Ethereum Volatility Rate Prediction

Ishita Singh (is26)

STAT429-Time Series Analysis

Prof. Hyoeun Lee

UIUC

Contributions
● Collected and cleaned daily ETH-USD price data from Investing.com. ● Stationarity checks for the time series plot & differencing. ● Built and compared multiple ARMA/ARIMA models; selected *ARIMA(4,0,4)* using AIC and residual checks. ● Extended to a seasonal model and selected *SARIMA(5,1,5)*(1,0,1)[7]* to capture weekly patterns. ● Evaluated SARIMA with a rolling 7-day window, computing RMSE for each window and visualising forecast stability over time. ● Modelled volatility using *GARCH(1,1)*, interpreted parameters and ran diagnostic tests. ● Produced 7-day ahead forecast vs actual plots for ARIMA, SARIMA and GARCH with confidence intervals and RMSE.

Abstract

The goal of this project is to turn Ethereum's noisy price history into short-term predictions using ARIMA, SARIMA and GARCH models. Using historical daily ETH-USD prices, I first cleaned and transformed the data into log-returns so that the series is more stable and easier to model. I then built a set of forecasting models in R, starting with classical time-series tools (ARIMA) and then adding weekly seasonality (SARIMA) to capture patterns that repeat every 7 days. After comparing different model combinations, I selected a best ARIMA model and a best SARIMA model based on AIC, likelihood, and residual diagnostics (checking that the residuals behave like random noise).

To see how these models perform in a realistic setting, I used a rolling 7-day evaluation: the model is re-trained on a moving window of past data and asked to forecast the next week, again and again over the sample. This gave a distribution of RMSE values instead of just one number, showing how forecast accuracy changes over time and during both calm and volatile periods.

However, crypto markets are not just about predicting the average price, volatility matters a lot. The residuals from ARIMA/SARIMA still showed changing volatility over time, so I fitted a GARCH(1,1) model on the returns to specifically model and forecast this time-varying risk. The GARCH model helps answer questions like "how uncertain might the next few days be?" rather than just "what is the next price?"

Overall, the project shows that:

- ARIMA and SARIMA can provide reasonable short-term price forecasts,
- accuracy varies over time, especially during high-volatility periods, and
- a GARCH model adds an important layer by modelling volatility instead of treating all days as equally risky.

Introduction

Motivation

Crypto markets are generally exciting, volatile, and hard to read. For traders and investors, even a small edge in understanding short-term moves can mean better entry/exit decisions, smaller drawdowns, and more realistic expectations.

Ethereum is one of the most important blockchains, sitting at the centre of DeFi, NFTs and smart-contract activity. Yet most discussions around ETH price are based on gut feeling, Twitter sentiment, or one-off charts, rather than a systematic, data-driven view of its behaviour over time.

This project is motivated by three simple needs:

1. How much of Ethereum's short-term price movement is actually predictable from past data, and how much is just noise?
2. Are there weekly patterns or recurring structures in ETH prices that a model like SARIMA can capture?
3. Crypto returns are very volatile. Even if we can't perfectly forecast prices, can we at least forecast volatility (risk) using GARCH and use that to judge how "calm" or "nervous" the market is likely to be?

Problem Statement

This project aims to answer the following core question:

> **Given historical daily Ethereum prices, how well can we forecast the next 7 days of price and volatility using time-series models?**

The project focuses on:

1. Forecasting with ARIMA/SARIMA
    a. Build and compare ARIMA and seasonal SARIMA models on log-transformed Ethereum prices.
    b. Use a rolling 7-day walk-forward evaluation to measure forecast accuracy (RMSE) over time.

2. Volatility Modelling with GARCH
    a. Fit a GARCH(1,1) model on daily log-returns to model volatility.
    b. Evaluate whether the GARCH model adequately captures volatility clustering and passes standard residual diagnostics.

3. Practical Insight
    a. Summarise what these models tell us about:
        i. How predictable ETH's short-term prices really are, and

ii. How useful volatility forecasts can be for understanding periods of high versus low market risk.

Dataset Overview

The analysis is based on a daily Ethereum price dataset downloaded from Investing.com. It contains one row per calendar day of trading and includes the following raw variables:

- date – Trading date.
- price – Daily closing price of ETH in USD.
- open – Opening price in USD.
- high – Highest traded price during the day in USD.
- low – Lowest traded price during the day in USD.
- vol – Reported trading volume for the day.
- change_percent – Daily percentage change in the closing price.

After cleaning, the dataset contains about 1,850 daily observations (from Nov 2020-2025), covering a multi-year period of Ethereum's price history, including both calm phases and highly volatile bull/bear cycles.

