International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 4, April, 2025, pp. 1005-1013
Journal homepage: http://iieta.org/journals/isi

Time Series Analysis of Netflix's Stock Closing Prices: From Data Processing to Forecasting

Anass Zatri[1*], Khalid Zine-Dine[2]

[1] ENSIAS, Mohammed V University in Rabat, Rabat 10112, Morocco
[2] Faculty of Sciences, Mohammed V University in Rabat, Rabat 10106, Morocco

Corresponding Author Email: anass_zatri@um5.ac.ma

https://doi.org/10.18280/isi.300417

Received: 14 February 2025
Revised: 15 April 2025
Accepted: 22 April 2025
Available online: 30 April 2025

Keywords:
time series, forecasting, data science

ABSTRACT

This study delves into the complexities of time series forecasting, a field that intersects statistics, data science, and econometrics to predict sequential data patterns. Focusing on the challenge of forecasting Netflix's closing stock prices from February 2018 to January 2022, we evaluated the performance of three distinct models: SARIMA (seasonal autoregressive integrated moving average), Prophet, and XGBoost (extreme gradient boosting). Each model demonstrated unique strengths and limitations. SARIMA offered solid baseline accuracy but struggled to capture abrupt price fluctuations inherent in stock market behavior. Prophet enhanced interpretability by effectively modeling seasonality and trends, yet it showed limitations in precision. In contrast, XGBoost excelled in capturing complex nonlinear patterns and better reflected the dynamic nature of stock price movements. The core innovation of this research lies in the development of a hybrid model combining SARIMA and XGBoost through optimized weighting. This hybrid approach outperformed individual models by balancing statistical robustness with adaptive learning capabilities, leading to improved accuracy and better trend representation. However, while Root Mean Square Error (RMSE) was used as the primary evaluation metric, it became evident that RMSE alone is insufficient to fully assess forecasting quality, particularly in capturing trend dynamics. This highlights the necessity for more comprehensive evaluation metrics, paving the way for future research in advancing time series forecasting methodologies.

1. INTRODUCTION

The challenge of predicting time series data has been a longstanding focus for researchers, mathematicians, and statisticians for many years. Situated at the intersection of statistics, econometrics, and data science, time series forecasting functions as an indispensable asset across a diverse array of industries, including finance, economics, meteorology, and energy management. The appeal of time series prediction extends beyond academic curiosity; its practical applications are equally significant. Reliable forecasts can aid decision-making, improve operational efficiency, and help identify emerging trends.

Among these applications, stock price forecasting holds a particularly critical role in financial markets. Accurate predictions of stock prices can support investors in portfolio management, risk assessment, and strategic planning. For companies, it can provide insights into market sentiment and valuation trends. Furthermore, financial institutions and hedge funds increasingly rely on advanced forecasting techniques to gain a competitive edge in high-frequency trading and investment strategies. Thus, the ability to anticipate market movements, even partially, can translate into substantial economic value.

However, forecasting stock prices presents unique and complex challenges. Unlike other time series data, financial markets are characterized by high volatility, where prices can fluctuate significantly within short periods due to market sentiment, investor behavior, or speculative actions. This volatility introduces a level of unpredictability that traditional models often struggle to capture effectively. Moreover, stock prices are inherently non-stationary, meaning their statistical properties, such as mean and variance, change over time, complicating model assumptions and requiring advanced techniques to ensure robust predictions.

Another major challenge lies in the influence of external shocks. Events such as geopolitical tensions, economic crises, and global pandemics like COVID-19 can cause abrupt and severe disruptions in financial markets. The COVID-19 pandemic, in particular, led to unprecedented levels of uncertainty and market turbulence, highlighting the vulnerability of stock prices to factors beyond historical patterns. These external variables are difficult to model and often require adaptive or hybrid approaches capable of responding to sudden changes in market dynamics.

In addition, the efficient market hypothesis (EMH) suggests that stock prices fully reflect all available information, implying that predicting future price movements based solely on past data is inherently limited. While this theory challenges the feasibility of consistent outperformance, advancements in machine learning and data-driven approaches have reopened the debate by uncovering subtle patterns and dependencies that traditional statistical methods might overlook.

In this paper, we examine three distinct time series forecasting models: SARIMA, FBProphet, and XGBoost, to evaluate their respective strengths and limitations in stock price prediction. We will concentrate on Netflix's stock prices over the period spanning from February 2018 to January 2022. Stock market predictions, known for their volatility and unpredictability, provide an ideal context to test and compare the capabilities of these models.

These datasets are publicly available under the CC0 Public Domain License, which supports the reproducibility of scientific research and encourages open access to data for researchers globally.

2. LITERATURE REVIEW

Time series forecasting has long been a cornerstone of quantitative analysis across various domains, notably in finance where anticipating market movements is of strategic importance. A time series is fundamentally a sequence of data points indexed in time order, capturing the dynamic behavior of systems over intervals. The complexity of forecasting arises from the intrinsic characteristics of time series data, such as trends, seasonality, autocorrelation, and potential non-stationarity. Over the decades, a wide range of methodologies has been developed to address these challenges, evolving from classical statistical approaches to modern machine learning and deep learning techniques.

2.1 Classical statistical models

The foundation of time series forecasting is built upon statistical models that assume linear relationships within data. Among the earliest models are the Autoregressive (AR) and Moving Average (MA) models. The AR model predicts future values based on a linear combination of past observations, while the MA model relies on past forecast errors. The integration of these two approaches led to the ARMA model, suitable for stationary time series where statistical properties remain constant over time [1].

