University of Padova
Department of Mathematics

Project: Netflix Forecasting

(SCP7079231) Business Economic and Financial Data

Group Members: Max Hans-Jürgen Henry Horn
Elio Shyti
Lennart Niels Bredthauer

Supervisor: Prof. Dr. Mariangela Guidolin

Date: July 2025

Table of Contents

1 Introduction

Forecasting stock prices is a critical problem in financial analysis. Accurate short-term forecasting can provide a competitive edge in fast-moving markets. However, stock prices are notoriously noisy, non-stationary, and influenced by a complex interplay of internal dynamics and exogenous influences.

Netflix Inc. (NFLX) serves as a particularly relevant case study due to its high volatility, sensitivity to market sentiment, and strong competitive dynamics within the streaming industry. As a leading technology stock, its price behavior is affected not only by internal performance but also by the actions of competitors such as Disney, Amazon, and Warner Bros. Discovery. This makes Netflix a compelling candidate for testing forecasting methods that aim to incorporate both endogenous patterns and exogenous influences.

This project aims to evaluate and compare the forecasting performance of two distinct modeling paradigms: traditional time series models and machine learning approaches. Specifically, we analyze the daily adjusted closing prices of Netflix Inc. (NFLX) using ARIMA and ARIMAX models, as well as Gradient Boosting Regression (GBR) models trained with various loss functions (Gaussian, Laplace, and Pseudo-Huber). We seek to identify which approach yields the most accurate and robust short-term forecasts.

The ARIMA and ARIMAX models are well suited for capturing linear temporal dependencies. On the other hand, GBR is a powerful ensemble learning technique capable of modeling nonlinear relationships and interactions, especially when enhanced with rich feature engineering and external inputs.

Throughout the analysis, we apply consistent rolling forecast strategies to mimic real-world forecasting scenarios. We assess model performance using multiple metrics, including MAE, RMSE, and MAPE, and examine the accuracy of forecasts and the ability to capture uncertainty. The overarching goal is not only to compare model accuracy but also to highlight the trade-offs in complexity, interpretability, and robustness between traditional and machine learning forecasting models.

2 Data Description

The dataset used in this analysis consists of historical daily stock price data for Netflix Inc. (NFLX), obtained from *Yahoo Finance* using the `quantmod` package in R [1]. To ensure reproducibility, the data was exported and saved as a CSV file. The dataset includes seven variables for each trading day: Date, Open, High, Low, Close, Adjusted Close and Volume.

The data spans the period from June 9, 2020 to June 9, 2025, including 1,256 observations corresponding to trading days within that range.

For the purpose of forecasting, we focus exclusively on the adjusted close price. Unlike the raw closing price, the adjusted closing price accounts for corporate actions such as stock splits and dividend payments. This adjustment provides a more accurate reflection of the stock's value over time and avoids artificial jumps in the price series, which is particularly important for time series modeling [2].

A preview of the first six rows of the dataset is shown in Table 3. The stock price over the five-year period ranged from a minimum of \$166.40 to a maximum of \$1,058.60, with a mean of \$504.70 and a median of \$493.60. This wide spread in values reflects the inherent volatility typical of tech stocks and suggests the presence of non-constant variance, which has implications for modeling.

As a preliminary check, the dataset was examined for missing values using the `is.na()` function in R. No missing entries were found in any of the columns, indicating that the time series is both complete and continuous, and therefore suitable for analysis without requiring further data cleaning.

3 Exploratory Time Series Analysis

To gain a deeper understanding of the structure and dynamics of Netflix's stock performance, we conducted a series of exploratory analyses focused on trend, volatility, and stationarity. These insights are essential for selecting and configuring appropriate forecasting models.

3.1 Trend and Volatility Inspection

Figure 1: Netflix adjusted Closing Price (June 2020 – June 2025)

The underlying time series of Netflix's adjusted closing prices exhibits an overall upward trend, particularly strong from mid-2023 onward. However, the data also reveals substantial volatility, including a sharp decline in early 2022 followed by a rapid recovery. These fluctuations are characteristic of tech stocks and suggest the presence of non-stationarity.

3.2 Rolling Statistics

To further explore the time-varying nature of the data, we applied rolling statistics using a 21-day (approximately one trading month) moving window. Figure 2 shows the rolling mean and rolling standard deviation. The rolling mean demonstrates clear shifts over time, reinforcing the presence of trends. The rolling standard deviation, scaled by a factor of 10 for visibility, exhibits periods of heightened volatility, indicating that the variance of the time series is not constant.

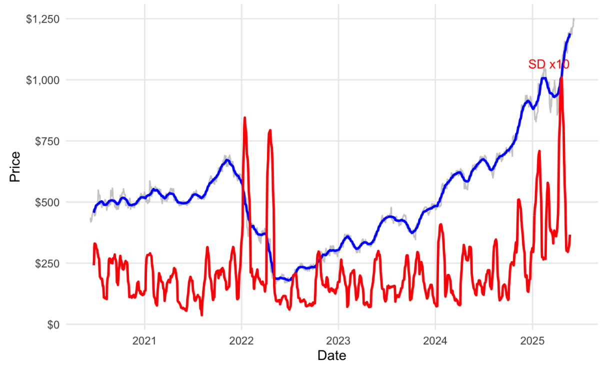


Figure 2: Rolling Mean (blue) and Scaled Standard Deviation (21-day window) in red of Netflix adjusted closing price

Stationarity is a core assumption for many time series models, particularly ARIMA. A stationary series maintains a constant mean and variance over time, enabling more reliable statistical inference. To assess stationarity, we performed both visual inspection and formal hypothesis testing.

We applied the Augmented Dickey-Fuller (ADF) test to the raw adjusted closing price series. The test yielded a Dickey-Fuller statistic of 0.18 with a p-value of 0.99 and a suggested lag order of 10. This high p-value indicates failure to reject the null hypothesis that the series contains a unit root, confirming that the time series is non-stationary.

These results are consistent with existing literature that highlights the challenges of modeling non-stationary financial time series. As noted by Han et al. [3], ARIMA models are sensitive to non-stationarity, necessitating transformations to ensure consistent statistical properties. Similarly, Box et al. [4] emphasize that for many real-world processes, it is often reasonable to assume that "some suitable difference of the process is stationary". This supports our choice to apply differencing as a key preprocessing step.

3.3 STL Decomposition

To better understand the structural components of the series, we applied the Seasonal and Trend decomposition using Loess (STL). This method decomposes the time series into three additive components:

- **Trend:** Captures the long-term direction of the data. A notable dip is observed in early 2022, followed by a consistent upward movement.
- **Seasonality:** Reveals short-term, recurring patterns. While the seasonal amplitude is relatively small, it displays regular oscillations that may reflect market cycles or micro-seasonal effects.
- **Remainder:** Represents irregular noise and residual volatility. The presence of large spikes in this component highlights market shocks and short-term anomalies not explained by trend or seasonality.

The STL decomposition supports the conclusion that the series is non-stationary, with a dynamic trend and evolving seasonal behavior.

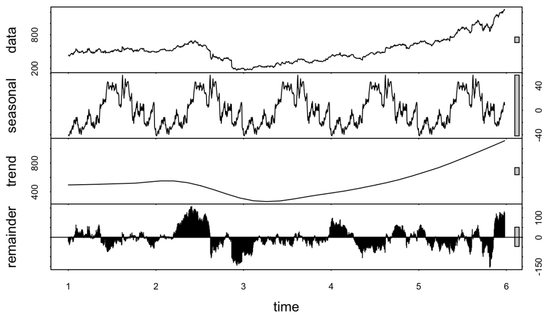


Figure 3: STL decomposition of Netflix adjusted closing price

Economic Interpretation of Price Behavior

The observed volatility in Netflix's stock price over the 2020–2025 period is closely linked to major economic and company-specific events. One of the most prominent price movements occurred in April 2022, when Netflix's stock dropped by over 35% in a single day. This sharp decline followed the announcement of a net loss of 200,000 subscribers in quarter one. It was the companies first subscriber loss in over a decade. A large part of this loss, around 700.000 subscribers, came from Netflix's decision to suspend its services in Russia following the country's invasion of Ukraine. In addition, the company projected it would lose another 2 million subscribers in the next quarter, which increased investor concerns. These announcements, along with rising competition in the streaming market and broader economic pressures like inflation and interest rate hikes, led to a major sell-off that erased tens of billions of dollars in market value [5].

After the sharp decline in 2022, Netflix rebounded strongly, driven by two key strategic shifts. First, the company introduced a lower-priced, ad-supported subscription tier, expanding its market reach and creating a new revenue stream. Second, it implemented stricter rules on password sharing, which led to a surge in paid subscriptions. According to the Financial Times, these changes contributed to Netflix adding over 45 million subscribers globally and regaining investors confidence. By mid-2024, its stock had nearly tripled from its post-crash levels, underscoring the success of its commercial strategy [6].

