table 9 indicate that a negative return less than the threshold value -0.04297 tends to switch to a positive return due to the (relatively) larger negative coefficients in the first regime $\phi_{1,1}, \phi_{1,2}$ which are both statistically significant. However, a return larger than the threshold tends to takes a longer time to reduce to a negative value due to the smaller magnitude in the coefficients of the second regime $\phi_{2,2}, \phi_{2,3}$ both of which are also statistically significant.

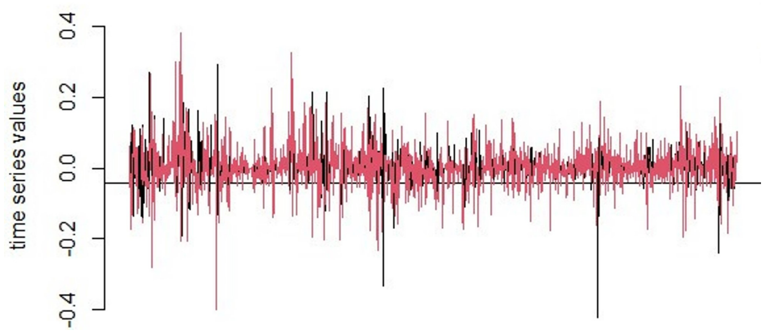

Figure 6: Regime Switching Plot for SETAR Model with Nonzero Threshold

Figure 6 above shows that majority of the time, the return series is in the second regime (red) when r_{t-1} is greater than the threshold value. We also notice that most of the large positive returns are in the second regime while most of the large negative returns are in the first regime indicating a clear distinction between the two regimes.

Logistic Smooth Transition Autoregressive Model (LSTAR). As an improvement to the last model, we fit the Ethereum return to a LSTAR model with AR(3). The results with bootstrap method for 365 days forecasting are as follows:

AIC	MSE	MAE
-12059	0.002961	0.040477

Table 10: Information Criterion and Forecast Errors on the LSTAR Model

The MSE indicates a better forecasting performance compared to a two-regime SETAR model with 0 threshold, but performs slightly worse compared to the same model with nonzero threshold. The optimal parameters of this model are:

Parameter	Estimate	Std.Error	p-value
$\phi_{1,0}$	-0.010573	0.009940	0.287502
$\phi_{1,1}$	-0.180616	0.080184	0.024290
$\phi_{1,2}$	-0.276519	0.051918	0.000000
$\phi_{1,3}$	0.019979	0.052458	0.703313
$\phi_{2,0}$	0.012845	0.010489	0.220688
$\phi_{2,1}$	0.219804	0.080862	0.006563
$\phi_{2,2}$	0.370215	0.058281	0.000000
$\phi_{2,3}$	0.031614	0.061053	0.604582
$1/s$	100.000003	53.003021	0.059203
Δ	-0.054967	0.009241	0.000000

Table 11: Optimal Parameters for LSTAR Model

where Δ, s are the location and scale of the model. Similar to the SETAR model, Table 11 shows that the "threshold" of the LSTAR model Δ is a negative value near zero which is also statistically significant. Δ can be seen as the threshold point where the weight of one regime is greater than the other. Similar to the SETAR model, we also see statistically significant negative parameter values for the first regime $\phi_{1,1}, \phi_{1,2}$. However, the LSTAR model has larger and statistically significant positive parameters in the second regime $\phi_{2,1}, \phi_{2,2}$ that was not present in the SETAR model.

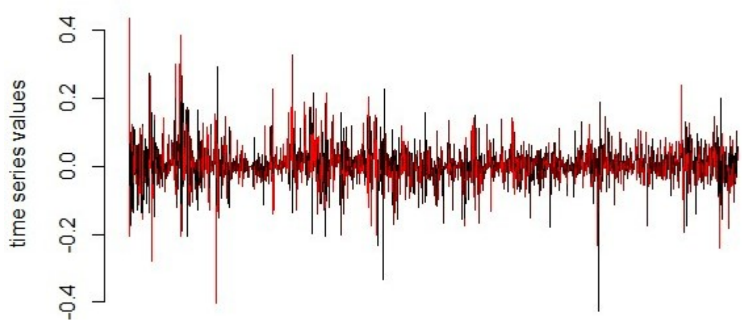

Figure 7: Regime Switching Plot for LSTAR Model

Figure 7 above shows a balance between the two regimes. Compared to Figure 6, the LSTAR model seems to have an equal distribution of large positive and large negative returns in both regimes. This difference can be attributed to the dynamic shifting of weights of both regimes as compared to having two constant AR models.