However, real-world data, especially financial time series, often exhibit non-stationary behavior due to trends or external shocks. To address this, the ARIMA model was introduced, incorporating differencing techniques to transform non-stationary data into a stationary form before applying ARMA processes. ARIMA became a widely adopted tool due to its balance of simplicity and effectiveness in capturing linear dependencies.

When seasonality is present, ARIMA is extended to SARIMA by adding seasonal components. SARIMA models are effective for datasets with clear periodic patterns, such as monthly sales or quarterly economic indicators. Despite their robustness, these models are constrained by their linear assumptions and require manual identification of parameters like order of differencing, lags, and seasonal cycles. They also struggle to adapt to structural changes or nonlinear dynamics, which are prevalent in stock market data.

2.2 Multivariate statistical approaches

Univariate models, while useful, often ignore the influence of external variables. To capture interactions among multiple time-dependent variables, Vector Autoregression (VAR) models were developed. VAR models generalize the AR approach by allowing for interdependencies across several series, making them valuable in macroeconomic and financial analyses [2].

In cases where non-stationary multivariate series exhibit long-term equilibrium relationships, the Vector Error Correction Model (VECM) is employed. VECM integrates cointegration concepts, ensuring that deviations from equilibrium are corrected over time [3]. While these models enhance forecasting by incorporating more information, they remain bound by linear frameworks and assumptions of stationarity or cointegration, limiting their flexibility in volatile environments.

2.3 Additive models: The role of prophet

Recognizing the need for more adaptive and user-friendly forecasting tools, Facebook introduced Prophet, an additive model designed to handle datasets with strong seasonality and trend shifts [4]. Prophet decomposes time series into trend, seasonality, and holiday effects, allowing analysts to model complex patterns without deep statistical expertise.

One of Prophet’s key strengths lies in its ability to manage multiple seasonalities, handle missing data, and remain robust to outliers. It is particularly suited for business applications like demand forecasting or website traffic analysis. However, Prophet assumes that trends evolve smoothly over time, which can be a significant limitation when applied to financial time series characterized by abrupt changes and irregular patterns. While it offers interpretability and ease of implementation, its predictive performance on volatile datasets such as stock prices often falls short compared to more sophisticated algorithms.

2.4 Machine learning approaches

The advent of machine learning has transformed time series forecasting by introducing models capable of capturing nonlinear relationships and complex interactions within data. Unlike statistical models, machine learning techniques do not require assumptions about data distribution or linearity, offering greater flexibility [5].

Among these techniques, ensemble methods, particularly gradient boosting algorithms like XGBoost, have gained prominence. XGBoost builds a series of decision trees sequentially, where each tree attempts to correct the errors of the previous ones. This iterative process, combined with regularization techniques, enhances predictive accuracy while controlling overfitting.

In time series forecasting, XGBoost is adapted through feature engineering, where temporal structures are converted into supervised learning problems. Features such as lag variables, rolling averages, and calendar effects are crafted to provide context. While XGBoost excels in handling structured data and delivering high accuracy, it lacks inherent mechanisms to account for temporal dependencies, relying heavily on the quality of engineered features. Moreover, its black-box nature can be a drawback in domains where interpretability is crucial.

2.5 Deep learning and LSTM networks

Deep learning has further expanded the horizons of time series forecasting, particularly through architectures designed to process sequential data. Recurrent Neural Networks (RNNs) introduced the concept of retaining information across time steps, but they suffered from issues like vanishing gradients. Long Short-Term Memory (LSTM) networks addressed these limitations by incorporating memory cells that retain information over longer sequences [6].

LSTM networks are adept at capturing long-term dependencies and complex temporal patterns, making them suitable for financial forecasting where historical context can influence future movements [7]. Their ability to model nonlinear dynamics without extensive feature engineering is a significant advantage. However, LSTMs require large datasets to perform effectively, are computationally intensive, and often criticized for their lack of transparency. The interpretability challenge, coupled with the risk of overfitting, necessitates careful tuning and validation.

2.6 Evaluation metrics in time series forecasting

Assessing the performance of forecasting models is a critical aspect of time series analysis. Traditional metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are commonly used to quantify prediction accuracy [8]. These metrics provide insights into the average deviation between predicted and actual values.

However, these measures primarily focus on point-wise accuracy and may not adequately capture a model's ability to follow underlying trends or predict directional changes. In financial contexts, correctly anticipating whether a stock price will rise or fall can be more valuable than minimizing numerical error. This has led to discussions around incorporating directional accuracy metrics and developing custom evaluation frameworks that better align with practical forecasting objectives.

2.7 Critical analysis and synthesis

The evolution of time series forecasting reflects a continuous effort to balance simplicity, interpretability, and predictive power. Classical statistical models offer transparency and perform well on linear, stationary data with predictable patterns. However, their rigidity makes them less suitable for complex, volatile datasets.

Machine learning models, particularly XGBoost, provide enhanced flexibility and accuracy but depend heavily on feature engineering and often operate as black-box systems. Deep learning approaches like LSTM push the boundaries of what can be modeled but introduce challenges related to data requirements, computational cost, and explainability.

Additive models like Prophet fill a niche for business applications requiring rapid deployment and interpretability but are not designed for highly erratic data such as stock prices.

Finally, the choice of evaluation metrics plays a crucial role in determining the perceived success of a forecasting model. Over-reliance on traditional error metrics can obscure important aspects of model performance, particularly in domains where trend fidelity and directional correctness are paramount.