4 Forecasting Strategy

4.1 Setup

All models forecast the same target: the daily adjusted closing price of Netflix Inc. (NFLX). To ensure consistency, we implemented a rolling window forecasting procedure across all models. Each model was trained on a 252-day window of past data (approximately one year of trading days) and used to generate a 5-day-ahead forecast. Forecasts were produced in non-overlapping 5-day increments to simulate a weekly investment decision cycle. This setup ensured identical forecast horizons, evaluation windows, and temporal alignment across all models.

4.2 Motivation

The rolling window approach was chosen to mimic real-world forecasting conditions, where models must rely only on past data without access to future values. This prevents lookahead bias and reflects the constraints faced by financial analysts. It also allows us to evaluate how each model performs over time, particularly during volatile periods or market shocks.

Rolling-origin (or rolling-window) evaluation is recommended in the literature as the most robust out-of-sample validation method under non-stationarity [7].

4.3 Evaluation Metrics

We assessed forecast performance using three standard metrics:

- **Mean Absolute Error (MAE)**: measures the average magnitude of forecast errors, regardless of direction.
- **Root Mean Squared Error (RMSE)**: penalizes larger errors more heavily, giving insight into the variability of forecast accuracy.
- **Mean Absolute Percentage Error (MAPE)**: expresses error as a percentage of the actual value, allowing comparability across time periods and price levels.

These metrics are commonly used in financial time series evaluation.

4.4 Forecast Window

We chose a 252-day training window because it represents roughly one full year of trading days, which is a common choice in financial forecasting. This length provides a good balance. It captures enough recent market behavior to stay responsive, while still including enough data to make the models stable. Using a one-year window is also in line with how many financial analysts and institutional investors approach forecasting, as shown in [8].

5 Methodology - ARIMA and ARIMAX Models

5.1 ARIMA Model

The ARIMA (AutoRegressive Integrated Moving Average) model, originally introduced by Box and Jenkins in the 1970s [4], is a popular tool for time series forecasting. It captures the underlying structure of a univariate time series through three components:

- **Autoregression (AR):** Incorporates the dependence between an observation and a specified number of its past values.
- **Integration (I):** Refers to the differencing of raw observations to make the time series stationary.
- **Moving Average (MA):** Models the relationship between an observation and the residual errors from a moving average model applied to lagged observations.

An ARIMA model is generally denoted as ARIMA(p, d, q), where p is the order of the autoregressive part, d is the degree of differencing, and q is the order of the moving average part. By appropriately selecting these parameters, ARIMA can effectively model a wide range of time series patterns, including trends and short-term dependencies.

Stationarity and Differencing

To satisfy the stationarity assumption required for ARIMA modeling, we applied first-order differencing to the adjusted closing price time series. The resulting differenced series, shown in Figure 4, appears to fluctuate around a constant mean, suggesting that the trend component has been successfully removed.

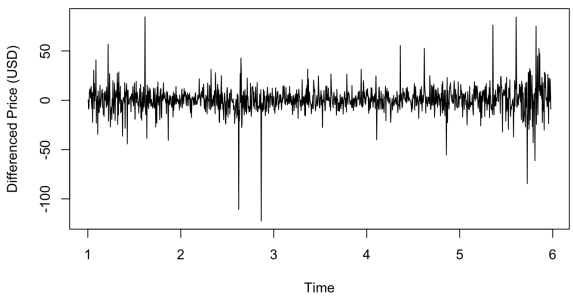


Figure 4: First-Order Differenced Series of Netflix adjusted closing price

To statistically confirm stationarity, we applied the Augmented Dickey-Fuller (ADF) test. The test yielded a Dickey-Fuller statistic of -10.459 with a p-value less than 0.01, allowing us to reject the null hypothesis of a unit root. This confirms that the series is stationary after first-order differencing.

ACF and PACF Analysis

To guide the choice of model parameters, we examined the autocorrelation (ACF) and partial autocorrelation (PACF) plots of the first-differenced series, shown in Figure 5 and Figure 6.

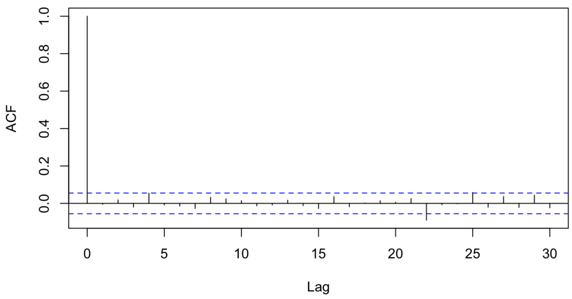


Figure 5: ACF of First-Order Differenced Series

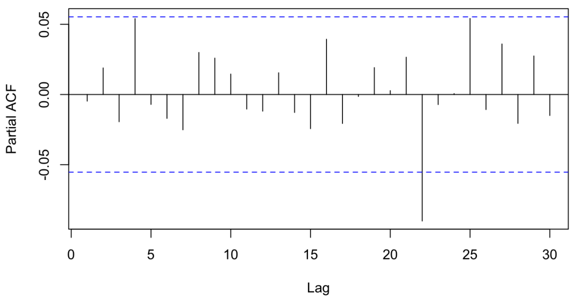


Figure 6: PACF of First-Order Differenced Series

The ACF plot shows a sharp drop after lag 1, which may indicate a short memory process. However, the PACF plot does not display a dominant spike at lag 1, instead, it reveals several small values across a range of lags, all within the confidence bounds, with the exception of a spike exceeding the confidence bound at lag 22. This suggests a weak autoregressive structure, potentially supporting the hypothesis that an AR component may not be necessary (i.e., AR = 0).

Model Selection

To identify the most suitable ARIMA specification, we compared several candidate models using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Table 1 summarizes the AIC and BIC values for each model.

Table 1: Model Comparison Based on AIC and BIC

Model	**AIC**	**BIC**
ARIMA(0,1,0)	**10076.02**	**10081.15**
ARIMA(1,1,0)	10078.01	10088.28
ARIMA(2,1,0)	10079.44	10094.85
ARIMA(1,1,1)	10080.01	10095.42
ARIMA(2,1,1)	10080.01	10095.42
ARIMA(4,1,0)	10079.11	10104.78
ARIMA(5,2,0)	10261.35	10292.16

Among all models, the ARIMA(0,1,0) specification achieved the lowest AIC and BIC, indicating the best in-sample fit with the fewest parameters (only one). This model implies that the time series follows a *random walk*, meaning that each observation is essentially the sum of the previous value and a random shock. In mathematical terms:

$$Y_t = Y_{t-1} + \varepsilon_t, \tag{1}$$

where ε_t denotes white noise. This implies that the best forecast for tomorrow's price is today's price, possibly plus a constant drift.

The suitability of ARIMA(0,1,0) aligns with findings from the ACF and PACF plots, which showed no strong autocorrelation at early lags, and with the ADF test, which confirmed non-stationarity in the original series. Financial time series, particularly stock prices, often exhibit this behavior due to market efficiency, where all known information is already priced in, and only new (random) information drives future price movements.

While slightly more complex models such as ARIMA(1,1,0) and ARIMA(2,1,0) also performed well, their marginal improvements in fit were offset by increased complexity. Additionally, the ARIMA(5,2,0) model suggested by `auto.arima()` was rejected, as second-order differencing was found unnecessary given our earlier stationarity results.

In conclusion, ARIMA(0,1,0) was selected for its empirical performance, and theoretical consistency with the behavior of asset prices in efficient financial markets.

Residual Diagnostics

After fitting the ARIMA(0,1,0) model, we conducted a residual diagnostic check to assess whether the model adequately captured the underlying structure of the data. This step is crucial to ensure that the residuals behave like white noise, which is a fundamental assumption for a well-specified time series model.

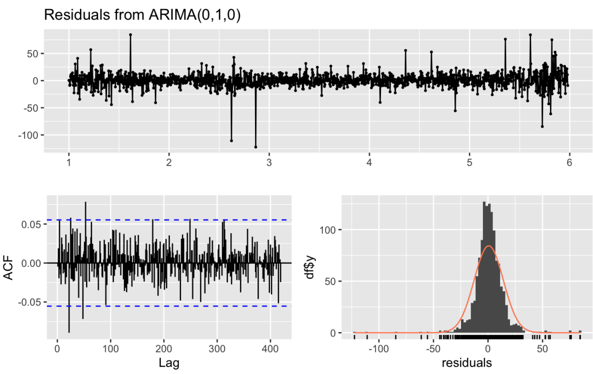


Figure 7: Residual Diagnostics of ARIMA(0,1,0) Model

Figure 7 displays the residual diagnostics plot for ARIMA(0,1,0), including the standardized residuals, the autocorrelation function (ACF) of the residuals, and a histogram with a fitted normal density curve.

- The residual time series shows no discernible pattern, suggesting that autocorrelation has been sufficiently removed.
- The ACF plot confirms the absence of significant autocorrelations at any lag, supporting the assumption that the residuals are uncorrelated.
- The histogram of residuals resembles a normal distribution, with most residuals concentrated around zero.