Markov Switching Autoregressive (MSA) Model. For the last model, we fit another regime switching model but with a stochastic scheme. Again, we use a 2-regime AR(3) process. We obtain the following results:

BIC	MSE	MAE
-6558.16	0.003890	0.047288

Table 12: Information Criterion and Forecast Errors on the MSA Model

The MSE and MAE indicates that the Markov Switching Autoregressive Model performs poorly compared to the SETAR and LSTAR model. Despite this, the MSE value is small enough to make the model useful. The optimal parameters of the model are:

Parameter	Estimate
$\phi_{1,0}$	-0.030332
$\phi_{1,1}$	0.249614
$\phi_{1,2}$	-0.025913
$\phi_{1,3}$	-0.015495
$\phi_{2,0}$	0.052308
$\phi_{2,1}$	0.364788
$\phi_{2,2}$	-0.003394
$\phi_{2,3}$	0.062573

Table 13: Optimal Parameters for MSA Model

with a Markov transition matrix and a matrix of limiting transition probabilities of

$$\boldsymbol{M} = \begin{bmatrix} 0.982098 & 0.017902 \\ 0.147357 & 0.852643 \end{bmatrix}, \qquad \lim_{p\to\infty} \boldsymbol{M}^p = \begin{bmatrix} 0.891674 & 0.108326 \\ 0.891674 & 0.108326 \end{bmatrix}. \tag{11}$$

The Markov transition matrix in (11) indicates the presence of volatility clustering since the probability of remaining in the same state is high and the first regime has a standard deviation 16% greater than the second regime. Figure 8 below shows that the first regime (shaded region) has a larger volatility than the second regime. This result is similar to the results of Chappell (2018). Unlike the two previous nonlinear models, the current model does not classify regimes based on the signs of the return, rather it classify regimes based on volatility of the returns. Moreover, the matrix of limiting transition probabilities indicate that in the long-run, almost 90% of the series will be in the first regime of higher volatility.

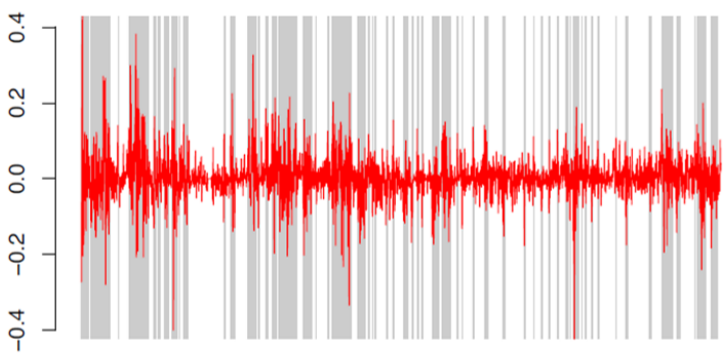

Figure 8: Regime Switching Plot for MSA Model

Summary of Results. Table 14 presents a summary of the results of the five models investigated. Results show that out of the five models fitted, the SETAR model with nonzero threshold performed the best in terms of forecast performance followed by the LSTAR model while the MSA model performed the worse. The SETAR model is given by

$$r_t = \begin{cases} -0.013778 - 0.201899 r_{t-1} - 0.235735 r_{t-2} + 0.011203 r_{t-3} & \text{if } r_{t-1} \leq -0.04297, \\ 0.002725 + 0.030573 r_{t-1} + 0.091036 r_{t-2} + 0.053924 r_{t-3} & \text{if } r_{t-1} > -0.04297. \end{cases} \quad (12)$$

Upon further inspection of the SETAR model, we see that it captures several characteristics on the return series and price of Ethereum:

1. **The model captures the asymmetry between positive and negative returns.** In line with the results of the ARMA(0,0)-EGARCH(1,1) model, positive returns are more destabilizing than negative returns due to the difference in the magnitudes of the coefficients in both regimes. Negative returns (less than the threshold) are followed by positive returns on average due to the large magnitude of $\phi_{1,1}, \phi_{1,2}$.

2. **The parameters of the model also reflects the overall uptrend of the value of Ethereum.** In line with the first observation, a large negative return less than the threshold is immediately supported, causing a positive return in the next period. A large positive return, however, is not immediately followed by a negative return indicating a weak resistance. The net effect is an overall uptrend in the price data.