For modelling, we created a few key derived series:

- log_price: Log of the closing price, used for ARIMA/SARIMA modelling to stabilise variance.
- d_log_price: First difference of log_price, used to check stationarity and as an input to some ARMA-type models.
- log_return: Daily log returns (difference of log prices), used as the input series for GARCH(1,1) volatility modelling.

We then split the cleaned log_price series into:

- Training set: first ~80% of the timeline (used to fit ARIMA/SARIMA/GARCH models).
- Test set: remaining ~20% of the timeline (used for 7-day ahead evaluation and rolling-window forecasts).

This structure allows us to assess not just how well the models fit the historical data, but how well they generalise to new, unseen days in Ethereum's price history.

Statistical Methods

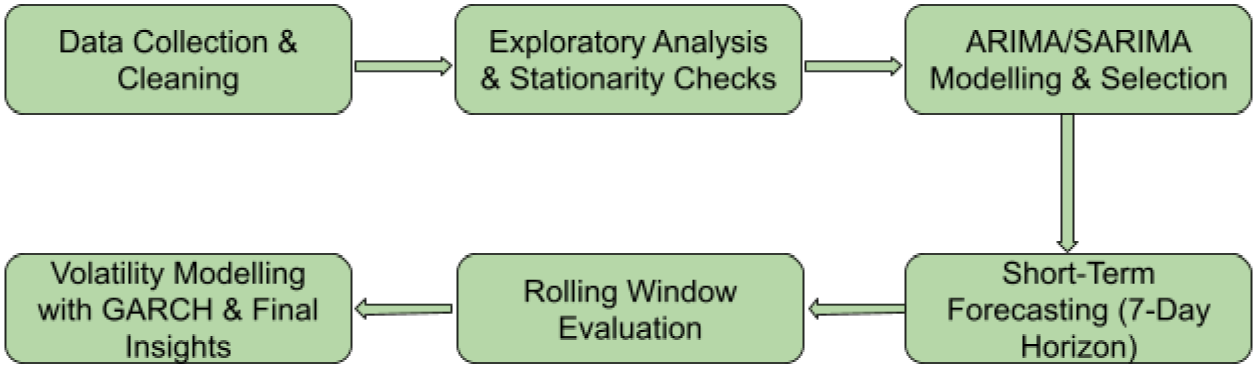

Fig1. Pipeline of the forecasting

Data Preprocessing

In all, the following are the preprocessing steps taken to make the raw Ethereum data ready for modelling:

1. **Loading the raw file:** I started from the CSV downloadable from Investing.com, containing 1,851 daily observations with the columns Date, Price, Open, High, Low, Vol. and Change %.

2. **Cleaning column names:** original variable names included spaces, capital letters and symbols. I therefore developed a cleaning function to turn them into simple and consistent names like date, price, open, high, low, vol and change_percent. This makes it easier to write code and avoids many errors later on.

3. **Parsing dates and ordering the series:** The date field was stored as character strings; for example, "11/25/2025". I converted it to a proper Date type and ordered this dataset in chronological order from the earliest to the latest day to ensure that the time series develops in a straightforward manner without jumps.

4. **Convert price to numeric:** The column of prices can include characters for commas. I removed these characters and converted the price, open, high, and low into numeric variables so that they could be used directly in calculations and models.

5. **Feature selection:** In order to do the forecasting, I kept only the relevant variables needed, which are directly useful for this task: date, open, high, low, and price. Volume and percentage change were not used within the main models; therefore, it was decided to drop those variables in order to keep the dataset compact.

6. **Checking for missing values and gaps:** Last but not least, I searched for missing values and for missed calendar days by comparing the observed dates against a full daily date sequence between the first and last observation. No missing value or gaps in dates was found apart from the expected first NA in log_return.

After these steps, the dataset is a clean, continuous daily time series of Ethereum prices and returns, suitable for *ARIMA, SARIMA and GARCH* modelling.