To formally test the white noise assumption, we applied the Ljung-Box test to the residuals. The test yielded a Q-statistic of 189.27 with 251 degrees of freedom and a p-value of 0.9986. Since the p-value is well above 0.05, we fail to reject the null hypothesis that the residuals are independently distributed. This result further supports the adequacy of the model.

Taken together, these diagnostics confirm that the ARIMA(0,1,0) model residuals behave like white noise. This suggests that the model has effectively captured all relevant structure in the differenced time series and that no further AR or MA components are needed, consistent with the interpretation of ARIMA(0,1,0) as a random walk process [4].

Forecasting and Evaluation

Forecasting was performed using a rolling window strategy as described in Section 4. The ARIMA(0,1,0) model achieved a mean absolute error (MAE) of 15.97, a root mean squared error (RMSE) of 23.69, and a mean absolute percentage error (MAPE) of 3.43%. These results suggest that, despite its simplicity, the model captures short-term trends reasonably well in a noisy and volatile financial environment.

To further examine the limitations of the ARIMA(0,1,0) model, we extended the forecasting exercise by varying the forecast horizon from 1 to 21 trading days. At each step, a 252-day rolling training window was used, and forecasts were generated for the corresponding horizon.

The plot in Figure 8 shows the resulting Mean Absolute Error (MAE) values across horizons. As expected, the MAE increases with longer forecast horizons, reflecting the compounding uncertainty in predicting further into the future. The curve remains relatively smooth, with only minor fluctuations, suggesting the model maintains stable performance across short- to mid-range horizons before accuracy gradually deteriorates. This trend highlights the inherent trade-off between forecast length and reliability in time series modeling.

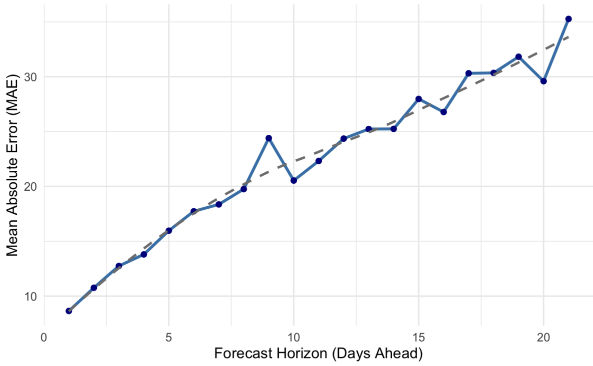


Figure 8: Mean Absolute Error (MAE) across Forecast Horizons (1–21 Days) for ARIMA(0,1,0) Model

5.2 ARIMAX with Exogenous Variables

Model Setup

The ARIMAX (AutoRegressive Integrated Moving Average with eXogenous variables) model is an extension of the traditional ARIMA model that incorporates external predictors (exogenous variables) [9]. While ARIMA models rely solely on the historical values and past errors of the target time series, ARIMAX also includes the influence of other related time series or factors that may help explain or predict the behavior of the main series. Theoretically, ARIMAX retains the ARIMA structure but adds a regression component for external variables, allowing the model to account for influences beyond the target series' own past values. To build the ARIMAX model for predicting the stock price of Netflix (NFLX), historical data for Disney (DIS), Amazon (AMZN) and Warner Bros. Discovery (WBD) were also used to serve as exogenous variables. The reason why we included these variables is because stocks within the same industry or sector often move together since they are affected by similar economic forces, regulatory changes or consumer demand [10]. For Netflix, competitors in the streaming or entertainment sector like Disney, Amazon, or Warner Bros. Discovery could impact market sentiment towards the whole industry. We imported the last 5 years of adjusted closing prices for these stocks. This extension makes ARIMAX especially useful in financial settings where prices are shaped by market-wide developments, not just internal company trends.

A key limitation of the ARIMAX model when used for forecasting is its dependence on the availability of the exogenous regressors (the ”X” variables) at the forecast time points. Since the model incorporates these external predictors to improve accuracy, making forecasts requires knowing the future values of these regressors. If the exogenous variables are not available at time t (the forecast horizon), it becomes impossible to generate forecasts from the ARIMAX model. To address this problem, we used lagged values of order 1 of the exogenous variables as regressors (time $t-1$).

Comparing model fit metrics with the ARIMA(0,1,0) model (without exogenous variables), the ARIMAX model demonstrated slightly better predictive performance. The mean absolute error (MAE) decreased from 15.97 to 14.37, the root mean squared error (RMSE) dropped from 23.69 to 22.21, and the mean absolute percentage error (MAPE) improved from 3.43% to 3.02%. Including competitors' stock prices improved accuracy marginally, as shown by lower MAE, RMSE, and MAPE values.

Residual Diagnostics

The Ljung-Box test performed on the residuals gives us a p-value of 0.3283 suggesting that the residuals behave like white noise, indicating that the model has adequately captured the temporal dependencies. We also show the distribution of the residuals in the figure 9.

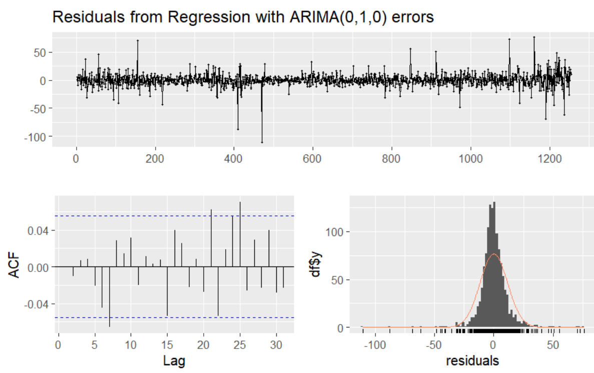


Figure 9: Residual Diagnostics of ARIMAX Model

Forecasting Results

The following plot compares the 5-day rolling window forecasts of NFLX stock prices using ARIMAX model with the actual observed values over time. The forecasted values (in red) closely follow the trend and fluctuations of the actual data (in blue), indicating that the model is capturing the underlying dynamics of the stock price effectively.

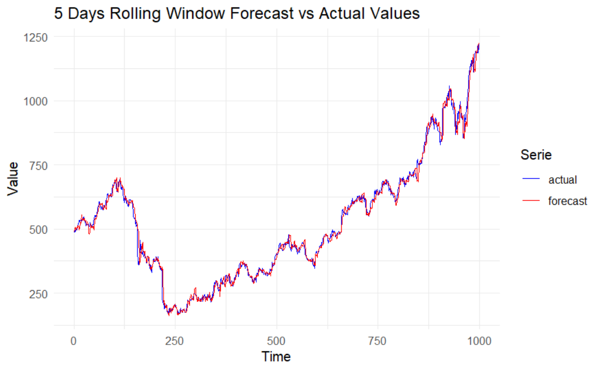


Figure 10: 5 Days Rolling Window Forecast For NFLX

This plot highlights the model's ability to adapt to both upward and downward movements in the stock price, suggesting that the ARIMAX approach can be a valuable tool for short-term forecasting.

6 Methodology - Gradient Boosting Regression

As the ARIMA in 5.1 and ARIMAX in 5.2 models revealed that the Netflix stock price behaves similarly to a random walk, we explored more flexible, non-linear approaches to improve forecast accuracy. Gradient Boosting Regression (GBR), an ensemble machine learning method, is well-suited for this task as it can model complex interactions between lagged values, technical indicators, and external market factors. Furthermore, we introduced a Gaussian [11], Pseudo-Huber [12] and Laplace [11] loss function, to evaluate over the models to achieve maximum accuracy.

6.1 GBR Model

Gradient Boosting builds an ensemble of sequential decision trees, where each subsequent tree is trained to correct the residual errors of the previous iteration. This approach enables the model to capture complex, nonlinear patterns in financial time series data [13]. The set up of the model is the same as in the ARIMAX model 5.2, but has differences in feature engineering and forecasting the data, which will be discussed in the following sections.

6.2 Feature Engineering

To enhance the predictive power of the Gradient Boosting Regression (GBR) model, a comprehensive set of features that capture both temporal dependencies and cross-market dynamics was created.

First, we introduced lagged values (1–3 day lags) for the *adjusted closing prices* of Netflix (NFLX) itself and of three key competitors: Amazon (AMZN), Disney (DIS), and Warner Bros. Discovery (WBD) [1]. In addition, we incorporated technical indicators commonly used in financial time series forecasting:

- **SMA** (Simple Moving Average, 5 days) – captures short-term price trends and momentum.
- **RSI** (Relative Strength Index, 14 days) – measures the speed and magnitude of recent price changes, helping to identify potential overbought or oversold conditions.
- **Volatility** – computed as the rolling 5-day standard deviation of Netflix's adjusted closing prices, to represent recent market uncertainty.

Finally, we transformed the target variable into daily log returns [13], a common practice in modeling financial time series, which stabilizes variance and facilitates learning nonlinear relationships:

$$\text{Log Return}_t = \log(P_t) - \log(P_{t-1})$$

The feature engineering process ensures that the model is informed not only by the internal behavior of Netflix stock but also by competitive dynamics and market sentiment. By implementing endogenous and exogenous inputs, the GBR model is better equipped to forecast near-term price movements under realistic, dynamic market conditions.