In conclusion, the literature highlights that no single approach universally outperforms others across all scenarios. The selection of an appropriate forecasting method must consider the specific characteristics of the dataset, the forecasting horizon, the need for interpretability, and the operational context. This ongoing challenge continues to drive research toward more adaptive, accurate, and interpretable forecasting solutions.

3. MODEL METHODOLOGIES FOR TIME SERIES FORECASTING

3.1 SARIMA

SARIMA is an extension of the ARIMA model that incorporates seasonality. Let's decompose its components:

Autoregressive (AR): This component captures the relationship between an observation and a number of lagged observations (previous time steps). Mathematically, the AR term can be represented as:

$$AR(p) = \emptyset_1 Y_{t-1} + \emptyset_2 Y_{t-2} + \cdots + \emptyset_p Y_{t-p}$$

where, $\emptyset$ represents the parameters of the model, and p is the order of the autoregression.

Moving Average (MA): This facet of the model represents the relationship between an observation and a residual error from a moving average model applied to lagged observations. In mathematical terms, this can be written as:

$$MA(q) = \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q}$$

where, θ are the parameters of the model, e represents the error term, and q is the order of the moving average.

Integrated (I): This refers to the differencing of observations to make the series stationary (i.e., constant mean and variance over time). A series is integrated of order d if we difference it d times to achieve stationarity.

$$I(d) = (1 - B)^d Y_t$$

where, B is the backshift operator, and d is the order of differencing.

Seasonality is incorporated into SARIMA as seasonal AR, MA, and differencing terms. The seasonal component is generally represented as SARIMA (p,d,q) (P, D, Q)s , where P, D, Q are the seasonal counterparts of p,d,q and s is the seasonal length [9-11].

3.2 FBprophet

Created by Facebook, FBprophet is a tool specifically designed to forecast time series data with patterns that fluctuate across different time scales, such as yearly, weekly, and daily cycles. It excels when working with time series that exhibit strong seasonal patterns and have multiple years of historical data. FBprophet is also highly adaptable, managing missing data and changes in trends effectively while being resilient to outliers.

The underlying model for FBprophet can be described as:

$$y(t)=g(t)+s(t)+h(t)+\epsilon t$$

where:

$g(t)$ represents the trend function which models non-periodic changes.

$s(t)$ captures periodic changes, i.e., seasonality.

$h(t)$ denotes the effects of holidays.

ϵt is the error term.

The trend function $g(t)$ is typically modeled using a logistic growth model or a piecewise linear model. Seasonal effects $s(t)$ are modeled using Fourier series.

3.3 XGBoost

XGBoost is not inherently a time series forecasting model but rather a powerful ensemble machine learning algorithm primarily used for structured/tabular data. It builds the model in a stage-wise fashion and generalizes the model by allowing optimization of an arbitrary differentiable loss function.

The principle behind XGBoost is to iteratively add new models (typically decision trees) to correct the errors made by the existing set of models. Mathematically, if we let y^i be the prediction of the i-th tree, then the prediction of n trees is:

$$y(x) = \Sigma_{i=1}^{n} y^i(x)$$

In the context of time series forecasting, features derived from the time series data, such as lags, rolling statistics, and date-related attributes, serve as input for the model, enabling it to predict future data points.

4. DATA ANALYSIS AND PREPARATION FOR FORECASTING

In this section, we dive deep into a practical application of time series forecasting, focusing on the closing prices of Netflix stock from February 2018 to January 2022. This span of nearly four years provides a comprehensive dataset, allowing for a robust analysis and prediction.

4.1 Data acquisition and preliminary analysis

The dataset in consideration captures the closing prices of Netflix over the selected duration. It is crucial to understand the behavior of this time series before applying any forecasting techniques. A visualization, typically a time series plot, serves as an excellent starting point. By plotting the data, we can obtain insights into potential trends, seasonalities, or other patterns in the stock's behavior.

Visually, we observe on the Figure 1 an annual seasonality between 2018 and 2019, followed by an upward trend starting from the end of 2019. Additionally, there is a noticeable decline in the stock price beginning in November 2022.

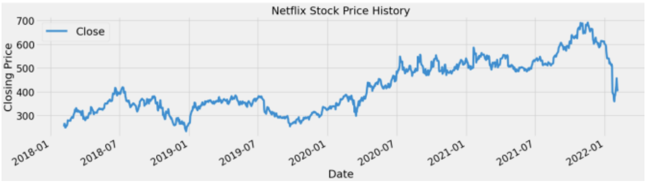


Figure 1. Netflix stock price history (February 2018–January 2022)

4.2 Data cleaning and imputation

Despite best efforts at data collection, datasets occasionally have missing values. This can be due to a multitude of reasons: holidays when the stock market is closed, data recording errors, or other unanticipated factors.

Missing data can distort predictions if not handled properly.

In our analysis, we adopted a simple yet effective approach to manage missing data points: imputation by averaging. For any missing value, we calculated the mean of the immediately preceding and succeeding values. This method ensures a smooth transition and reduces any potential shocks in the time series that could arise from abrupt imputations.

4.3 Model preparation: Training and testing

Before we delve into forecasting, it is paramount to split the dataset into training and test sets. This approach ensures that we train our models on a majority of the data and subsequently test their accuracy on unseen data. Given the noticeable decline in stock price beginning in November 2022, we decided to use the data from October 2022 as our test set. The historical data leading up to this point forms our training set.