Time series plot

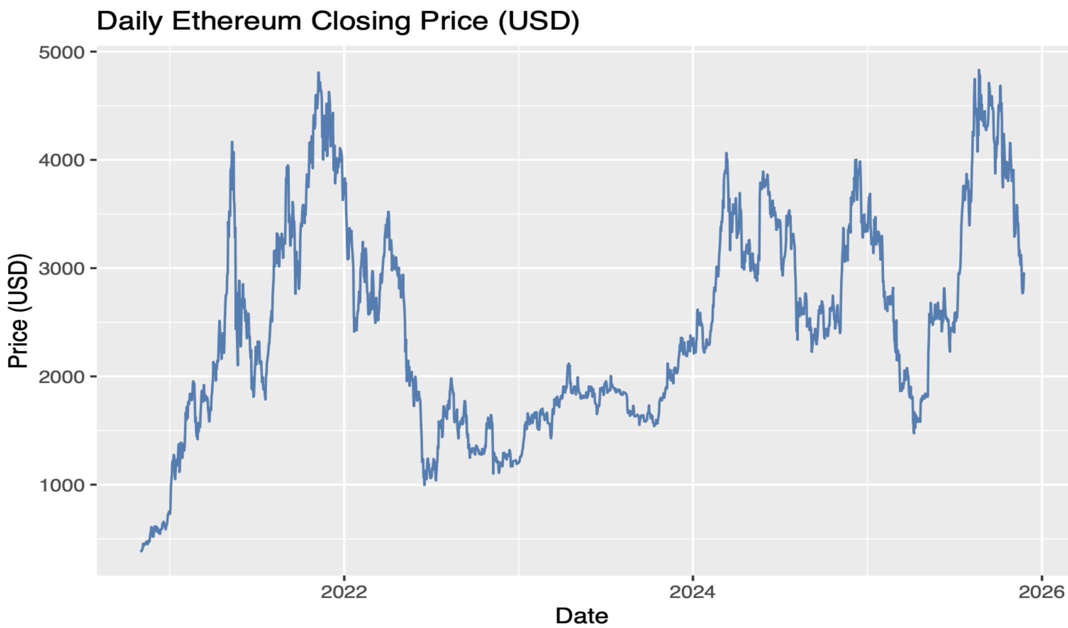

Fig 2. Raw time series plot.

This plot shows the daily closing price of Ethereum (in USD) over the full sample period. Prices move in big waves: there are sharp run-ups to peaks above $4,000-4,800, followed by steep crashes back near $1,000-1,500 and several large rebounds afterwards. Overall, the series is highly volatile, with frequent and sizable swings rather than a smooth, steady trend.

Stationarity

To build good time series models, the Ethereum price series needs to be *stationary*, its average level and spread should not drift too much over time.

ADF test on **l**og_price (train) had a p-value ~0.086 so failed to reject unit-root. Therefore, the series was non-stationary. Then, I took the logarithm of the daily closing price. Next, I *differenced* the log series (today's log price minus yesterday's log price). This step removes the long-term upward and downward trend so that we focus on the day-to-day movements around zero. The differenced log series looks like a "white-noise" cloud with no obvious trend.

Finally, I formally checked stationarity using the Augmented Dickey-Fuller (ADF) test. *The ADF statistic was -10.919 with a p-value of 0.01*, so we accept the alternative that the differenced log prices are stationary. This transformed series is what I use as input for the ARIMA, SARIMA, and GARCH models.

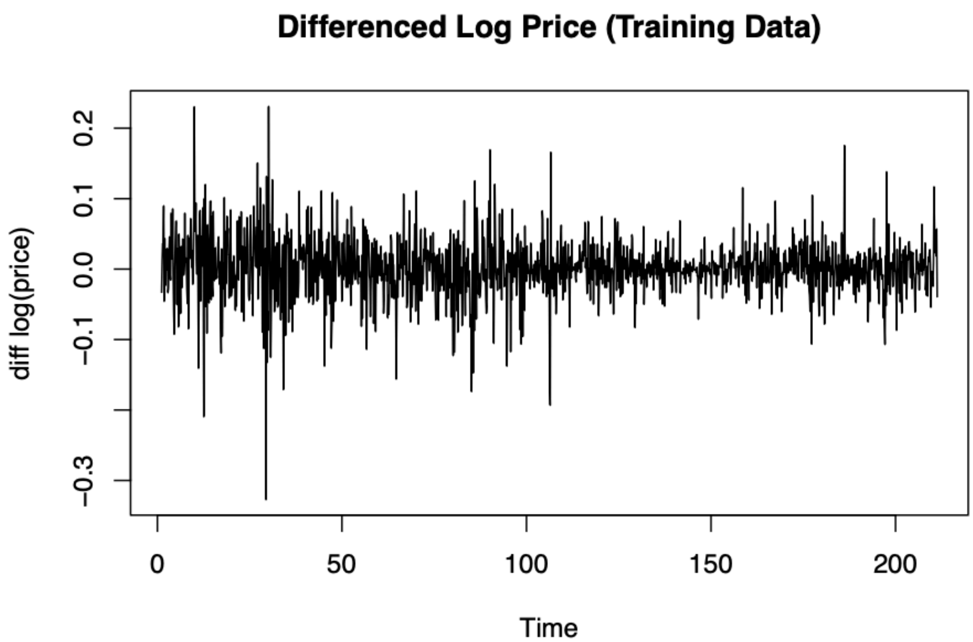

Fig 3. Differenced time series plot.