3. **The model captures the leptokurtic distribution of the returns.** The model was able to empirically forecast large positive and negative returns. The first regime captures most of the large negative returns while the second regime captures the large positive returns.

Model	AIC	BIC	HQIC	MSE	MAE
ARMA(0,0)-IGARCH(1,1)	-3.1031	-	-3.0992	0.002963	0.040250
ARMA(0,0)-EGARCH(1,1)	-3.1053	-	-3.0996	0.002965	0.040270
SETAR ($\gamma \neq 0$)	-12063	-	-	0.002956	0.040210
LSTAR	-12059	-	-	0.002961	0.040477
MSA	-	-6558.16	-	0.003890	0.047288

Table 14: Summary of Models

5 Conclusion

This paper attempted to model the Ethereum return series with five linear and nonlinear models. Using the mean squared error (MSE) as the measure of forecast performance, we found that the Self-Exciting Threshold Autoregressive (SETAR) model performed the best out of the models investigated. The model was able to forecast 1 year of returns with an MSE of 0.002956, an improvement from the MSE obtained by Udom (2019). The model also captures several key characteristics of the Ethereum return series. However, further improvements to the current model are still needed. The current model exhibits an ARCH effect according to the McLeod-Li test. Future studies can incorporate volatility modeling to the current model to capture the volatile nature of cryptocurrency returns. With this, the new model might be able to provide a better forecasting performance.

References

Chappell, D. (2018, September). *Regime Heteroskedasticity in Bitcoin: A Comparison of Markov Switching Models* [MPRA Paper]. Retrieved 2021-07-26, from https://mpra.ub.uni-muenchen.de/90682/

Conway, L. (2021, June). *The 10 most important cryptocurrencies other than bitcoin.* Retrieved 2021-07-26, from https://www.investopedia.com/tech/most-important-cryptocurrencies-other-than-bitcoin/

Frankenfield, J. (2021a, May). *Cryptocurrency.* Retrieved 2021-07-26, from https://www.investopedia.com/terms/c/cryptocurrency.asp

Frankenfield, J. (2021b, July). *Ethereum.* Retrieved 2021-07-26, from https://www.investopedia.com/terms/e/ethereum.asp

Gyamerah, S. A. (2019). Modelling the Volatility of Bitcoin Returns Using GARCH Models. *Quantitative Finance and Economics*, *3*(4), 739–753. Retrieved 2021-07-26, from http://www.aimspress.com/rticle/doi/10.3934/QFE.2019.4.739 doi: DOI: 10.3934/QFE.2019.4.739

Naimy, V. Y., & Hayek, M. R. (2018). Modelling and predicting the Bitcoin volatility using GARCH models. *International Journal of Mathematical Modelling and Numerical Optimisation*, *8*(3), 197. Retrieved 2021-07-26, from http://www.inderscience.com/link.php?id=88994 doi: DOI: 10.1504/IJMMNO.2018.088994

PricewaterhouseCoopers. (n.d.). *Making sense of bitcoin, cryptocurrency and blockchain.* Retrieved 2021-07-26, from https://www.pwc.com/us/en/industries/financial-services/fintech/bitcoin-blockchain-cryptocurrency.html

Radovanov, B., Marcikić, A., & Gvozdenović, N. (2018, November). A Time Series Analysis of Four Major Cryptocurrencies. *Facta Universitatis, Series: Economics and Organization*, *0*(0), 271–278. Retrieved 2021-07-26, from http://casopisi.junis.ni.ac.rs/index.php/FUEconOrg/article/view/3893 doi: DOI: 10.22190/FUEO1803271R

Tsay, R. S. (1986). Nonlinearity tests for time series. *Biometrika*, *73*(2), 461–466. Retrieved 2021-07-27, from https://www.jstor.org/stable/2336223 doi: DOI: 10.2307/2336223

Udom, E. X. (2019). Estimating and Forecasting Bitcoin Daily Returns Using ARIMA-GARCH Models. *International Journal of Science and Research (IJSR)*. Retrieved 2021-07-26, from https://www.semanticscholar.org/paper/Estimating-and-Forecasting-Bitcoin-Daily-Returns-Udom/aaba437935df4052ed9f6a7ff9ee40f36a3df7aa