6.3 Evaluation of GBR Models

We compared three commonly used loss functions in Gradient Boosting Regression (GBR): Gaussian, Pseudo-Huber [12] (implemented via the `xgbm` package), and Laplace [11]. The goal was to identify the loss function that delivers the most accurate forecasts for Netflix's adjusted daily closing prices.

Evaluation over Performance Metrics
A detailed performance comparison of the models across different forecast horizons is provided in Table 4 in the Appendix B. The results show that the Laplace loss consistently achieves the lowest error metrics (MAE, RMSE, MAPE) across all horizons, with particularly strong improvements in short-term forecasts.

Importance of Predictor Variables
In the Laplace-based GBR model, the most important features contributing to predictive performance are shown in Figure 11.

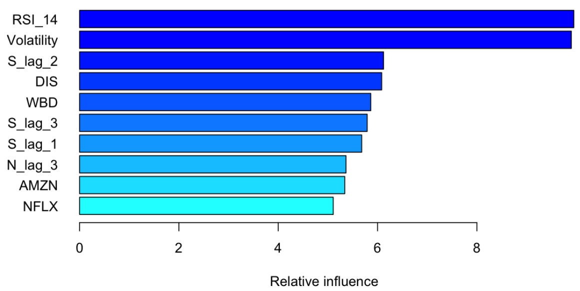


Figure 11: Predictor variable importance spectrum for Laplace GBR model

Figure 11 shows that RSI_14 (approx. 9.95) and Volatility (approx. 9.90) are the most important predictors in the Laplace GBR model, highlighting the strong influence of recent momentum and market uncertainty on Netflix stock forecasts. Among the exogenous variables, the second lag of Disney's stock price (approx. 6.12) and Disney's stock (approx. 6.08) emerge as the most influential external factors.

The features importance for the Gaussian and Pseudo-Huber are not discussed, but can be seen in the appendix C.

Evaluation of Tracking Short-Term Trend
We also compared how the different loss functions affect the model's ability to track short-term trends. The corresponding 5-day ahead forecast trajectories for the Laplace, Pseudo-Huber, and Gaussian GBR models, based on the most recent 30 predictions, are presented in the Figures in Appendix D.

Among the three, the Laplace model provides a slightly more stable and consistent forecast path, closely following the actual price trend and showing somewhat reduced volatility in predictions, as illustrated in Figure 12.

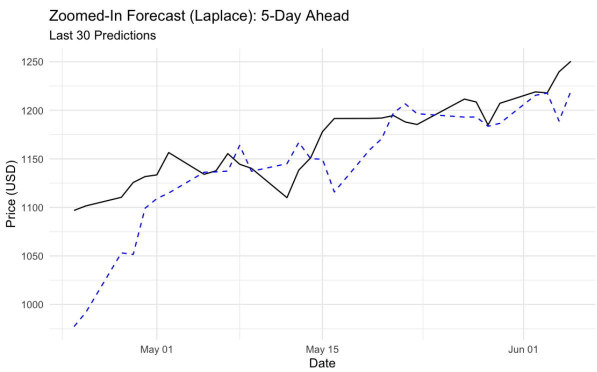


Figure 12: Prediction of the Stock Price using Laplace Loss [11]

The Pseudo-Huber model also performs well, but exhibits slightly larger deviations during periods of stronger volatility. The Gaussian model demonstrates good tracking performance, though it remains somewhat more sensitive to noise and tends to underreact slightly to sudden shifts. This is consistent with its sensitivity to outliers.

Evaluation over Errors
Across all models, the largest forecast errors occur during sharp market movements, both in steep declines (2022) and rapid ascents (2025). As shown in Appendix E, the Gaussian model shows the highest error spikes, while Pseudo-Huber reduces extreme deviations. The Laplace

model achieves the most robust performance, with consistently smaller errors and better handling of volatility.

6.4 Training

Based on the model evaluation results, we focused further analysis on the Laplace model. For comparative purposes, we also generated forecasts using the Gaussian model. We implemented the GBR model using the `gbm()` [13] function in `R`.

The Gradient Boosting model was trained using a rolling window approach with a window size of 252 days, corresponding to approximately one year of trading data. At each iteration, the model was fit on the most recent 252 observations of the dataset, capturing the most up-to-date patterns in the stock price and its engineered features.

The model was trained with the following key hyperparameters:

- `n.trees` = 2500 (number of boosting iterations),
- `shrinkage` = 0.05 (learning rate controlling contribution of each tree),
- `interaction.depth` = 3 (maximum depth of each tree, controlling model complexity).

The hyperparameter `n.trees` is normally set to 5000 [13], since the results showed signals of overfitting a reduction to 2500 was implemented.

The separate models were constructed for forecast horizons ranging from 1 to 5 days ahead. For each horizon h, a dedicated rolling procedure was implemented. Specifically, at each iteration, the model was trained on the slice $[i - 251, i]$ of the dataset and then used to predict the log return for day $i + 1$. This predicted log return was then transformed into a price forecast by applying the exponential transformation.

To evaluate better in our models, we computed best-case, central, and worst-case forecast scenarios by constructing confidence intervals based on the empirical spread of model residuals [14], under both Laplace and Gaussian assumptions. Specifically, for the Gaussian model, the confidence bounds were calculated using the standard deviation (SD) of the residuals, multiplied by a factor of 2.58 [14], corresponding to a 99% confidence interval. For the Laplace model, the median absolute deviation (MAD) [15] was used as a robust measure of scale, with the appropriate multiplier of 6.64 for a 99% confidence interval. The derivation of this value is provided in Appendix F.

6.5 Forecasting

A 5-day ahead forecasts of the models under Gaussian and Laplace loss functions, incorporating best-case, worst-case, and central scenario bounds, see Figures 13 and 14 are compared.

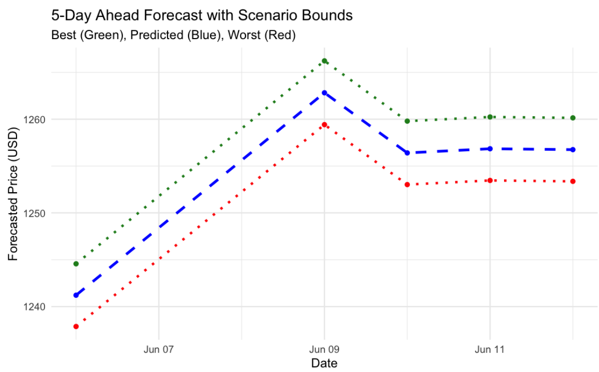


Figure 13: Trend Forecasting for 5 days with GBR and Gaussian Loss Function [11]

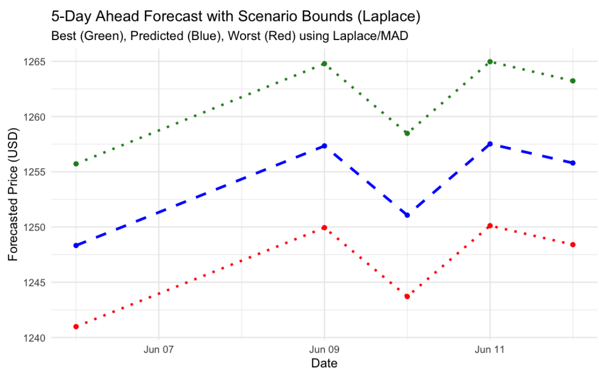


Figure 14: Trend Forecasting for 5 days with GBR and Laplace Loss Function [11]

We dynamically update a three-lag feature structure, shifting existing lag values forward by one position, and inserting each new predicted price as the most recent lag. This process is designed to avoid data leakage by dynamically updating lagged predictors with estimated future values at each horizon. This recursive procedure ensures that each day's forecast relies solely on historically available or previously predicted information.

Both models display a similar trend. We observe an upward movement in the first three days, followed by a slight flattening towards Day 5.

However, important differences appear in the forecast uncertainty intervals. The Gaussian model produces noticeably narrower bands, especially beyond Day 3, suggesting an underestimation of future volatility. This behavior is consistent with the Gaussian distribution's sensitivity to the mean and its inability to capture heavy-tailed noise, which is common in financial time series [15]. As a result, the Gaussian model appears overconfident about the close price range.

In contrast, the Laplace model produces wider and more stable scenario bounds throughout the forecast horizon. The use of the Median Absolute Deviation (MAD) allows the model to remain responsive to local fluctuations and better handle the fat-tailed nature of stock returns [16]. The scenario bands of the Laplace model reflect a more realistic view of future uncertainty, especially during volatile periods.

In summary, while both models agree on directional trends, the Laplace model delivers more reliable risk assessment and uncertainty quantification for financial forecasts, making it better suited for stock price prediction in real-world market environments.

7 Evaluation and Model Comparison

7.1 Evaluation Metrics

We directly compare our results from our model analysis in term of MAE, RMSE and MAPE. The following table provides a consolidated overview of each model's performance metrics.

