ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/sandeep-yadav/402

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 1 & Iss: 1

Research Article

A Comparative Study of ARIMA, Prophet and LSTM for Time Series Prediction

Sandeep Yadav*

Citation: Yadav S. A Comparative Study of ARIMA, Prophet and LSTM for Time Series Prediction. *J Artif Intell Mach Learn & Data Sci 2022*, 1(1), 1813-1816. DOI: doi.org/10.51219/JAIMLD/sandeep-yadav/402

Received: 02 February, 2022; **Accepted:** 26 February, 2022; **Published:** 28 February, 2022

***Corresponding author:** Sandeep Yadav, Silicon Valley Bank, USA, E-mail: sandeep.yadav@asu.edu

ABSTRACT

Time series forecasting is critical in domains such as finance, weather prediction and demand planning, where accurate predictions drive informed decision-making. This study provides a comparative analysis of three popular time series forecasting models: Autoregressive Integrated Moving Average (ARIMA), Facebook Prophet and Long Short-Term Memory (LSTM) neural networks. These models, representing statistical, automated trend analysis and deep learning approaches, offer distinct advantages based on dataset characteristics.

ARIMA, a traditional statistical model, excels in short-term forecasting with stationary data but requires manual tuning and struggles with non-linear complexities. Facebook Prophet simplifies forecasting by handling seasonality and missing data automatically, making it well-suited for business applications. LSTM, a deep learning model, captures long-term dependencies and non-linear patterns in time series data, often outperforming traditional models in accuracy but at the cost of higher computational requirements.

The models are evaluated using finance dataset, with metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Results reveal that ARIMA is computationally efficient for simple patterns, Prophet excels in seasonal datasets and LSTM achieves the highest accuracy for complex, long-term patterns. This study offers a practical guide for selecting time series forecasting models based on data complexity, computational resources and application needs.

Keywords: Time Series Forecasting, ARIMA, Facebook Prophet, Long Short-Term Memory (LSTM), Statistical Models, Deep Learning, Predictive Analytics, Seasonality Detection, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Financial Time Series

1. Introduction

Time series forecasting is a fundamental problem in various domains such as finance, weather forecasting, supply chain management and economics. Accurate forecasts of future events based on historical data play a critical role in decision-making processes, allowing businesses and organizations to optimize operations, mitigate risks and allocate resources more effectively. Given the widespread importance of time series predictions, selecting the most appropriate forecasting model is crucial for achieving reliable and actionable insights.

Over the years, numerous models have been developed to address the complexities of time series data, which often includes trends, seasonality and irregular fluctuations. Among the most widely used approaches are Autoregressive Integrated Moving Average (ARIMA), Prophet and Long Short-Term Memory (LSTM) neural networks. These models represent

three distinct methodologies: traditional statistical modeling (ARIMA), automated and robust trend analysis (Prophet) and deep learning-based sequence modeling (LSTM).

ARIMA is a statistical approach that has been a standard tool for time series analysis for decades. It is particularly useful for stationary time series and excels in short-term forecasting. However, ARIMA requires significant manual tuning and may struggle with more complex, non-linear data. Prophet, developed by Facebook, simplifies the forecasting process by automatically detecting trends and seasonality, making it highly effective for business applications with missing data or outliers. LSTM, a type of recurrent neural network, is capable of learning long-term dependencies and handling non-linear patterns in time series, offering superior predictive performance in complex datasets.

This paper presents a comprehensive comparison of these three models, evaluating their strengths and weaknesses. By analyzing their performance in terms of accuracy, interpretability and computational efficiency, this study aims to provide insights into the most suitable forecasting model for different types of time series data.

2. Literature Review

Time series forecasting has been extensively studied across multiple fields, with a wide range of models developed to address the complexities of predicting future values based on past observations. This section provides an overview of three widely used models-ARIMA, Prophet and LSTM-and examines existing literature on their applications and performance in time series forecasting.

A. ARIMA (Autoregressive Integrated Moving Average)

The Autoregressive Integrated Moving Average (ARIMA) model, first introduced by Box and Jenkins (1970), is one of the most widely used statistical methods for time series forecasting. ARIMA is a combination of three components:

- Autoregression (AR): Uses the relationship between an observation and several lagged observations.
- Integrated (I): Involves differencing the observations to make the time series stationary.
- Moving Average (MA): Models the relationship between an observation and the residual errors from past observations.