5. FORECASTING WITH DEFINED MODELS

With the dataset duly preprocessed and split, we proceeded to apply the forecasting models detailed earlier in this paper: SARIMA, FBprophet, and XGBoost. Each of these models offers unique methodologies and insights, making it fascinating to juxtapose their predictions.

5.1 SARIMA

To ascertain the optimal parameters for SARIMA, PAC (Partial Autocorrelation Function) and AC (Autocorrelation Function) plots were utilized to probe the p, d, and q orders. A grid search was subsequently conducted using the stepwise_model to identify the optimal parameters, landing on arima_order=(1,1,0) and seasonal_order=(2,1,0,52). However, it's salient to note that the prediction variance remains slight, with a relatively flat prediction curve.

After analyzing our initial time series data, the Dickey-Fuller test yielded a p-value of P=0.42 as shown in Figure 2, which is above the 0.05 threshold. This result indicates that our series is non-stationary. In simpler terms, the series doesn't have consistent statistical properties like mean and variance over time.

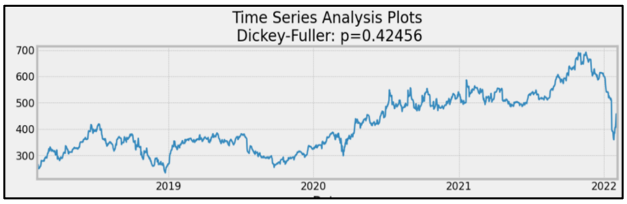


Figure 2. Stationarity test for the series (Dickey-Fuller)

One common approach to make a time series stationary is to use differencing [12, 13]. This involves subtracting the current value from the previous one. In our case, we applied first-order differencing as depicted in Figure 3 below.

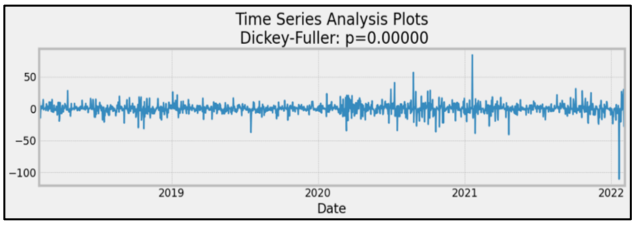


Figure 3. First-order differencing and Dickey-Fuller test

After this transformation, we performed the Dickey-Fuller test again to evaluate the stationarity of the now transformed series. The visual representation from our updated chart shows a more consistent pattern, suggesting that the series has become stationary (p<0,05).

Upon examining the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots of our time series data, vital insights regarding the SARIMA model parameters were gleaned:

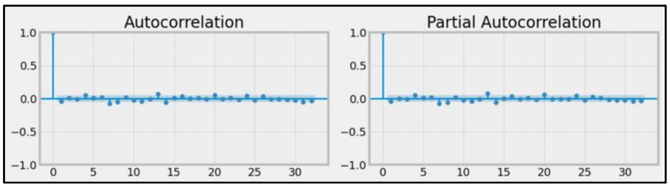


Figure 4. Autocorrelation function and partial autocorrelation function

The sharp cut-off observed in the ACF plot (Figure 4) after the first lag suggests the presence of a significant moving average component in the data. This indicates a 'q' parameter value of 1 for the SARIMA model.

Similarly, the PACF plot (Figure 4) also displayed a sharp drop after the first lag, highlighting the influence of an autoregressive term. This justifies selecting a 'p' parameter value of 1.

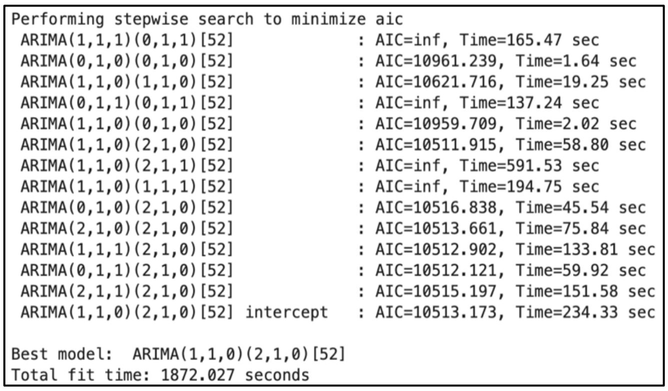
Given these observations from both plots, it became evident that the appropriate values for the autoregressive (p) and moving average (q) parameters are both set to 1, enhancing the SARIMA model's predictive capability for the dataset.

To determine the seasonal component of the SARIMA model, an initial exploratory analysis of the time series was conducted to detect potential periodic patterns. Given that stock market data often exhibits recurring behaviors influenced by weekly trading cycles, we hypothesized the presence of a weekly seasonality. Since the dataset is based on daily closing prices, a full seasonal cycle corresponds to approximately 52 weeks in a year, leading to the selection of s=52 as the seasonal period. And while the initial PACF analysis suggested limited visible seasonal lags, further empirical testing through grid search optimization indicated that setting the seasonal autoregressive order to P=2 provided a better balance between model fit and predictive stability. This choice likely captures latent seasonal dependencies that are not immediately apparent due to the inherent noise and volatility of financial time series data. The seasonal differencing order was set to D=1 to address residual seasonality, and Q=0 was maintained given the absence of significant autocorrelation at seasonal lags. The overall configuration, validated by minimizing the Akaike Information Criterion (AIC), confirmed seasonal_order = (2,1,0,52) as optimal for this dataset.

