A COMPARATIVE STUDY OF ARIMA AND LSTM MODELS FOR FORECASTING NETFLIX TRADING VALUE

Ungureanu Viorica Mariela, Assoc. Prof. PhD

“Constantin Brâncuși” University of Târgu Jiu, ROMANIA

ABSTRACT: Forecasting financial time series is a fundamental task for investors, analysts, and financial institutions. This study evaluates the predictive performance of ARIMA and LSTM models on the daily trading value of Netflix (NFLX), defined as the product of the adjusted closing price and trading volume. The dataset is pre-processed and partitioned into training and test subsets to ensure robust model evaluation. ARIMA models are employed to capture linear temporal dependencies, while LSTM networks are utilized to learn nonlinear and long-term patterns inherent in financial time series. Forecast accuracy is assessed using standard error metrics, including MAE, RMSE, and MAPE. The comparative analysis reveals the strengths and limitations of each approach, offering practical insights into their applicability for financial forecasting.

KEY WORDS: Financial Time Series Forecasting, ARIMA Model, LSTM Neural Network, Stock Market Prediction, Netflix Trading Value

1. INTRODUCTION

Financial time series forecasting is a challenging problem due to the non-stationary, noisy, and often non-linear nature of financial data. Traditional statistical methods, such as the Autoregressive Integrated Moving Average (ARIMA) model, have been widely applied in econometrics and quantitative finance, offering interpretable parameters and reliable modelling of linear trends and seasonal components. However, financial series frequently exhibit non-linear dependencies, abrupt changes, and long-range correlations, which limit the predictive accuracy of purely linear models. Techniques from recent research on hybrid statistical and AI-based generation of time series [5] demonstrate the potential of combining decomposition and neural network approaches to better model complex temporal patterns.

Recent advances in deep learning have introduced more flexible architectures for temporal modelling. Among these, Long Short-Term Memory (LSTM) networks—a variant of recurrent neural networks (RNNs)—have demonstrated strong capabilities in learning complex temporal dependencies and capturing non-linear dynamics within sequential data. Unlike classical models, LSTMs can retain long-term information through gated mechanisms, allowing them to adapt to structural breaks and volatility commonly observed in financial markets. These characteristics make LSTMs particularly effective for capturing relationships that evolve over time and are not easily represented by linear models.

Current research supports the complementary nature of these two approaches. Traditional models like ARIMA remain valuable for their interpretability and performance on stationary or quasi-linear series, while LSTMs and other deep learning methods are better suited for complex, non-stationary environments [7]. Recent reviews highlight that hybrid and deep learning-based approaches are becoming dominant in time series forecasting due to their superior adaptability and accuracy in real-world applications [6], [4].

In this study, we focus on forecasting the daily trading value of Netflix (NFLX), defined as the product of the adjusted closing price and daily trading volume, as a proxy for market activity. The objective is to evaluate and compare the predictive performance of ARIMA and LSTM models on this univariate time series, emphasizing the advantages and limitations of statistical versus neural network-based methods in financial forecasting.

2. METHODOLOGY

2.1 Data Description and Preprocessing

The dataset analysed in this study was obtained from the Kaggle repository ("Netflix Stock Price History" by Adil Shamim [8]). It consists of daily trading data for Netflix (NFLX) covering the period from January 2002 to January 2025 (see Figure 1). It should be noted that the dataset was used primarily to compare the performance of different forecasting methods, rather than for a detailed market analysis.

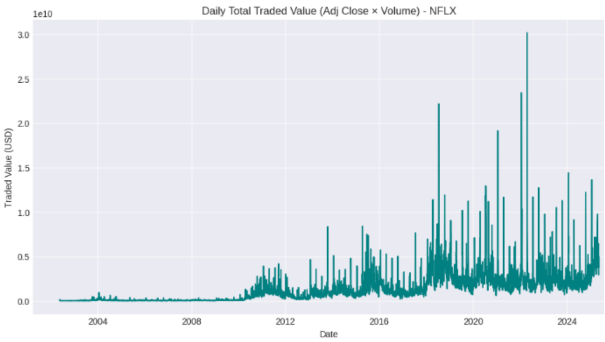

Figure 1

The dataset includes standard stock market attributes such as *Date*, *Open*, *High*, *Low*, *Close*, *Adjusted Close*, and *Volume*.

The primary variable of interest, referred to as the **trading value**, was computed as the product of the adjusted closing price and the daily trading volume (*Adjusted Close* × *Volume*), providing a measure of the total market activity for each trading day.