Analysis A (ARIMA/SARIMA/SARIMAX)

1. ACF and PACF plots

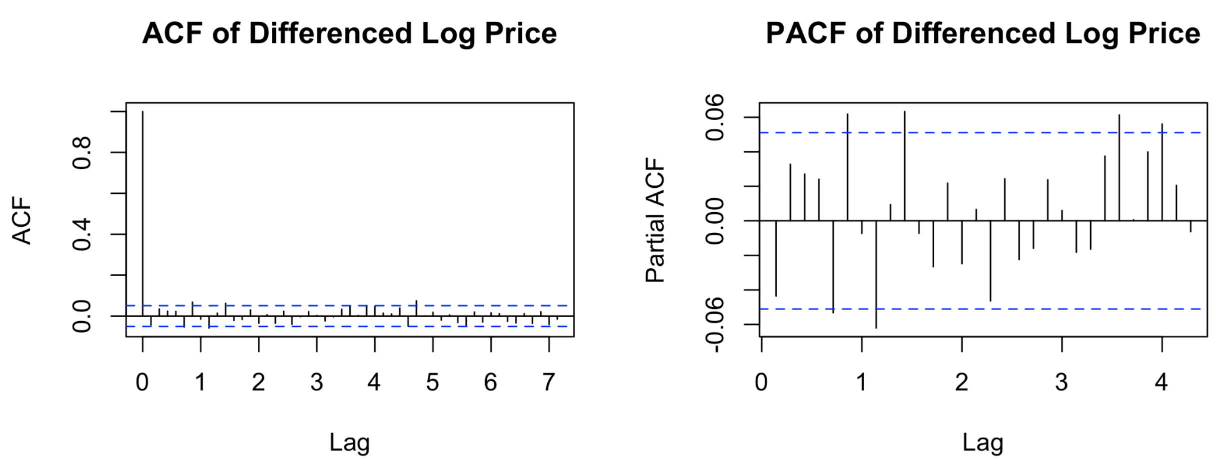

Fig 4. ACF and PACF plot of the differenced time series.

- ACF: all autocorrelations after lag 0 are very small and within 95% bands so strong evidence of weak serial dependence, similar to white noise.
- PACF: small, noisy spikes with no clear cutoff pattern.
- Interpretation: we still fit low and moderate-order ARMA models and use information criteria to decide.

2. ARIMA Modelling and Diagnostics

After turning the ETH closing prices into a stationary series (taking logs, then first differences), I used that differenced log price series to build non-seasonal ARIMA models. I kept the time order and used 80% of the data to train the models and 20% to test them. I tried several ARMA/ARIMA options (like (2,2), (3,3), (4,4), (0,1), (2,1)) and, for each, checked how well it fit using the log-likelihood and AIC. The ARIMA(4,1,4) model gave the best balance of fit and simplicity, so I chose it as my final model.

```
##          model       AIC   logLik
## 4 ARMA(4,0,4) -5138.121 2579.061
## 3 ARMA(3,0,3) -5135.167 2575.583
## 1 ARMA(1,0,1) -5133.148 2570.574
## 5 ARMA(0,0,1) -5132.728 2569.364
## 6 ARMA(2,0,1) -5131.518 2570.759
## 2 ARMA(2,0,2) -5129.870 2570.935
```

Fig 5. Comparison table of ARMA models.

Then I checked whether this model was reasonable by looking at its residuals. The residuals over time looked like random noise, and their ACF plus Ljung-Box tests showed no strong leftover autocorrelation.

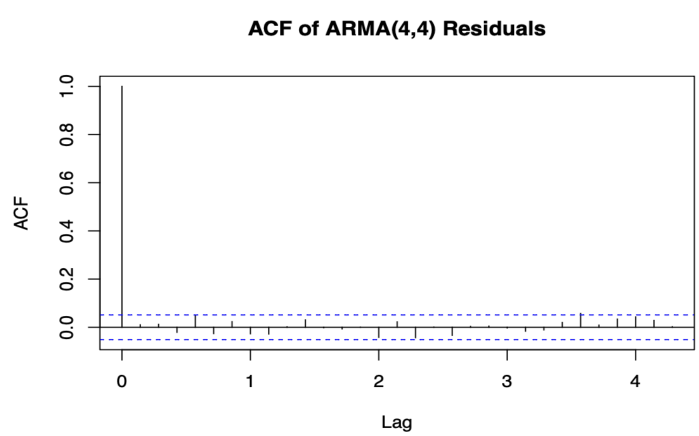

Fig 6. Residual Diagnostic of ARMA model.