In the process of analyzing time series data, the determination of optimal parameters for models like SARIMA stands as a pivotal step. Not only do autocorrelation function (ACF) and partial autocorrelation function (PACF) plots offer insights into this, but also a grid search method proves instrumental. We operate this search with the stepwise_model to pinpoint the most suitable parameters.

Figure 5 shows the results of the stepwise selection: the parameters emerge as arima_order=(1,1,0) and seasonal_order=(2,1,0,52).

The search hinges on the Akaike Information Criterion (AIC) as its guiding metric. AIC quantifies the goodness of fit of a model, yet simultaneously penalizes models that veer into excessive complexity. Ideally, a model possesses a low AIC value, signifying a superior fit. Hence, the essence of the grid search revolves around identifying the parameter combination that minimizes the AIC, ensuring a harmonious balance between model fit and its inherent simplicity.



```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,1,1)[52]             : AIC=inf, Time=165.47 sec
 ARIMA(0,1,0)(0,1,0)[52]             : AIC=10961.239, Time=1.64 sec
 ARIMA(1,1,0)(1,1,0)[52]             : AIC=10621.716, Time=19.25 sec
 ARIMA(0,1,1)(0,1,1)[52]             : AIC=inf, Time=137.24 sec
 ARIMA(1,1,0)(0,1,0)[52]             : AIC=10959.709, Time=2.02 sec
 ARIMA(1,1,0)(2,1,0)[52]             : AIC=10511.915, Time=58.80 sec
 ARIMA(1,1,0)(2,1,1)[52]             : AIC=inf, Time=591.53 sec
 ARIMA(1,1,0)(1,1,1)[52]             : AIC=inf, Time=194.75 sec
 ARIMA(0,1,0)(2,1,0)[52]             : AIC=10516.838, Time=45.54 sec
 ARIMA(2,1,0)(2,1,0)[52]             : AIC=10513.661, Time=75.84 sec
 ARIMA(1,1,1)(2,1,0)[52]             : AIC=10512.902, Time=133.81 sec
 ARIMA(0,1,1)(2,1,0)[52]             : AIC=10512.121, Time=59.92 sec
 ARIMA(2,1,1)(2,1,0)[52]             : AIC=10515.197, Time=151.58 sec
 ARIMA(1,1,0)(2,1,0)[52] intercept   : AIC=10513.173, Time=234.33 sec

Best model:  ARIMA(1,1,0)(2,1,0)[52]
Total fit time: 1872.027 seconds
```

Figure 5. Stepwise search of SARIMA model parameters

The prediction process now commences using the SARIMA model with the chosen parameters. The forecast specifically targets the test sample for October 2022. To assess the model's accuracy and precision, several metrics come into play. Firstly, the RMSE, a common metric for accuracy, stands at 41.33. Additionally, the average stock price during this period is 643.11. By comparing the RMSE to this average, the error, expressed as a percentage of the mean, is calculated to be 6.4%. This percentage provides us a relative measure of the model's performance against the mean stock price.

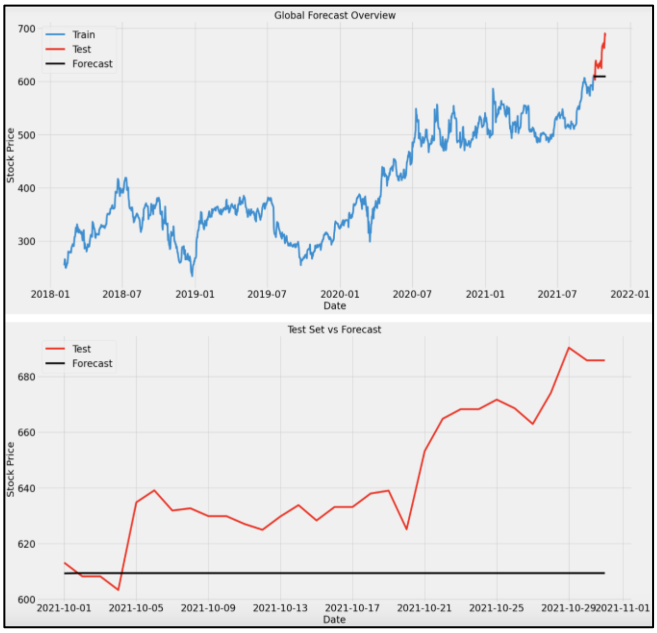


Figure 6. SARIMA forecast for Netflix stock price-October 2022

While the SARIMA prediction exhibits a commendable accuracy with only a 5.35% error relative to the average, we notice from the Figure 6 representation that the model fails to capture the intricate price fluctuations. Instead, it renders a forecast that appears largely linear in nature [14].

5.2 FBprophet

Transitioning to forecasting using Facebook Prophet, this model employs Fourier series to encapsulate both monthly and weekly seasonality (Figure 7).