Prior to modelling, the dataset underwent standard preprocessing procedures. Missing values were handled through linear interpolation using the *pandas* interpolate(method='linear') function, which replaces each missing entry with the arithmetic mean of its immediate neighbours, ensuring continuity in the time series. Outliers were addressed by substituting anomalous observations with the average of surrounding values to reduce the influence of extreme fluctuations on model training. The series was then normalized using Min–Max scaling to facilitate convergence of the neural network models. Finally, the dataset was partitioned into training and test subsets, with the last 10–11 days reserved for evaluating out-of-sample forecasting performance.

2.2 ARIMA-Based Models

The Autoregressive Integrated Moving Average (ARIMA) model is a classical statistical approach widely used for forecasting univariate time series [1-2]. ARIMA models capture linear dependencies and trends through the combination of three components: the autoregressive (AR) term, the differencing (I) term, and the moving average (MA) term.

The general ARIMA (p, d, q) model is defined as: $\varphi(B)(1-B)^d y_t = \theta(B)\varepsilon_t$, where

$$\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \ldots - \varphi_p B^p,$$
$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q,$$
$$\varepsilon_t \sim \text{i.i.d. } (0, \sigma^2),$$

with B denoting the backshift operator, d the order of differencing, p the autoregressive order, and q the moving average order.

To determine the appropriate ARIMA orders p and q, we analysed the autocorrelation (ACF) and partial autocorrelation (PACF) functions of the series. The ACF measures correlation between the series and its lagged values, while the PACF measures correlation after removing intermediate lags. Significant spikes in the ACF suggest potential moving average (MA) components, and spikes in the PACF indicate possible autoregressive (AR) components [1].

Since the trading value series exhibited non-stationarity, first-order differencing (d=1) was applied to remove trends in the mean. The ACF and PACF of the differenced series were then inspected: sharp cutoffs in PACF indicate the AR order, while cutoffs in ACF suggest the MA order.

Candidate ARIMA models were further evaluated using information criteria (AIC and BIC) to balance model fit and complexity. Among them, ARIMA(2,1,4) achieved an AIC of 252819.13, slightly lower than ARIMA(1,1,1) with 252888.05, indicating a better fit. Therefore, ARIMA(2,1,4) was selected, effectively capturing the autocorrelation and moving average patterns of the series while maintaining parsimony.

In this study, two ARIMA-based approaches were applied to the Netflix trading value series:

A. Standard ARIMA. The model was fitted directly on the univariate trading value series to capture linear temporal dependencies. Parameters p, d and q were selected based on the autocorrelation and partial autocorrelation functions, as well as AIC/BIC criteria. Forecasts were generated for the test period, along with 95% confidence intervals. The predictive performance of the standard ARIMA model can be observed in Figure 2.

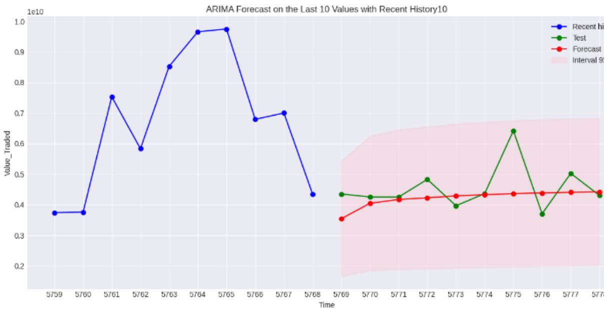

Figure 2

Figure 2 presents the last 10 values from the training set, the 10-step ARIMA forecast, and the corresponding test values. The ARIMA forecast aligns closely with the actual test values, capturing the overall dynamics of the total trading value. This performance is supported by the obtained error metrics (Mean Absolute Error (MAE) = 5.54×10^8, Mean Squared Error (MSE) = 6.27×10^{17}, Root Mean Squared Error (RMSE) = 7.92×10^8, Mean Absolute Percentage Error (MAPE) = 11.23%), as reported in Table 1.

B. STL-ARIMA (ARIMA on decomposed components): The trading value series was first decomposed using Seasonal-Trend decomposition via Loess (STL) into trend, seasonal, and residual components [3]. STL is a robust method that separates a univariate time series into additive components: the trend captures long-term changes, the seasonal component identifies repeating patterns over a fixed period, and the residual accounts for irregular fluctuations and noise. This decomposition allows each component to be modelled separately, improving forecast accuracy. In this study, the seasonal period was set to 12 to reflect the annual pattern in the data, ensuring that both trend and seasonality were properly captured for subsequent ARIMA modelling.

ARIMA models were then applied to the trend and residual components, while the seasonal component was reintroduced to produce combined forecasts [7]. Specifically, ARIMA(1,1,3) was fitted to the trend component and ARIMA(1,0,1) to the residual component. This setup allows the model to explicitly capture both long-term trends and short-term fluctuations. This approach explicitly models seasonality, allowing the model to better capture short-term fluctuations and improve forecast accuracy.

Performance metrics for the STL-based ARIMA model were comparable to the standard ARIMA model, with minor improvements in capturing seasonal variations.