Table 2: Forecasting Accuracy of Different Models

Model	**MAE**	**RMSE**	**MAPE**
ARIMA (0,1,0)	15.97	23.69	3.43 %
ARIMAX	14.37	22.21	3.02 %
GBR (Laplace)	**12.95**	**18.77**	**2.70%**

Looking at the results, we see that all three models deliver fairly similar forecasting performance, with only moderate differences across MAE, RMSE, and MAPE. Still, the Gradient Boosting model using the Laplace loss stands out as the best overall. It achieves the lowest errors on all three metrics, with an MAE of 12.95, RMSE of 18.77, and MAPE of just 2.70 %. This suggests that combining boosting with the Laplace loss function, which is more robust to outliers, offers practical benefits over standard ARIMA and ARIMAX approaches in our analysis of Netflix stock data.

7.2 Results

Clearly, the Gradient Boosting Regression model with the Laplace loss function represents the strongest performance overall. In fact, every loss function tested within the GBR framework, as shown in Table 4 in appendix B, outperformed both the ARIMA and ARIMAX models across all evaluation metrics, highlighting the advantages of this more flexible, non-linear approach in capturing the dynamics of the stock price data.

7.3 Discussions

It was somewhat unexpected to find that the ARIMA model identified in section 5.1 was an ARIMA(0,1,0), essentially modeling the series as a random walk without mean reversion or auto-regressive structure. Because of this, we extended the approach by fitting an ARIMAX model, adding exogenous variables in the form of competitors' stock prices. This step was motivated by the idea that external market movements might contain predictive signals absent in Netflix's own past values. While this improved the model slightly, as seen in Section 5.2, the gains were modest and the underlying series still exhibited random walk characteristics, limiting the explanatory power of these linear time series approaches.

Given these challenges, we turned to a more flexible, non-linear and non-parametric method: Gradient Boosting Regression, discussed in Section 6. This model is better equipped to capture complex interactions and hidden patterns that traditional ARIMA-based models may miss. Its superior performance on all error metrics highlights the potential of machine learning methods in financial forecasting, particularly when the data does not strongly adhere to the assumptions required by classical time series models.

Limitations
Despite being the best-performing approach in our analysis (see Section 7.2), the Gradient Boosting model has also drawbacks. One important limitation is that in multi-day forecasting, we update lagged features using previous predictions, which means that forecast errors can be propagated over time. This recursive strategy, which is necessary for generating realistic multi-step predictions, is inherently carrying the risk of error propagation. In addition, the model's interpretability is lower compared to simpler time series methods. Furthermore, the reliance on a relatively short historical window for feature engineering may not fully capture longer-term seasonal effects. Future work could consider hybrid models that integrate machine learning with domain-specific time series structures to address these issues.

8 Conclusion

In this project, we compared classical time series models with modern machine learning approaches to forecast Netflix's stock price. While the ARIMA and ARIMAX models provided a reasonable baseline, they were limited by the series' random walk characteristics and small improvements from exogenous inputs. In contrast, the Gradient Boosting Regression model, especially under the Laplace loss function, achieved the best overall performance by capturing complex, non-linear patterns. This result illustrates the challenge of balancing forecast accuracy with model transparency, a key issue in financial modeling.

9 References

References

[1] Yahoo! Finance. Yahoo! finance – financial data, netflix, amazon, disney and warner brothers studios, 2025. Accessed via the quantmod R package.

[2] James Felton and Pawan Jain. True returns: Adjusting stock prices for cash dividends and stock splits. *Advances in Financial Education*, 17(Summer):192–205, 2019. Originally posted May 10, 2018. Available at https://ssrn.com/abstract=3354728.

[3] H. Han, Z. Liu, M. Barrios Barrios, et al. Time series forecasting model for non-stationary series pattern extraction using deep learning and garch modeling. *Journal of Cloud Computing*, 13(2), 2024.

[4] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Hoboken, New Jersey, 5 edition, 2015.

[5] The Washington Post. Netflix loses subscribers for first time in over a decade, 2022. Accessed: 2025-07-02.

[6] Andrew Edgecliffe-Johnson and James Fontanella-Khan. How netflix won the streaming wars. *Financial Times*, 2024. Accessed June 2025.

[7] Leonard J. Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, 2000.

[8] Ines Wilms, Sarah Gelper, and Christophe Croux. Multivariate volatility forecasts for stock market indices: How essential is dimension reduction? *International Journal of Forecasting*, 37(3):1055–1074, 2021.

[9] Billy M Williams. Multivariate vehicular traffic flow prediction: evaluation of arimax modeling. *Transportation Research Record*, 1776(1):194–200, 2001.

[10] Eric De Bodt, B Espen Eckbo, and Richard Roll. Corporate rivalry and return comovement. *Tuck School of Business Working Paper (3218544)*, 2020.

[11] Jerome Friedman Trevor Hastie, Robert Tibshirani. *The Elements of Statistical Learning*. Springer, New York, 2nd edition, 2006.

[12] Kaan Gokcesu and Hakan Gokcesu. Generalized huber loss for robust learning and its efficient minimization for a robust statistics, 2021.

[13] Trevor Hastie Robert Tibshirani Gareth James, Daniela Witten. *An Introduction to Statistical Learning*. Springer, New York, 2013.

[14] George Athanasopoulos Rob J Hyndman. *Forecasting: Principles and Practice*. Springer, Australia, 2018.

[15] Samuel Kotz, Tomasz J. Kozubowski, and Krzysztof Podgórski. *The Laplace Distribution and Generalizations: A Revisit with New Applications*. Springer, 2001.

[16] Enrico Scalasc Cheoljun Eoma, Taisei Kaizojib. Fat tails in financial return distributions revisited: Evidence from the korean stock market. page 2.

10 Appendix

A Sampled Data of the Netflix Stock Dataset

Table 3: Sample of the Netflix stock dataset showing daily trading details, including opening, high, low, and closing prices, trading volume, and adjusted close values for a subset of dates.

Date	**Open**	**High**	**Low**	**Close**	**Volume**	**Adjusted**
2020-06-09	421.65	434.73	420.31	434.05	6,797,000	434.05
2020-06-10	436.00	439.69	430.55	434.48	4,896,900	434.48
2020-06-11	428.20	445.57	424.16	425.56	7,462,900	425.56
2020-06-12	429.00	434.06	412.45	418.07	6,461,100	418.07
2020-06-15	421.40	426.49	415.42	425.50	4,467,900	425.50
2020-06-16	425.76	437.96	425.18	436.13	5,507,900	436.13

B Model Comparison Table

Table 4: Comparison of GBR Models: Gaussian [11], Pseudo-Huber [12], and Laplace Loss [11] Functions. This table summarizes the average forecasting errors computed over the entire rolling forecast period, which is 5 years from beginning of June 2025.

Horizon	**Gaussian**			**Pseudo-Huber**			**Laplace**		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
1	5.81	7.89	1.25	5.74	8.01	1.30	**3.61**	**4.80**	**0.75**
2	10.65	15.32	2.29	10.62	15.67	2.33	**9.62**	**14.44**	**2.07**
3	14.08	20.41	3.06	14.23	20.84	3.10	**13.44**	**19.89**	**2.90**
4	17.20	24.61	3.71	17.17	25.07	3.74	**16.66**	**24.43**	**3.60**
5	20.14	28.43	4.33	20.24	28.91	4.37	**19.44**	**28.27**	**4.19**
Mean	13.18	19.73	2.93	13.60	19.70	2.97	**12.95**	**18.77**	**2.70**