ARIMA is well-suited for univariate time series data that can be made stationary through differencing. Several studies have demonstrated its effectiveness in short-term forecasting. For instance, Pai and Lin (2005) found that ARIMA performed well in the financial time series, offering accurate forecasts for stock prices and interest rates. Similarly, in demand forecasting, the model has been used extensively due to its simplicity and reliability[1].

However, ARIMA's limitations include its inability to handle non-linear patterns and its requirement for manual tuning, such as selecting the order of AR, I and MA components. In complex datasets with seasonal or irregular patterns, ARIMA often struggles, especially when the time series is non-stationary[4]. This limitation has led to the exploration of more advanced methods such as Prophet and LSTM.

B. Prophet

Prophet, developed by Facebook's research team[2], is designed for business forecasting, particularly for datasets with daily or weekly observations that exhibit strong seasonal effects. Prophet uses an additive model that incorporates the following components:

- Trend: Captures the long-term increase or decrease in the data.
- Seasonality: Models periodic changes in the data.
- Holidays and Special Events: Allows for external events to be added as regressors.

One of Prophet's key advantages is its ability to automatically handle missing data and outliers, which are common in business time series data. The model also accommodates changes in seasonality and trends, making it flexible for real-world applications. Prophet has been used successfully in forecasting tasks such as demand planning, financial forecasting and resource allocation (Sani & Maharaj, 2020).

In comparison to ARIMA, Prophet does not require the data to be stationary, making it easier to implement in non-stationary time series. Studies demonstrate that Prophet performs well in cases with clear seasonal patterns and trends, although it may not perform as effectively in datasets with short-term volatility or high-frequency data.

C. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM), a variant of recurrent neural networks (RNNs), was introduced by Hochreiter and Schmidhuber (1997) to address the limitations of traditional RNNs, particularly in learning long-term dependencies. LSTM models can retain information over longer sequences, making them highly effective for time series forecasting, especially in datasets with complex patterns and non-linear relationships.

LSTM's architecture includes three main gates-input, forget and output gates-that regulate the flow of information through the network. This unique structure enables LSTMs to capture both short- and long-term dependencies, which is particularly valuable for time series data with irregular trends or cyclical patterns[16].

In recent years, LSTM has gained popularity for time series prediction across various domains, including energy forecasting, stock market prediction and weather forecasting. However, LSTMs require large datasets and significant computational resources for training and their black-box nature makes interpretability a challenge.

The literature demonstrates that ARIMA, Prophet and LSTM each have their strengths and limitations in time series forecasting. ARIMA remains a popular choice for simpler, short-term forecasts, especially in stationary datasets. Prophet excels in scenarios involving business-related time series with strong seasonal components, offering ease of use and flexibility. LSTM, on the other hand, stands out in its ability to model complex, non-linear patterns and long-term dependencies, making it suitable for more intricate time series tasks, though it requires significant data and computational power.

As the demand for accurate time series predictions continues to grow, this comparative study aims to provide further insights

into the performance of these models across different datasets and forecasting challenges.

3. Proposed Methodology

This study systematically evaluates and compares the performance of ARIMA, Prophet and LSTM models on diverse time series datasets. The methodology is designed to analyze each model's strengths and weaknesses in handling various characteristics of time series data, such as seasonality, trends and non-linear dependencies. The proposed framework is divided into the following key steps:

A. Dataset Selection and Preprocessing

To ensure a comprehensive evaluation, the methodology uses Stock Prices dataset to analyze financial data with high volatility.

Data Normalization:

Time series values are scaled to the range [0, 1] for models requiring neural networks to improve convergence.

$$X_{scaled} = \frac{X - min(X)}{max(X) - min(X)}$$

Stationarity Check:

For ARIMA, the augmented Dickey-Fuller (ADF) test is used to verify stationarity. If the data is non-stationary, differencing is applied:

$$X_t' = X_t - X_{t-1}$$

Train-Test Split:

Each dataset is split into a training set (80%) and a testing set (20%).

B. ARIMA Implementation

The ARIMA model is implemented with parameter tuning for (p, d, q). Autoregressive (p): Number of lag observations. Integrated (d): Differencing order to achieve stationarity and Moving Average (q): Order of the moving average.