Appendix: R Codes

Load the libraries.

```
library(quantmod)
library(TSA)
library(tseries)
library(forecast)
library(rugarch)
library(tsDyn)
library(MSwM)
library(NHMSAR)
library(nonlinearTseries)
library(xts)
library(ggplot2)
library(normtest)
library(e1071)
```

Load the data.

```
# Get ETH-USD data
ETH=read.csv("ETHUSD.csv")
dates=as.character(ETH$Date)
ETH$Date=as.POSIXct(dates,format="%m-%d-%Y")

# Plot candlestick
ggplot(ETH,aes(x = Date, y = Close)) +
  geom_candlestick(aes(open = Open, close = Close, high = High, low = Low)) +
  labs(title = "", x = "", y = "") + theme_tq()

# Get closing prices
close=ETH[,2]

# Get daily return (close to close)
ret=na.omit(diff(log(close)))

# Cut ETH to fit return
ETH=ETH[2:dim(ETH)[1],1:dim(ETH)[2]]
ETH$Return=ret

# Plot Returns
ggplot(ETH,aes(x = Date, y = 100*Return)) +
  geom_line() +
  labs(title = "", x = "", y = "") + theme_tq()

# Convert ETH into xts
ETH$Date=NULL
ETH=xts(ETH, as.POSIXct(dates[2:length(dates)], format="%m-%d-%Y"))

# Check for leptokurtic
kurtosis.norm.test(ret)
plot(density(ret))
kurtosis(ret)
```

Check ARMA order.

```
adf=adf.test(ret,k=7)
adf
eacf(ret)
```

ARMA-IGARCH(1,1) order.

```
spec1=ugarchspec(variance.model=list(model="iGARCH",garchOrder=c(1,1)),
                 mean.model=list(armaOrder=c(0,0)),distribution.model="std")
mod1=ugarchfit(spec=spec1,data=ret)
mod1
roll1=ugarchroll(spec=spec1,data=ret,n.ahead=1,forecast.length=365,
                 refit.every=7,refit.window="recursive")
report(roll1,type="fpm")
```

ARMA-EGARCH(1,1) order.

```
spec2=ugarchspec(variance.model=list(model="eGARCH",garchOrder=c(1,1)),
                 mean.model=list(armaOrder=c(0,0)),distribution.model="std")
mod2=ugarchfit(spec=spec2,data=ret)
mod2
roll2=ugarchroll(spec=spec2,data=ret,n.ahead=1,forecast.length=365,
                 refit.every=7,refit.window="recursive")
report(roll2,type="fpm")
```

Nonlinearity test.

```
test=nonlinearityTest(ret)
test
```

Get training and test dataset.

```
ret.test=ts(ret[1813:2177],frequency=365.25,start=c(2020,7,26))
ret.train=ts(ret[1:1812],frequency=365.25,start=c(2020,7,26))

ret.arr.train=array(ret.train,c(length(ret.train),1,1))
ret.arr.test=array(ret.test,c(length(ret.test),1,1))
```

SETAR Model.

```
mod3.setar=setar(ret,m=3)
summary(mod3.setar)
mod3.forecast=setar(ret.train,m=3)
pred3=predict(mod3.forecast,type="bootstrap",n.ahead=365)$pred
pred3=ts(pred3,frequency=365.25,start=c(2020,7,26))
accuracy(pred3,ret.test)
```

LSTAR Model.

```
mod4.lstar=lstar(ret,m=3,th=1)
summary(mod4.lstar)
mod4.forecast=lstar(ret.train,m=3,th=1)
pred4=predict(mod4.forecast,type="bootstrap",n.ahead=365)$pred
pred4=ts(pred4,frequency=365.25,start=c(2020,7,26))
accuracy(pred4,ret.test)
```

MSA Model.

```
ret.arr=array(ret,c(length(ret),1,1))
theta.init=init.theta.MSAR(ret.arr,M=2,order=3)
mod5=fit.MSAR(ret.arr,theta.init)
summary(mod5)
mod5$BIC
regimes.plot.MSAR(mod5,ret.arr)
theta.init

theta.init=init.theta.MSAR(ret.arr.train,M=2,order=3)
mod5.forecast=fit.MSAR(ret.arr.train,theta.init)
summary(mod5.forecast)
pred5=prediction.MSAR(data=ret.arr.test,theta.init,ex=1:1)
mse=mean((ret.arr.test-pred5$y.p)^2)
mae=mean(abs(ret.arr.test-pred5$y.p))
mse
mae
```