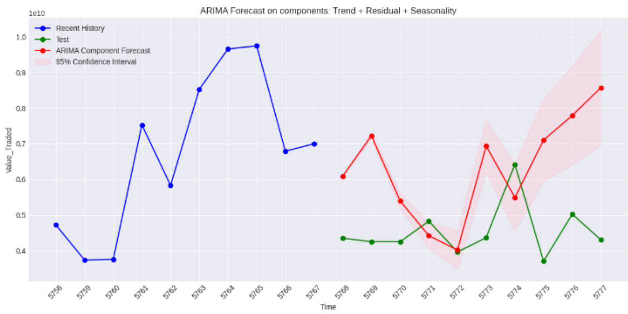

Figure 3

Figure 3 shows the last 10 training values, the STL-ARIMA forecast, and the actual test values, highlighting how trend and seasonality contribute to the predictive performance.

The STL-based ARIMA forecast yielded higher errors compared to the standard ARIMA model, with MAE = 1.97×10^9, MSE = 6.47×10^{18}, RMSE = 2.54×10^9, and MAPE = 4.73×10^1 %, reflecting the increased difficulty of modeling the residual component and capturing short-term fluctuations (Table 1).

2.3 LSTM Model

Long Short-Term Memory (LSTM) networks are a class of recurrent neural networks designed to capture long-term dependencies in sequential data [4-6]. Unlike ARIMA, LSTMs can model non-linear relationships and adapt to dynamic patterns in financial time series.

For the Netflix trading value, is considered the following LSTM approach:

Data preparation: Input sequences were created using a fixed look-back window of 60 days, with the target being the total trading value (Adjusted Close × Volume) of the next day. Only one-dimensional sequences were used for this main model, consistent with the ARIMA analysis. The data were normalized using Min-Max Scaler, and no rows were removed via early stopping or similar preprocessing.

Network architecture: The model consists of a single LSTM layer with 60 units, followed by a dense output layer. This simpler architecture captures nonlinear temporal dependencies in the product series while minimizing the risk of overfitting due to the limited sample size.

Training procedure: The model was trained using the Adam optimizer and the MSE loss function. Training was performed for 20 epochs with a batch size of 16. No early stopping was applied in this configuration.

Forecasting: Once trained, the LSTM generated forecasts using a 10-step recursive approach, where each predicted value was appended to the input sequence for the next step. Predicted values were rescaled to the original scale for comparison with actual trading values. Performance metrics for this univariate LSTM are shown in Table 1, while the visual representation of the forecasts, including the last 10 training values, the 10-step predictions, and the actual test values, is presented in Figure 4.

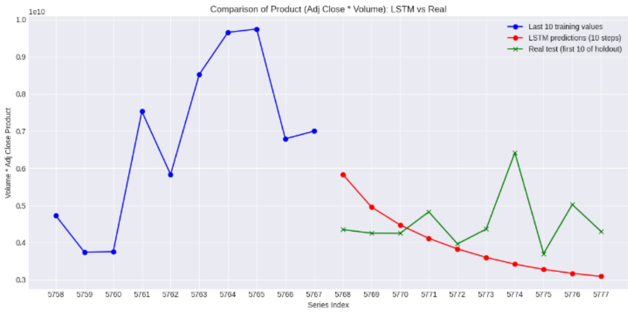

Figure 4

2.4 Evaluation Metrics

Forecasting performance across the different models—standard ARIMA, STL-based ARIMA, univariate LSTM, and bidimensional LSTM—was assessed using four widely adopted metrics:

Mean Absolute Error (MAE): MAE measures the average magnitude of the errors between predicted and observed values, without considering their direction. It provides an intuitive understanding of the typical deviation expected from the model's forecasts. Lower MAE values indicate more accurate predictions.

Mean Squared Error (MSE): MSE calculates the average of the squared differences between predicted and observed values. By squaring the errors, it penalizes larger deviations more heavily and provides a sense of the overall error variance. Lower MSE values indicate better overall predictive accuracy.

Root Mean Squared Error (RMSE): RMSE penalizes larger errors more heavily than MAE due to its quadratic formulation. It is particularly useful for identifying models that produce occasional large deviations and emphasizes overall forecast reliability.