Finally, I used the model to make 7-day-ahead forecasts, converted them back to the original price scale, and compared them with actual ETH prices. The RMSE was fairly small, and an

“Actual vs Predicted” plot showed that most real prices fell inside the forecast bands, meaning the short-term forecasts were reasonably accurate.

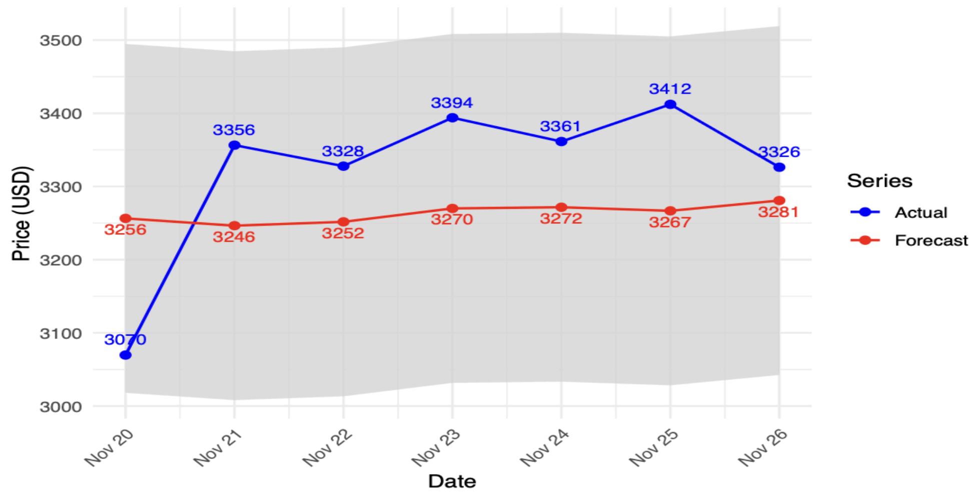

Fig 7. 7-day forecast of ARMA model.

On the basis of AIC and LLR test, ARIMA(4,1,4) is chosen as the best ARIMA model. The following graph is the result of a 7-day forecast on the chosen model which gave a **MSE of 119.439 USD**.

3. SARIMA Modelling and Diagnostics (Week Seasonality)

Since crypto markets often move in weekly cycles, I extended the ARIMA model to a seasonal ARIMA (SARIMA) with a 7-day period. I again used the differenced log prices, kept the original time order, and split the data into training and test sets. I tried several SARIMA models (starting from smaller ones like SARIMA(1,1,1)*(1,0,1)7 and, for each, looked at the log-likelihood and AIC. The SARIMA(5,1,5)*(1,0,1)7 models had the best scores, so I chose it as my final seasonal model.

```
##                          model   logLik       AIC
## 1 SARIMA(1,1,1)x(1,0,1)[7] 2569.813 -5129.626
## 2 SARIMA(2,1,2)x(1,0,1)[7] 2578.465 -5142.930
## 3 SARIMA(3,1,3)x(1,0,1)[7] 2579.505 -5141.009
## 4 SARIMA(5,1,5)x(1,0,1)[7] 2585.075 -5144.151
```

Fig 8. Comparison table of SARIMA models.

I then checked its residuals. The residual plots didn’t show any clear pattern, and their ACF plus Ljung-Box tests suggested no remaining autocorrelation.