Equation:

$$X_t = c + \emptyset_1 X_{t-1} + \emptyset_2 X_{t-2} + \cdots + \emptyset_p X_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

Where X_t time series value at time t. c constant. Φ autoregressive coefficients. Θ is moving average coefficients. hite noise. Use ADF to determine stationarity and apply differencing if necessary. Tune p, d, q using AIC/BIC criteria then fit the ARIMA model and forecast future values.

C. Prophet Implementation

Prophet uses an additive model to decompose the time series into trend, seasonality and holiday effects:

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

Where $g(t)$ is trend component (piecewise linear or logistic growth), $s(t)$ seasonal component (modeled using Fourier series) and $h(t)$ is holiday effects.

Define holidays or specia l events relevant to the dataset (optional). Then we must fit the Prophet model with automatic detection of trends and s easonality. Forecast future values using the model's in-bui l t functionality. The Prophet model automatically handles mi s sing data and anomalies. Also, it decomposes the forecast into interpretable components.

D. LSTM Implementation

Long Short-Term Memory (LSTM) is implemented to capture long-term dependencies and non-linear patterns. The LSTM architecture includes a forget gate which removes irrelevant information. Input Gate updates the cell state with new information and an output gate which produces the output at each time step.

Equations:

1. Forget Gate: $f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)$
2. Input Gate: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ & $\tilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C)$
3. Cell State Update: $C_t = f_t.C_{t-1} + i_t.\tilde{C}_t$
4. Output Gate: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ & $h_t = o_t \cdot tanh(C_t)$

Prepare input sequences using a sliding window approach. Normalize data for faster convergence then design the LSTM architecture with input, hidden and output layers. Train the model using backpropagation through time (BPTT). Predict future values and evaluate performance.

The models are evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) & Mean Absolute Percentage Error (MAPE).

4. Experimental Setup & Evaluation

The experimental setup involves evaluating the performance of three time series fore c a sting models-ARIMA, Prophet and LSTM-using simulated stock price data. The evaluation focuses on metrics such as MAE (M e a n Absolute Error) and RMSE (Root Mean Squared Error), alongside visualizations to compare predicted and actual values.

4.1. Dataset and Correlation Analysis

The dataset consists of 5 0 0 data points simulating a time series of stock prices wi t h seasonal and noise components. Performed feature engineering to improve the model results by adding numerical features representing the day of the week and the month to capture peri o d ic trends. Also introduced lagged variables (1-day and 7-day lags) to incorporate short-term and weekly dependencies. Added 7-day rolling mean and standard deviation to capture loca l trends and volatility. The heatmap (Figure 1) visualizes the correlation between engineered features and stock prices within the raining dataset. Features like lagged values and rolling means show strong correlations, confirming their predictive potential.

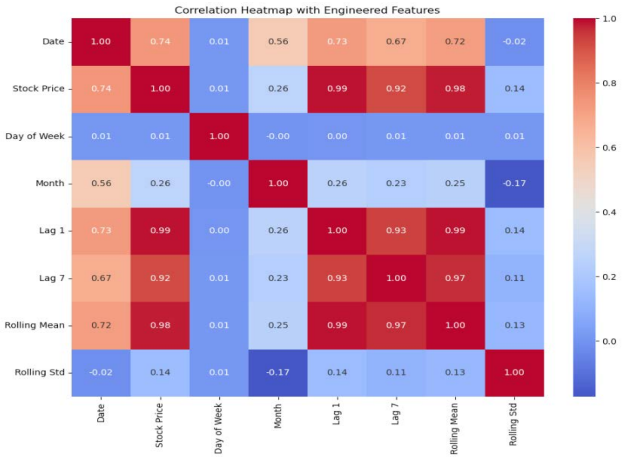

Figure 1: Correlation Heatmap with Engineered Features.

4.2. Line Plot of Actual vs. Predicted Values

(Figure 2) displays the comparison between actual stock prices and model predictions over time. LSTM predictions closely follow the actual values, outperforming ARIMA and Prophet, especially during non-linear trends. ARIMA and Prophet show deviations in regions with irregular trends.