Mean Absolute Percentage Error (MAPE): MAPE expresses forecast errors as a percentage of actual values, allowing for relative comparisons across different scales and series. It offers insight into the proportional accuracy of the predictions and is

especially informative for non-stationary financial series.
These metrics collectively provide a comprehensive view of model performance, highlighting both absolute and relative deviations between predicted and actual trading values. Based on the results summarized in Table 1, the standard ARIMA model consistently achieved the lowest errors across all metrics (MAE = 5.54×10^8, MSE = 6.27×10^{17}, RMSE = 7.92×10^8, MAPE = 11.2 %), indicating its strong capability to capture the overall dynamics of the Netflix trading value. The univariate LSTM produced slightly higher errors (MAE = 1.05×10^9, MSE = 1.79×10^{18}, RMSE = 1.34×10^9, MAPE = 21.4 %), reflecting its ability to capture nonlinear short-term variations, but with less overall accuracy than the standard ARIMA approach. The STL-based ARIMA model showed the largest errors (MAE = 1.97×10^9, MSE = 6.47×10^{18}, RMSE = 2.54×10^9, MAPE = 47.3 %), demonstrating that decomposing the series and modelling components separately did not improve forecasting performance in this case.
In addition to quantitative measures, visual inspection of forecast plots—including the last values from the training set, predicted values, and actual test values—complements the numerical evaluation, providing a clear understanding of each model's practical performance in financial time series forecasting. Based on both these assessments, the comparative analysis indicates that the standard ARIMA model outperforms both the univariate LSTM and the STL-based ARIMA in terms of overall accuracy and reliability for forecasting Netflix trading values. While the LSTM captures short-term nonlinear fluctuations, its higher errors suggest limited effectiveness for precise long-term predictions. The STL-based ARIMA's relatively poor performance further highlights that decomposition does not necessarily enhance forecast accuracy for this dataset. Together, these results emphasize that, for this specific financial time series, the classical ARIMA approach remains the most robust and dependable method for capturing the underlying market dynamics.

Table 1

Metric	ARIMA	ARIMA-STL	LSTM
MAE	**5.54×10^8**	1.97×10^9	1.05×10^9
MSE	**6.27×10^{17}**	6.47×10^{18}	1.79×10^{18}
RMSE	**7.92×10^8**	2.54×10^9	1.34×10^9
MAPE	**11.2%**	47.3 %	21.4%

5. CONCLUSION

The comparative evaluation of forecasting performance reveals clear distinctions between the models. The standard ARIMA consistently achieved the lowest errors across both absolute and relative metrics, demonstrating its strong capability to capture the overall dynamics of Netflix trading values. Its superior MAE, MSE, RMSE, and MAPE indicate reliable performance in both magnitude and proportional accuracy, making it the most robust choice for practical forecasting tasks.
The univariate LSTM, while capable of capturing short-term nonlinear fluctuations, produced higher errors, suggesting limited effectiveness for precise long-term predictions. This highlights that, for this dataset, the model's ability to learn complex patterns did not translate into superior overall accuracy compared to the simpler ARIMA approach.
The STL-based ARIMA model exhibited the largest errors, indicating that decomposing the series into trend, seasonal, and residual components and modelling them separately did not improve forecast accuracy. Although decomposition can provide interpretability, in this case it failed to enhance predictive performance.
Overall, these results illustrate that, for Netflix trading values, classical statistical approaches like ARIMA can outperform more complex deep learning methods in terms of reliability and accuracy. Combining quantitative metrics with visual inspection of forecast trajectories provides a comprehensive evaluation of model

performance, guiding the selection of the most appropriate method for financial time series forecasting.

REFERENCES

[1] Brockwell, P. J., Davis, R. A., *Introduction to Time Series and Forecasting*, Springer, New York, USA, 2002.

[2] Chatfield, C., *The Analysis of Time Series: An Introduction*, Sixth Edition, Chapman & Hall/CRC Texts in Statistical Science, CRC Press, 2016.

[3] Cleveland, R. B., *STL: A Seasonal-Trend Decomposition Procedure Based on Loess*, vol. 6, no. 1, pp. 3–73, 1990.

[4] Hall, T., Rasheed, K., *A Survey of Machine Learning Methods for Time Series Prediction*, Applied Sciences, 15(11), 5957, 2025.

[5] Ungureanu, I. I., Portase, R. L., Dinsoreanu, M., Lemnaru, C., Potolea, R., *Hybrid Statistical and AI Methods for Synthetic Time Series Generation in Practical Applications*, in 2024 IEEE 20th International Conference on Intelligent Computer Communication and Processing (ICCP), IEEE, pp. 1–8, October 2024.

[6] Kong, X., Chen, Z., Liu, W., Ning, K., Zhang, L., Muhammad Marier, S., Xia, F., *Deep Learning for Time Series Forecasting: A Survey*, International Journal of Machine Learning and Cybernetics, pp. 1–34, 2025.

[7] Taslim, D. G., Murwantara, I. M., *Comparative Analysis of ARIMA and LSTM for Predicting Fluctuating Time Series Data*, Bulletin of Electrical Engineering and Informatics, 13(3), pp. 1943–1951, 2024.

[8] Kaggle, *Netflix Stock Price History*, available at: https://www.kaggle.com/datasets/adilshamim8/netflix-stock-price-history