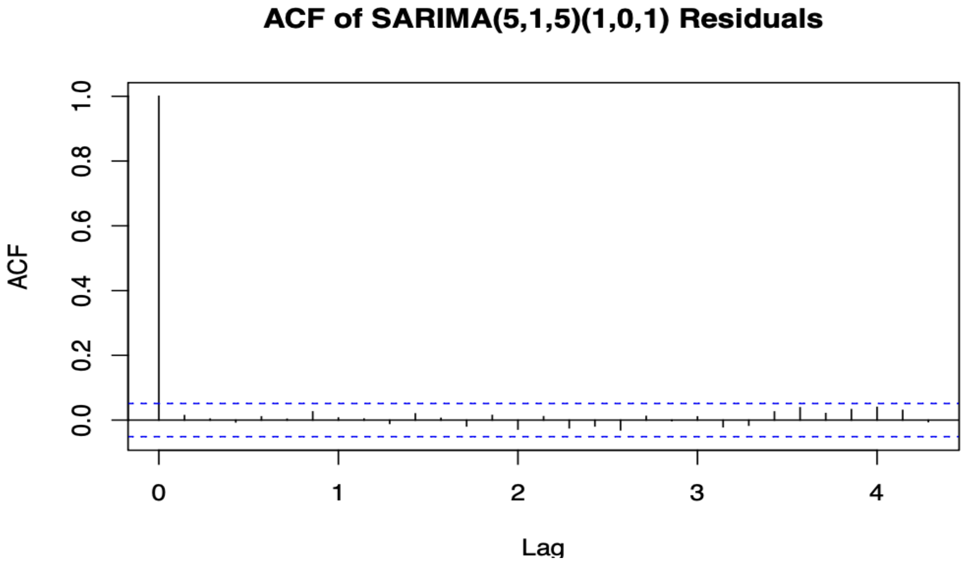

Fig 9. Residual Diagnostics of SARIMA model.

Finally, I used this SARIMA model to make 7-day-ahead forecasts and compared them with the actual ETH prices. I reported the RMSE and plotted “Actual vs Predicted” prices with a 95% prediction band, which showed that the seasonal model tracks weekly movements in Ethereum prices reasonably well.

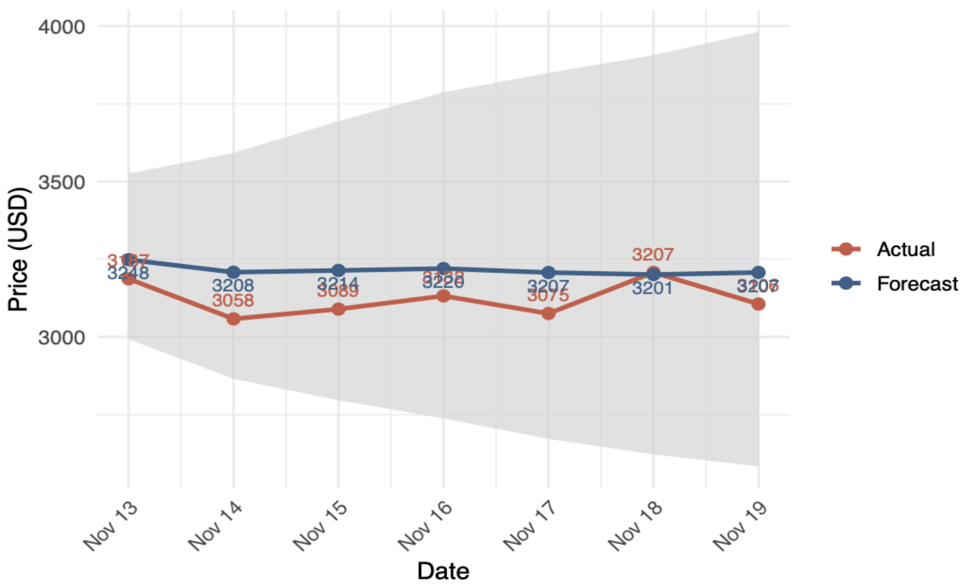

Fig 10. 7-day forecast of SARIMA model.

The SARIMA model gave a **MSE of 105.015 USD** which is slightly better than the ARIMA MSE. The following graph is the forecast result of our chosen SARIMA model. On the basis of AIC and LLR test, SARIMA(5,1,5)*(1,0,1) is chosen as the best model.

4. SARIMAX

To see if adding extra information could improve forecasts, I extended the best SARIMA model to a SARIMAX by including yesterday's log-price as an additional regressor. The chosen specification was SARIMAX(5,1,5) × (1,0,1)7, which still captures both the weekly seasonality and short-term dynamics of Ethereum prices. The coefficient on the lag-1 log price is positive but not strongly significant, and the estimation produces a mild convergence warning, suggesting that this extra regressor does not bring much new signal beyond what the SARIMA structure already uses. The resulting accuracy is RMSE ~ 0.25 on the log scale (about $675 on the price scale), which is very similar to but not clearly better than the SARIMA-only model.