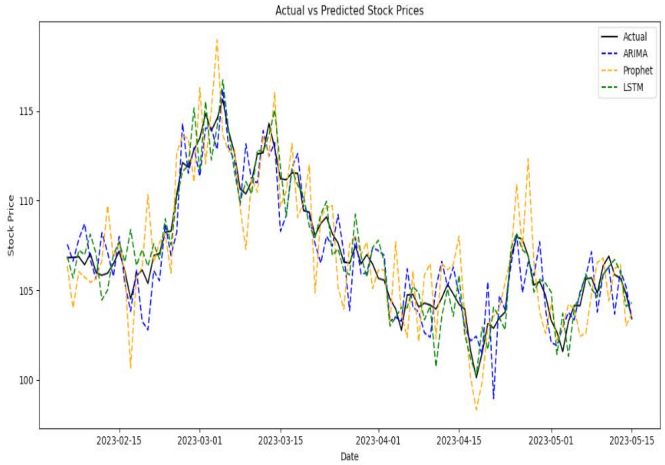

Figure 2: Actual Vs Predicted Stock Prices.

Below Bar Charts shows comparison of MAE and RMSE for each model. LSTM achieves the lowest error metrics, demonstrating superior accuracy. Prophet shows higher errors due to challenges in capturing short-term fluctuations. After feature engineering LSTM demonstrates the largest improvement.

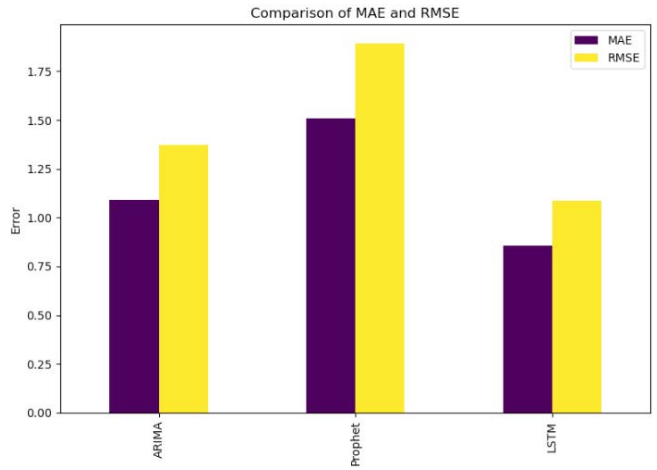

Figure 3: MAE & RMSE Comparison.

4.3. Implementation and Infrastructure

The study leveraged a combination of computational tools and cloud-based resources to simulate, train and evaluate models. Python was the primary programming language, utilizing libraries such as NumPy, pandas and scikit-learn for data processing and evaluation. Visualizations were created with Matplotlib and Seaborn to provide clear insights into model performance. Model implementations, including ARIMA, Prophet and LSTM, utilized specialized packages such as statsmodels, fbprophet and TensorFlow. The experiments were conducted on a mid-tier cloud-based environment equipped with GPUs for deep learning models, ensuring efficient training of the LSTM architecture. This infrastructure supported the integration of advanced feature engineering, data preprocessing and model evaluation, enabling scalable analysis of stock market data and rapid iteration for performance tuning.

5. Conclusion

This study evaluated ARIMA, Prophet and LSTM models for stock market forecasting, incorporating advanced feature engineering techniques to enhance model performance. The results demonstrate that LSTM outperforms ARIMA and Prophet in capturing complex, non-linear patterns and delivering accurate predictions. While ARIMA remains effective for linear trends and Prophet excels in handling seasonality, they are limited in their ability to adapt to high volatility and intricate dependencies in stock prices. Feature engineering, such as adding lagged features, rolling statistics and time-based variables, significantly improved the predictive accuracy of all models. These findings underscore the importance of selecting models and features that align with the complexity of the data and the forecasting objectives.

6. References

1. Box GEP, Jenkins GM. Time Series Analysis: Forecasting and Control. Holden-Day, 1976.
2. https://www.tandfonline.com/doi/full/10.1080/00031305.2017.1380080
3. https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext
4. https://otexts.com/fpp2/
5. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889
6. https://www.sciencedirect.com/science/article/abs/pii/S0925231201007020?via%3Dihub
7. https://www.deeplearningbook.org/
8. https://arxiv.org/abs/1705.07874
9. https://link.springer.com/book/10.1007/978-3-319-52452-8
10. Rizwan MS, Ahmad G, Ashraf D. Systemic risk: The impact of COVID-19 on the banking sector. Journal of Banking & Finance, 2020;117:105803.
11. https://arxiv.org/abs/1704.02971
12. https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full
13. https://arxiv.org/abs/1506.04214
14. Chollet F. Deep Learning with Python. Manning Publications, 2018.
15. https://scikit-learn.org/
16. Brownlee J. Introduction to Time Series Forecasting with Python. Machine Learning Mastery, 2017.