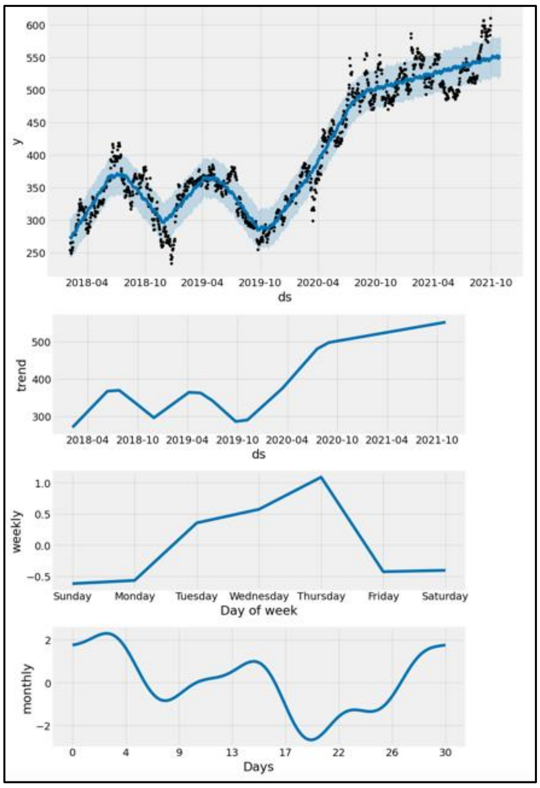


Figure 7. FBprophet trend and seasonality modelisation

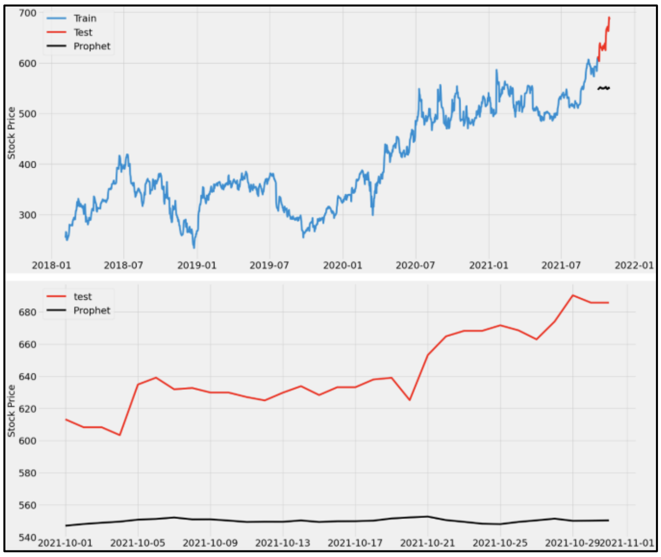


Figure 8. FBprophet forecast for Netflix stock price-October 2022

Upon examination of the prediction outcomes, as we can see on the Figure 8, Prophet's results offer a less linear projection in comparison to SARIMA. This is in alignment with Prophet's ability to capture more nuanced fluctuations. However, when considering the RMSE value of 96 and an error percentage of 14.92% relative to the average stock price of 643.11, it's evident that while Prophet tries to capture the stock's volatile tendencies [4, 15]. But it does so at the expense of overall accuracy.

5.3 XGBoost forecast

Shifting our attention to XGBoost, this powerful ensemble machine learning tool is typically utilized for regression and classification challenges. Unlike traditional time series models, XGBoost operates fundamentally as a regression model, but when tailored aptly, can excel in time series predictions.

A crucial facet of employing XGBoost in our analysis involves converting date data into external variables. To bolster prediction precision, we've integrated explanatory variables such as 'dayofweek', 'month', 'year', 'dayofyear', 'dayofmonth', and 'weekofyear'. This method allows the model to assimilate calendar-related nuances potentially influencing stock prices [5, 12, 16, 17].

We note that XGBoost model produced an RMSE of 52.27. With an average stock price of 643.11, the error percentage relative to this average is 8.12%. While its RMSE is competitive, XGBoost's performance is somewhat surpassed by SARIMA but is notably better than Prophet in terms of accuracy.

From Figure 9 inspection, XGBoost stands out as the most adept model in capturing the price variations over time. It accurately tracks the fluctuations and trends for the initial 21 days of the month. However, its predictions seem to deviate in the latter 10 days, indicating a potential overfitting or lack of adaptability to sudden changes.

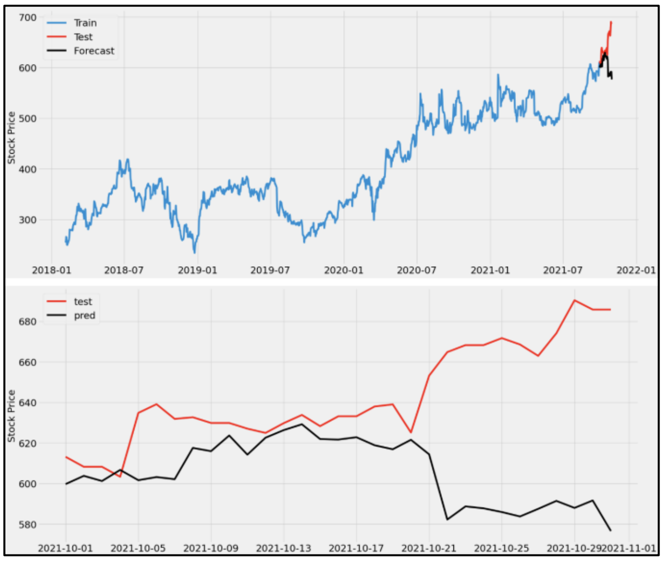


Figure 9. XGBoost forecast for Netflix stock price-October 2022

5.4 Hybrid approach

Given the notable precision of the SARIMA model coupled with its primary drawback of producing linear predictions that don't capture price fluctuations, and considering the performance of XGBoost in accurately tracking temporal price variations, we suggest a combined approach. In this paper, we propose a mixed-function model that blends SARIMA and XGBoost predictions. By judiciously adjusting the weights between the two models, we aim to harness the strengths of each while compensating for their respective weaknesses. This hybrid approach seeks to amalgamate the precision of SARIMA with the adaptability of XGBoost, offering a potentially more robust forecasting tool for stock price prediction.