C Predictor Importance of Gaussian and Pseudo-Huber GBR Models

C.1 GBR with Gaussian Loss

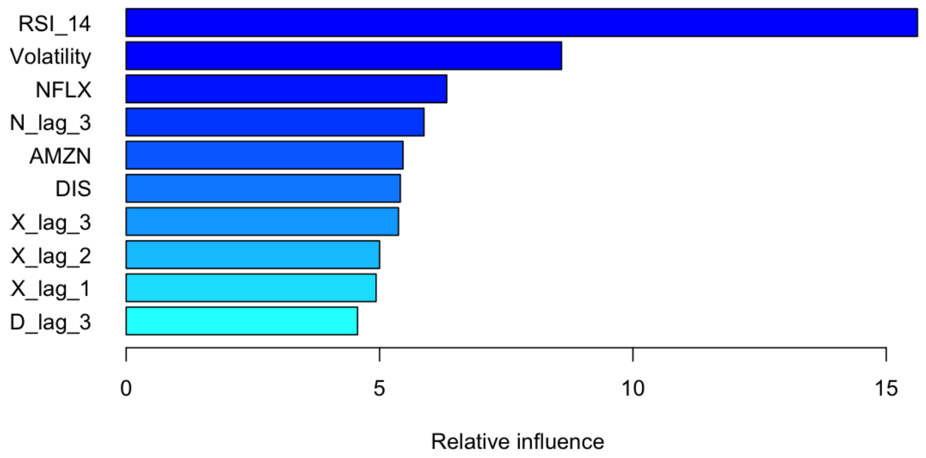


Figure 15: Predictor variable importance spectrum for Gaussian [11] GBR model

C.2 GBR with Pseudo-Huber Loss

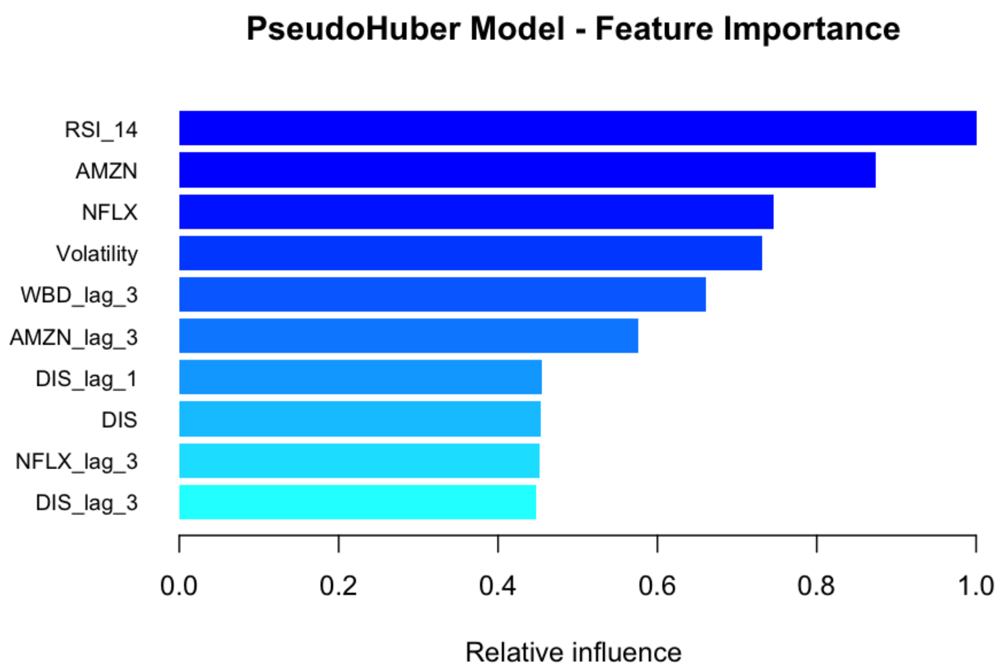


Figure 16: Predictor variable importance spectrum for Pseudo-Huber [12] GBR model

D Zoomed Forecast with 5day-ahead Window

D.1 GBR with Gaussian Loss

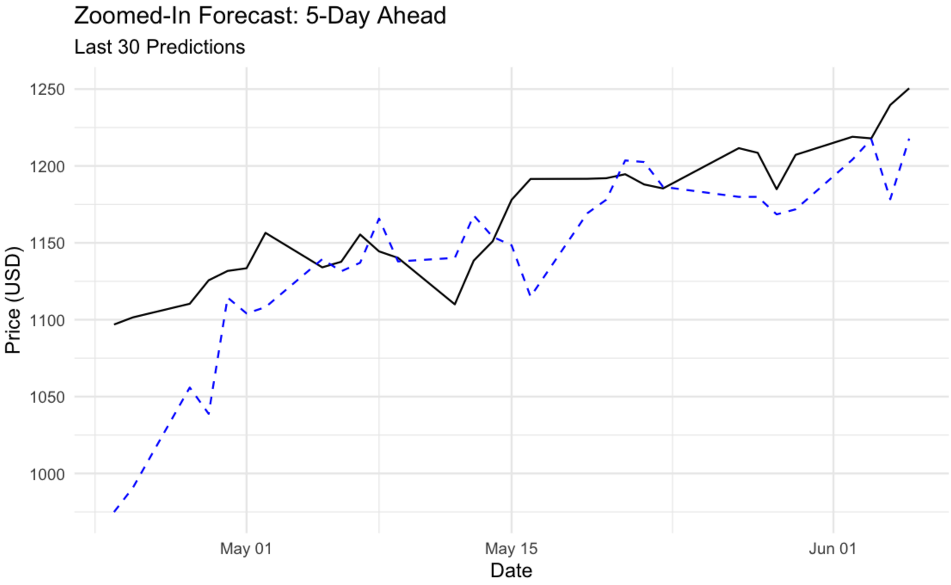


Figure 17: Trend Modeling with Gaussian Loss [11]

D.2 GBR with Pseudo-Huber Loss

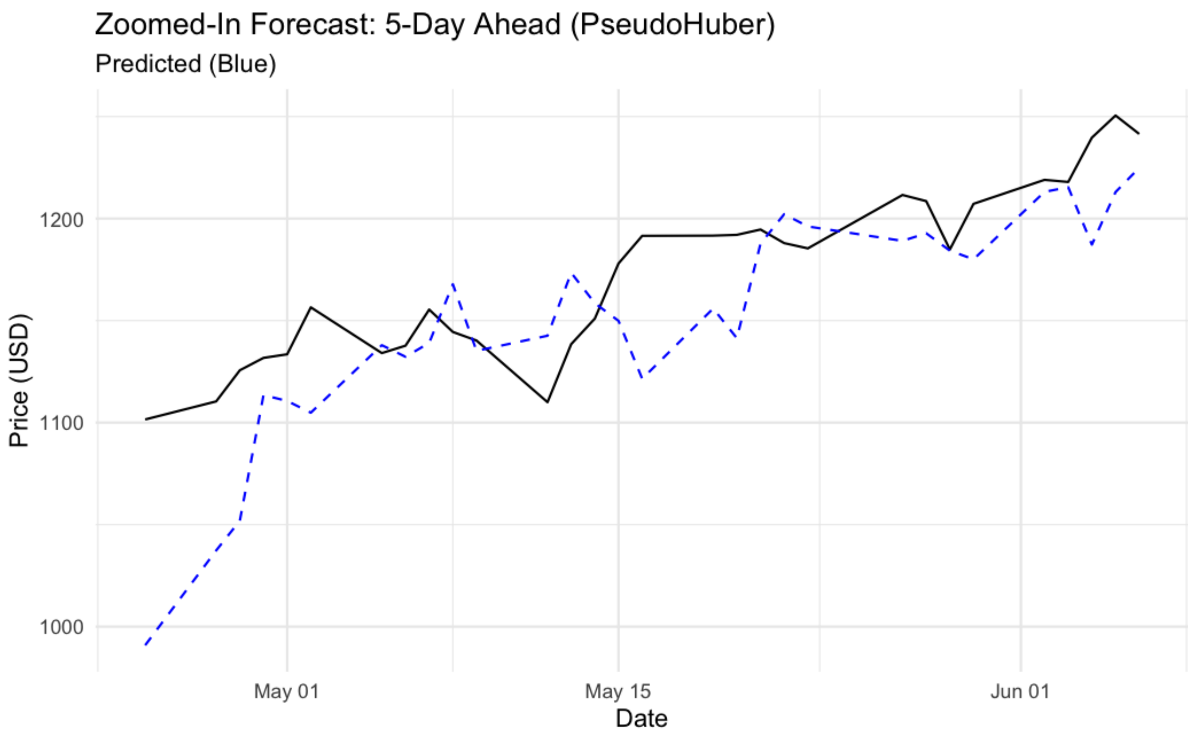
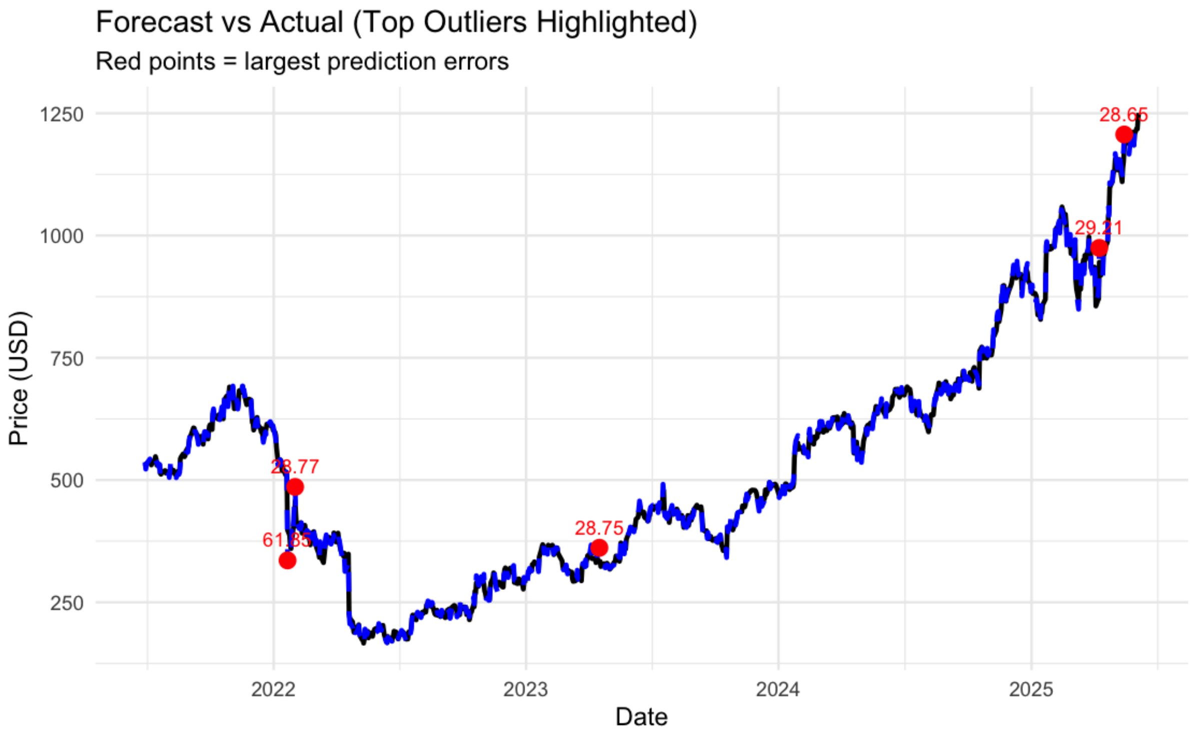