5. Rolling Window Evaluation

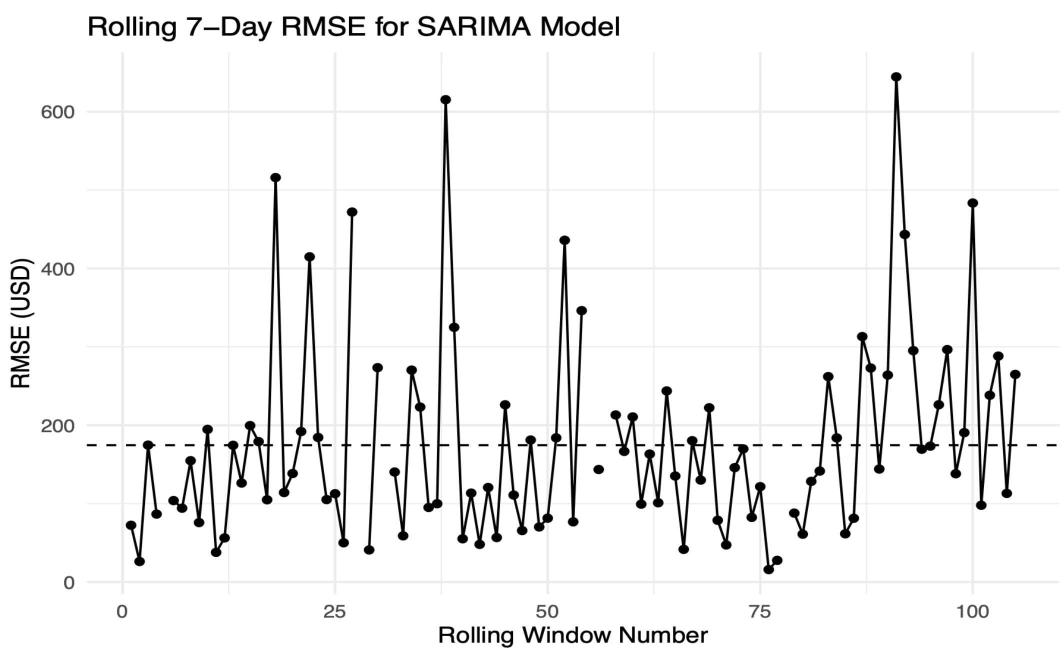

Fig 11. Rolling window evaluation of SARIMA model.

This plot shows how the 7-day forecast error (RMSE) for the SARIMA model changes over time as we roll the window forward. Most windows have errors around $150-$200, but there are several windows where the RMSE suddenly shoots up to $400-$600. The forecast accuracy is not stable: during calm periods the model does fine, but during turbulent periods the errors explode. These high swings suggest volatility clustering and that the size of the errors is changing over time. This kind of time-varying variance is exactly what GARCH models are designed to capture, which is why this graph motivates adding a GARCH model.

Analysis B (GARCH)

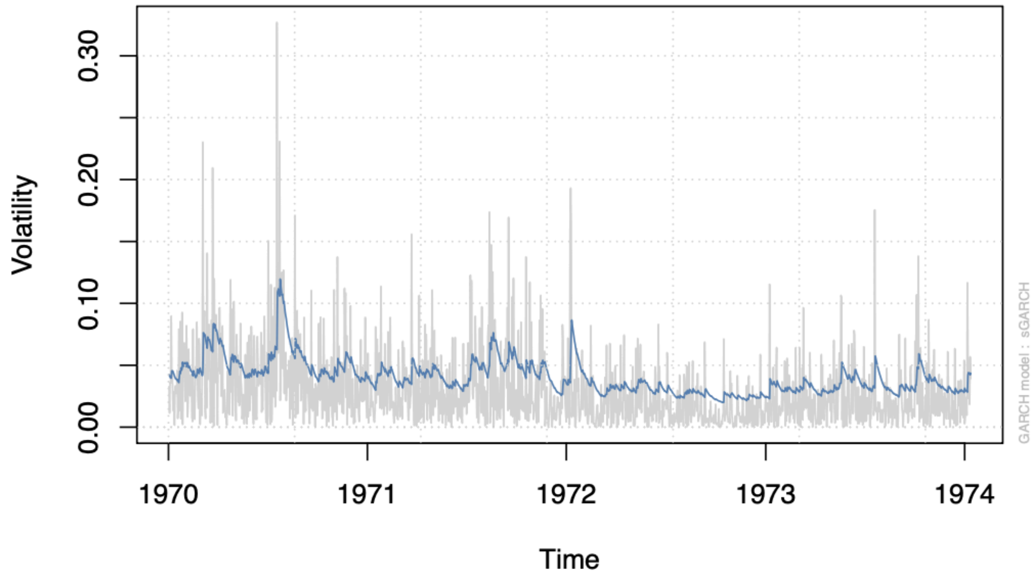

Fig 12. GARCH model.

- The grey spikes show daily return shocks, the blue line is the GARCH-estimated conditional volatility over time.

- Volatility comes in clusters: long stretches of calm periods are interrupted by bursts of very high volatility, rather than being constant.

- After large spikes, the blue line decays slowly instead of dropping immediately, showing that shocks to volatility are persistent.