This model can be represented mathematically using weighted averages:

Let's denote:

- **Psarima** as the prediction from the SARIMA model.
- **Pxgb** as the prediction from the XGBoost model.
- **Wsarima** as the weight for the SARIMA prediction.
- **Wxgb** as the weight for the XGBoost prediction.

$$\text{Phybrid} = \text{Wsarima} \times \text{Psarima} + \text{Wxgb} \times \text{Pxgb}$$

In our case, we choose the weights 0.6 for SARIMA and 0.4 for XGBoost. They are the result of the optimization process which, based on our testing and objectives, yielded the most advantageous balance between capturing fluctuations and maintaining predictive accuracy [18].

To justify the weighting scheme in the hybrid SARIMA-XGBoost model, an empirical evaluation was conducted with a focus beyond merely minimizing RMSE. While it is true that assigning a higher weight to SARIMA (up to 0.8) resulted in lower RMSE values, this came at the cost of producing overly smoothed forecasts that failed to capture the inherent volatility and short-term dynamics of stock price movements.

For instance, SARIMA alone achieved an RMSE of 41.33, but its predictions were largely linear and did not reflect the actual market fluctuations. The essence of financial forecasting is not only statistical accuracy but also the ability to track realistic price behavior, which is critical for decision-making in volatile environments.

Several weight combinations were tested to find a balance between stability and responsiveness. Table 1 illustrates this iterative process.

Table 1. Iterative evaluation of different weight combinations for the hybrid SARIMA-XGBoost model

Weight SARIMA	Weight XGBoost	RMSE	Observations
0.8	0.2	42.10	Very flat predictions, close to SARIMA only behavior
0.7	0.3	43.80	Improved dynamics but still overly conservative
0.6	0.4	45.35	Optimal trade-off: captures fluctuations with controlled error
0.5	0.5	46.12	Increased reactivity but signs of noise
0.4	0.6	47.80	Highly reactive, unstable forecasts, higher RMSE

For the hybrid model with weights of 0.6 for SARIMA and 0.4 for XGBoost, we obtain the RMSE=45.35 With an average stock price of 643.11, the error percentage relative to this average is 7.05%

Visually from Figure 10, the hybrid model demonstrates superior capability in capturing the stock price fluctuations over time compared to SARIMA alone. Furthermore, while the XGBoost model adeptly tracks these fluctuations, the hybrid approach commits fewer errors, offering a more balanced and precise forecasting tool.

While the hybrid model demonstrated promising results during the evaluation period, it is important to highlight that the testing was conducted exclusively over October 2022. This period was deliberately selected due to its notable market volatility, providing a relevant context to assess the responsiveness and adaptability of the forecasting models. However, this limited timeframe may not fully capture the diversity of market conditions encountered throughout the year. As such, further testing across extended periods and varying market scenarios would be essential to confirm the robustness and generalizability of the proposed hybrid approach [19, 20].

To strengthen model robustness, future work should include:

- Data augmentation techniques for financial time series [21-23],
- Transfer learning for domain adaptation [24],
- Regularization-aware hybrid architectures [25],
- Interpretable deep learning components like attention mechanisms [26].

Finally, advanced stationarity tests, such as KPSS or Phillips–Perron, could be explored alongside Dickey-Fuller to ensure robust preprocessing [27].

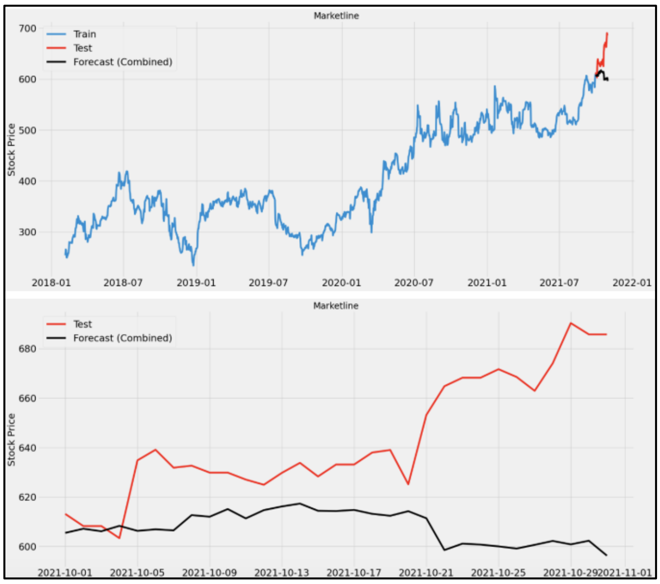


Figure 10. Hybride model (SARIMA/XGBoost) forecast for Netflix stock price-October 2022

6. CONCLUSION

In this study, we focused on forecasting Netflix's closing stock price using three well-known models: SARIMA, Prophet, and XGBoost. Each model displayed unique strengths and limitations in its predictive capabilities:

SARIMA showed strong accuracy, as indicated by its low RMSE, but struggled to capture the inherent fluctuations in stock prices over time, resulting in somewhat linear predictions.

Prophet, designed to account for the periodic nature of time series data, was less accurate than SARIMA but offered more nuanced predictions, better reflecting the stock's price movements and avoiding overly linear outcomes.

XGBoost, despite being traditionally a regression model, demonstrated notable strength in capturing the stock's price oscillations over the tested period. It excelled at tracking price fluctuations, particularly during the initial weeks of our test set.