Figure 18: Trend Modeling with Pseudo-Huber Loss [12]

E Outliers and Prediction Errors across Gaussian, Pseudo-Huber and Laplace Models

E.1 GBR with Gaussian Loss



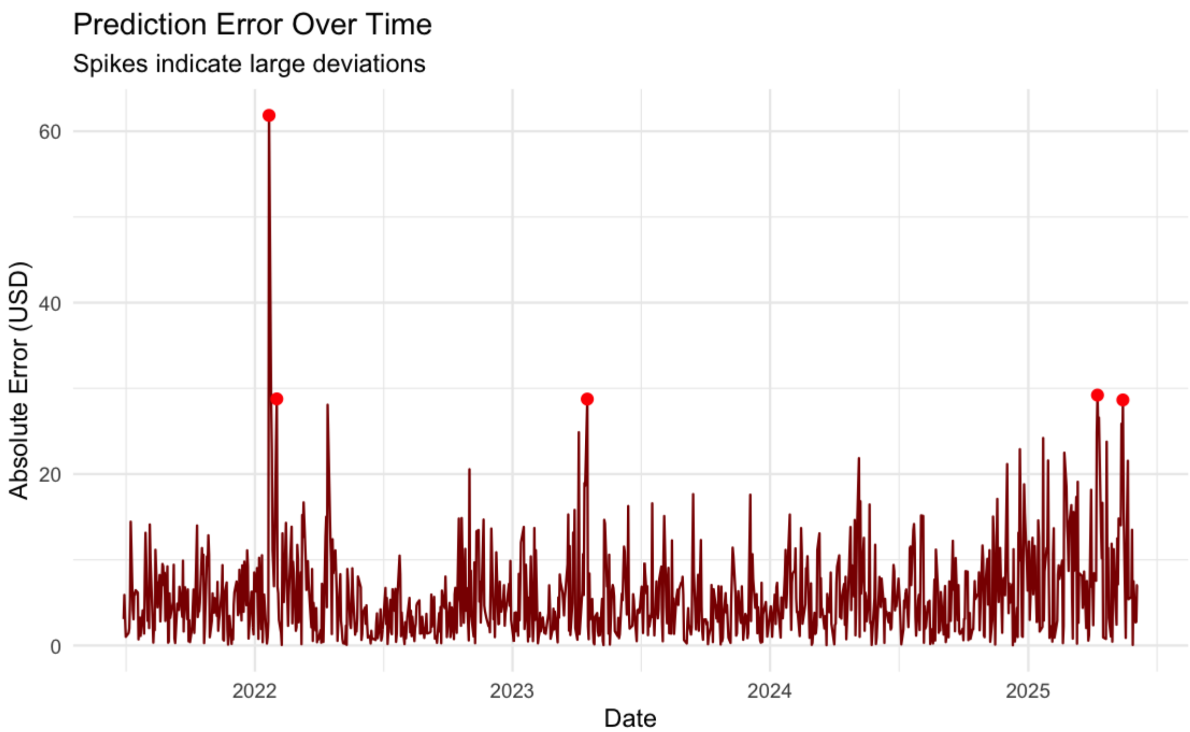
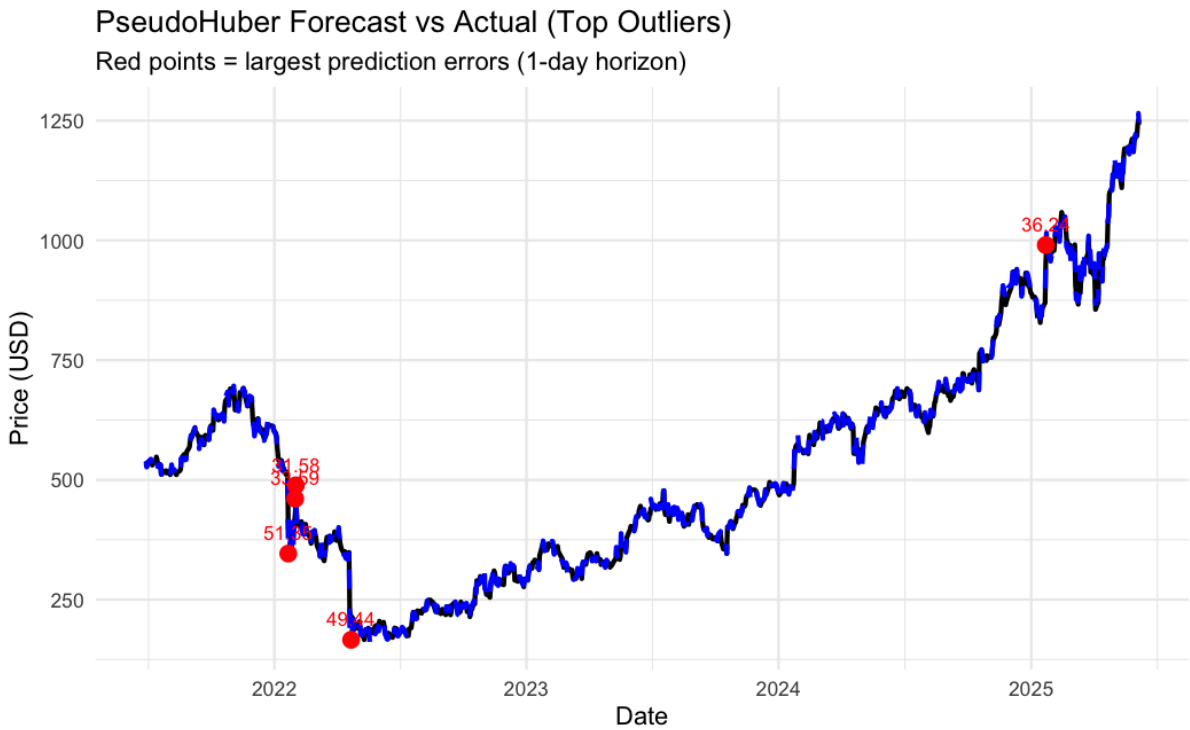


Figure 19: Comparison of forecast errors: (Top) Gaussian model [11] outliers; (Bottom) Gaussian model absolute errors over time. Errors mainly on sharp drops (negative shocks), with the largest outliers up to 60 USD.