- Overall, the plot confirms that a GARCH-type model is appropriate for this series because the variance clearly changes over time instead of staying flat.

To model the changing risk in Ethereum prices, I fitted a basic GARCH(1,1) model to the daily log-returns. The volatility parameters (α_1 and β_1) are highly significant and their sum is close to 1, which means volatility shocks fade slowly, periods of high risk tend to persist. The model passes the main residual diagnostics, so it captures volatility clustering reasonably well. However, its forecast error is about 0.43 on the log scale (~ $1,330 in price), so it mainly helps quantify risk and does not improve point price forecasts compared to the ARIMA/SARIMA models.

Results

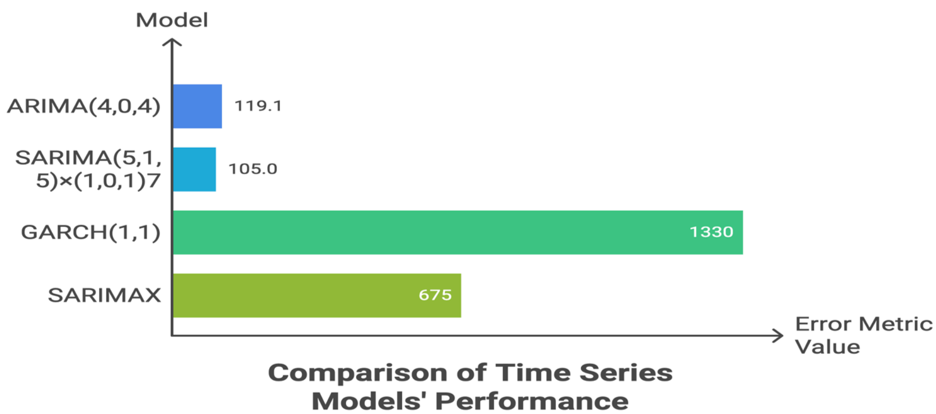

Fig 13. Root Mean Squared Error of all the models.

- Best forecasting model: SARIMA(5,1,5)×(1,0,1)[7] on log-prices, with the lowest test error (RMSE ~ 105 USD) and well-behaved residuals.

- ARIMA(4,0,4) also performs reasonably (RMSE ~ 119 USD), but is consistently weaker than the seasonal SARIMA, confirming the value of modelling weekly patterns.

- Conducted a log-likelihood ratio test to compare both the models. Since the AIC was lower and p-value for the log-likelihood ratio was less than 0.05, adding weekly seasonality (the SARIMA model) gave a significantly better fit than plain ARIMA.

- SARIMAX (adding lag-1 log-price as an exogenous regressor) gives RMSE ~ 0.25 on log scale (~675 USD on price), which is similar to but not better than SARIMA, so the extra regressor does not add much predictive power.

- GARCH(1,1) captures volatility dynamics well (highly persistent $\alpha_1+\beta_1 \sim 0.99$) but gives a much larger price error (RMSE ~ 0.43 log, ~1330 USD), so it is useful for risk/volatility modelling rather than for improving point forecasts.

- Overall, for short-term Ethereum price prediction, a pure SARIMA model on log-prices is the most effective and parsimonious choice among all models tried.

Discussion

Limitations

1. **Data Scope:** Single asset (Ethereum USD) from one source and one time period; results may not generalize to other cryptos, exchanges, or market regimes.

2. **Feature Choice:** Models use only past prices (and a simple lagged regressor for SARIMAX); no volume, on-chain metrics, or news/sentiment, so many drivers of price are ignored.

3. **Model Class:** All models are linear (ARIMA/SARIMA/SARIMAX, GARCH). Sudden jumps and non-linear effects typical in crypto are only partially captured, which limits forecast accuracy.

4. **Evaluation Horizon:** Forecasts are evaluated mainly on short 7-day windows; performance for longer horizons or under extreme stress periods is not fully tested.

Conclusion

- After log-transforming and differencing, Ethereum prices were modeled with ARIMA, SARIMA, SARIMAX, and GARCH in a consistent forecasting pipeline.

- SARIMA(5,1,5)×(1,0,1)7 gave the best price forecasts (lowest RMSE/MSE), slightly improving on the non-seasonal ARIMA(4,0,4).

- GARCH(1,1) captured volatility clustering well but did not improve point price forecasts compared to ARIMA/SARIMA.

- SARIMAX with a simple lag-1 exogenous term performed similarly to SARIMA, suggesting that richer external features are needed to add value.

- Overall, classical time-series models provide a solid baseline for short-term ETH forecasting, but future work should include stronger exogenous signals and non-linear models to better handle large, news-driven market moves.