The highlight of this paper, however, was the introduction and application of a hybrid model, which combined the strengths of SARIMA and XGBoost. By optimizing the weights of these models at 0.6 and 0.4 respectively, the hybrid approach outperformed the individual models. It not only delivered high accuracy but also effectively captured the stock's price variations, striking a balance between precision and trend representation.

Moreover, the scope of this study was constrained by a short evaluation window, focusing solely on a single month of data. Although this allowed for a focused comparison under volatile conditions, future work should involve testing the hybrid model across longer and more diverse timeframes to ensure its applicability in different market environments. Expanding the evaluation period will be crucial to validate the consistency and generalizability of the forecasting performance observed in this study.

Lastly, while RMSE was a reliable measure of model performance in this study, it doesn't tell the whole story. Finding a more comprehensive metric that evaluates both accuracy and the ability to capture trends remains an open area for future research. Pursuing this could bring us closer to developing a model that is both precise and better reflects the true behavior of time series data.

REFERENCES

[1] Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M. (2015). Time Series Analysis: Forecasting and Control. John Wiley & Sons.

[2] Lütkepohl, H. (2005). New Introduction to Multiple Time Series Analysis. Springer Science & Business Media.

[3] Johansen, S. (1995). Likelihood-based Inference in Cointegrated Vector Autoregressive Models. OUP Oxford.

[4] Taylor, S.J., Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1): 37-45. https://doi.org/10.1080/00031305.2017.1380080

[5] Chen, T., Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785-794. https://doi.org/10.1145/2939672.2939785

[6] Fischer, T., Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2): 654-669. https://doi.org/10.1016/j.ejor.2017.11.054

[7] Nelson, D.M., Pereira, A.C.M., de Oliveira, R.A. (2017). Stock market's price movement prediction with LSTM neural networks. In 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, pp. 1419-1426. https://doi.org/10.1109/IJCNN.2017.7966019

[8] Hyndman, R.J., Koehler, A.B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4): 679-688. https://doi.org/10.1016/j.ijforecast.2006.03.001

[9] Armstrong, J.S., Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. International Journal of Forecasting, 8(1): 69-80. https://doi.org/10.1016/0169-2070(92)90008-W

[10] Spiliotis, E., Makridakis, S. (2021). Evaluation of time series forecasting models for stock market applications. PLOS One, 16(4): e0250031. https://doi.org/10.1371/journal.pone.0250031

[11] Abadie, J., Meslier, F. (1979). Etude de l'utilisation des modèles ARIMA pour la prévision à très court terme de l'énergie journalière produite par Electricité de France. RAIRO-Operations Research, 13(1): 37-54.

[12] Demir, S., Mincev, K., Kok, K., Paterakis, N.G. (2019). Introducing technical indicators to electricity price forecasting: A feature engineering study for linear, ensemble, and deep machine learning models. Applied Sciences, 10(1): 255. https://doi.org/10.3390/app10010255

[13] Kumar, V. (2021). Statistical Tests to Check Stationarity in Time Series. Analytics Vidhya.

[14] Smyl, S., Kuber, K. (2016). Data preprocessing and augmentation for multiple short time series forecasting with recurrent neural networks. In 36th International Symposium on Forecasting.

[15] Bergmeir, C., Hyndman, R., Benítez, J. (2016). Bagging exponential smoothing methods using STL decomposition and Box-Cox transformation. International Journal of Forecasting, 32: 303-312, https://doi.org/10.1016/j.ijforecast.2015.07.002

[16] Lago, J., Marcjasz, G., De Schutter, B., Weron, R. (2021). Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. Applied Energy, 293: 116983. https://doi.org/10.1016/j.apenergy.2021.116983

[17] Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T. (2012). Learning from Data. New York: AMLBook. USA.

[18] Smyl, S. (2020). A hybrid forecasting approach using machine learning feature engineering. International Journal of Forecasting, 36(1): 86-98. https://doi.org/10.1016/j.ijforecast.2019.03.017

[19] Gunduz, S., Ugurlu, U., Oksuz, I. (2020). Transfer learning for electricity price forecasting. arXiv preprint arXiv:2007.03762. https://doi.org/10.48550/arXiv.2007.03762

[20] Hmamouche, Y. (2018). Prédiction des séries temporelles larges. Doctoral dissertation, AMU-Aix Marseille Université.

[21] DeVries, T., Taylor, G.W. (2017). Dataset augmentation in feature space. arXiv preprint arXiv:1702.05538. https://doi.org/10.48550/arXiv.1702.05538

[22] Jorge, J., Vieco, J., Paredes, R., Sánchez, J.A., Benedí, J.M. (2018). Empirical evaluation of variational autoencoders for data augmentation. In VISIGRAPP (5: VISAPP), pp. 96-104. https://doi.org/10.5220/0006618600960104

[23] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing, 321: 321-331. https://doi.org/10.1016/j.neucom.2018.09.013

[24] Luo, Y., Lu, B. (2018). EEG data augmentation for emotion recognition using a conditional Wasserstein GAN. In 40th IEEE EMBC, pp. 2535-2538, https://doi.org/10.1109/EMBC.2018.8512865

[25] Lefieux, V. (2007). Modèles semi-paramétriques appliqués à la prévision des séries temporelles. Cas de la consommation d'électricité. Doctoral dissertation, Université Rennes 2.

[26] Hyndman, R.J., Athanasopoulos, G. (2018). Forecasting:

Principles and Practice. OTexts.

[27] Demir, S., Mincev, K., Kok, K., Paterakis, N.G. (2021). Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting. Applied Energy, 304: 117695. https://doi.org/10.1016/j.apenergy.2021.117695