E.2 GBR with Pseudo-Huber Loss



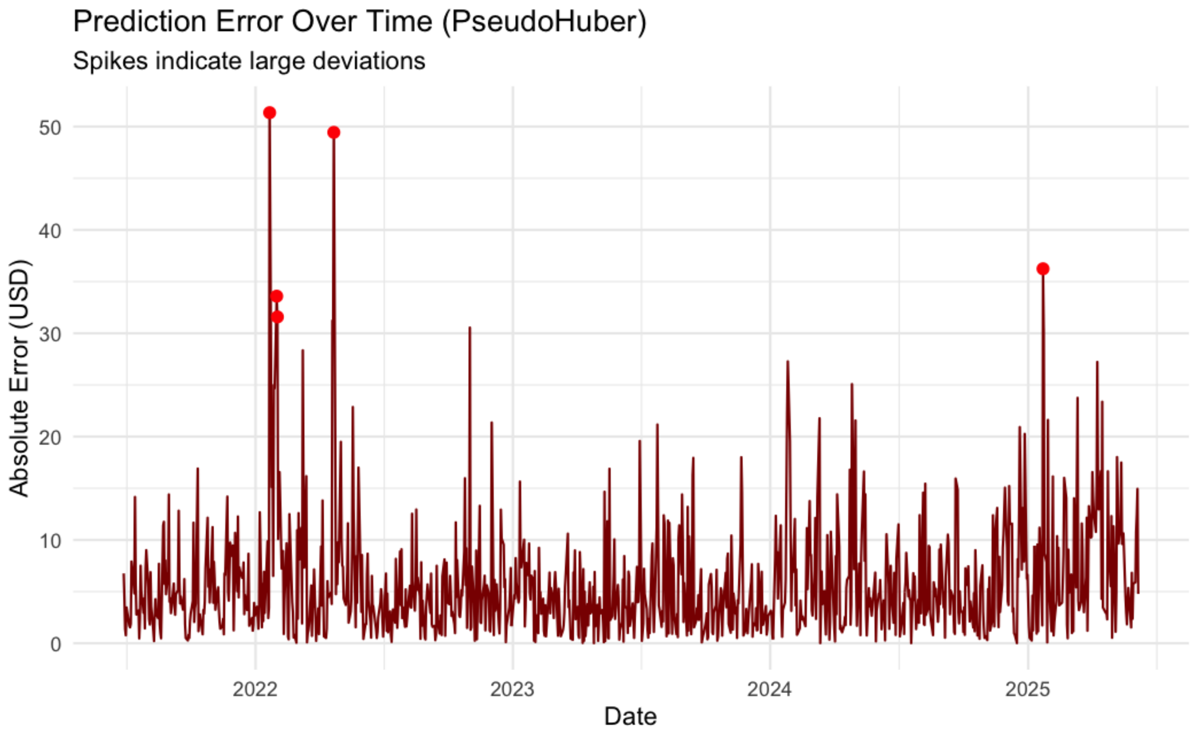
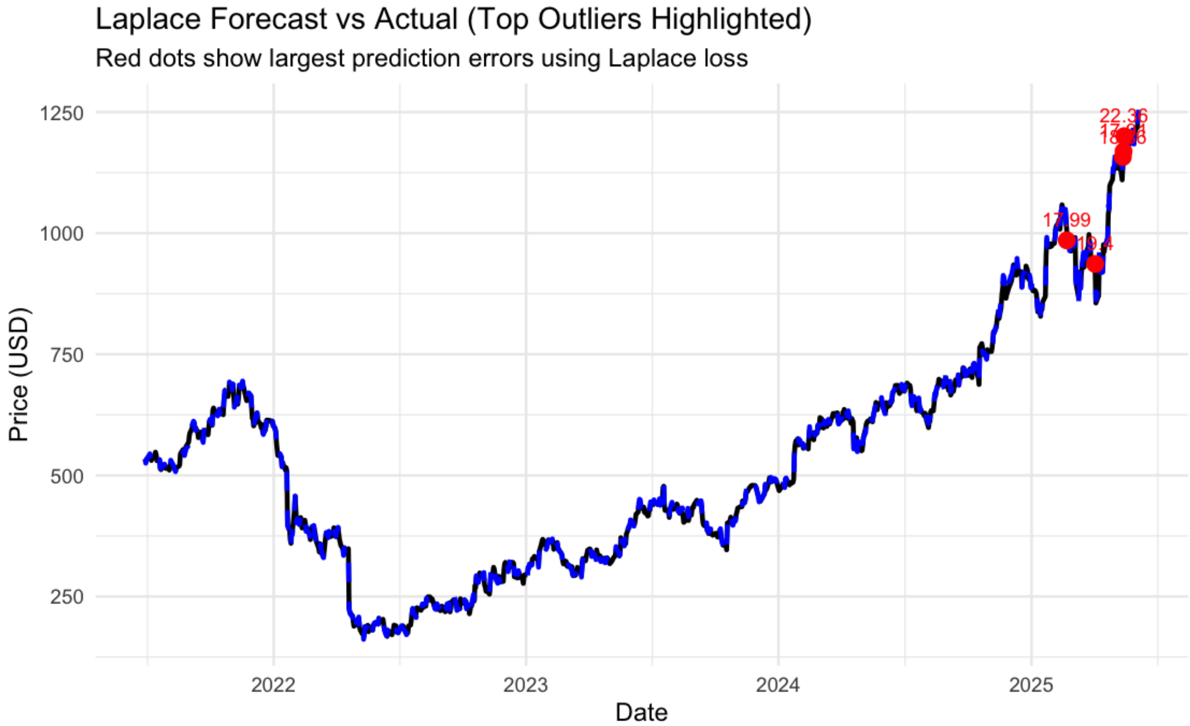


Figure 20: Comparison of forecast errors: (Top) Pseudo-Huber [12] model outliers; (Bottom) Pseudo-Huber model absolute errors over time. Errors occur during both sharp rises and drops, with outliers up to 50 USD.

E.3 GBR with Laplace Loss



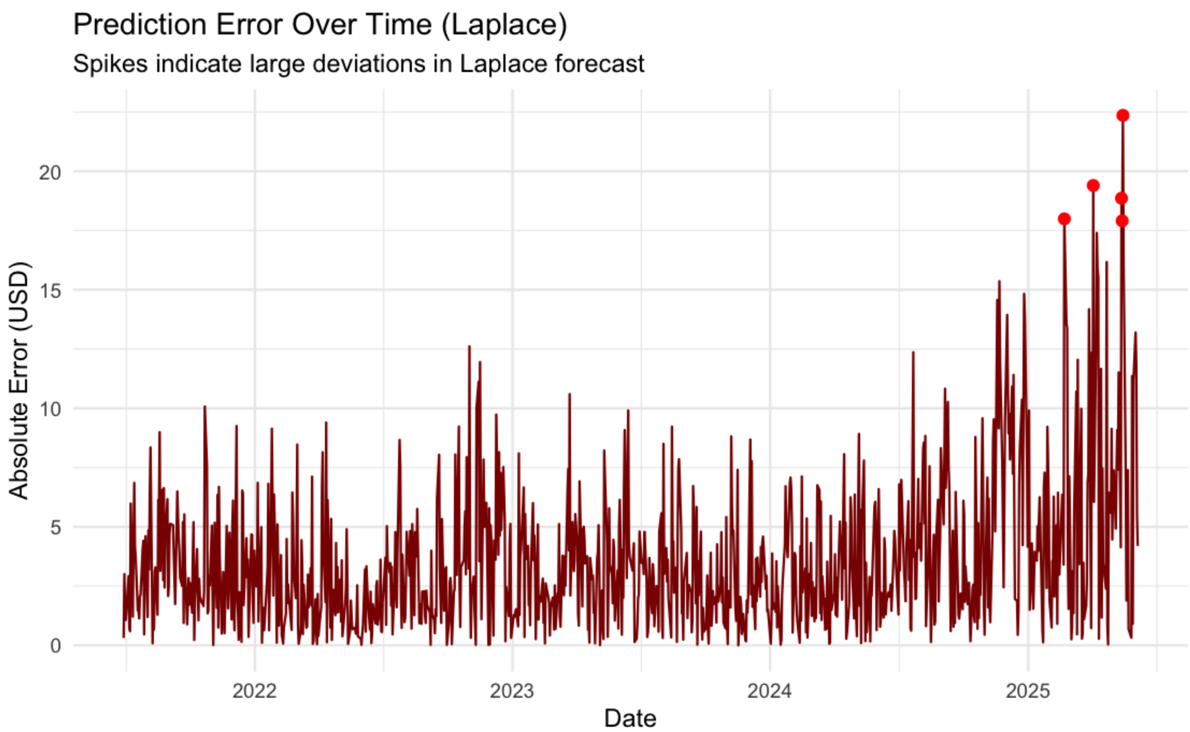


Figure 21: Comparison of forecast errors: (Top) Laplace model [11] outliers; (Bottom) Laplace model absolute errors over time. Errors occur mainly during steep upward movements, with smaller outliers (max 20 USD).

F Derivation of Confidence Interval Bound for Laplace Residuals

Let the model residuals $e = y - \hat{y}$ follow a Laplace distribution centered at zero with scale parameter b:

$$e \sim \text{Laplace}(0, b)$$

The probability density function (PDF) of the Laplace distribution is given by:

$$f(e) = \frac{1}{2b} \exp\left(-\frac{|e|}{b}\right)$$

We are interested in the probability that the residual lies within a symmetric interval around zero:

$$P(|e| \leq z) = \int_{-z}^{z} f(e)\, de$$

Using symmetry of the distribution:

$$P(|e| \leq z) = 2 \int_{0}^{z} \frac{1}{2b} \exp\left(-\frac{e}{b}\right)\, de$$

This simplifies to:

$$P(|e| \leq z) = \int_{0}^{z} \frac{1}{b} \exp\left(-\frac{e}{b}\right)\, de$$

Integrating:

$$P(|e| \leq z) = \left[-\exp\left(-\frac{e}{b}\right)\right]_0^z = 1 - \exp\left(-\frac{z}{b}\right)$$

Thus, the cumulative probability that the residual lies within the interval $[-z, z]$ is:

$$\boxed{P(|e| \leq z) = 1 - \exp\left(-\frac{z}{b}\right)}$$

Solving for z given a desired confidence level α, we obtain:

$$z = -b \cdot \ln(1 - \alpha)$$

Since $b = \frac{\text{MAD}}{\ln(2)}$, the interval for a given prediction becomes:

$$\hat{y} \pm z = \hat{y} \pm \frac{\text{MAD}}{\ln(2)} \cdot \ln\left(\frac{1}{1-\alpha}\right)$$

For example, for a 99% confidence interval:

$$z = \frac{\text{MAD}}{\ln(2)} \cdot \ln(100) \approx 6.64 \cdot \text{MAD}$$

This is consistent with the formulation presented in [15].