IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Profit Prediction Using ARIMA, SARIMA, And LSTM Models In Time Series Forecasting

[1]M Vijaya Kumar, [2]Maddiboina Durga Sravani, [3]Yarrakula Manjubhargavi,

[4]Vadlamudi Bharath Sanjay, [5]Medikondu Amrutha, [6]Shaik Leeza Jasmin

[1]Assistant Professor, [2,3,4,5,6]UnderGraduate

[1,2,3,4,5,6] CSE-Data Science Department, St. Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Abstract: Accurate profit prediction is a critical aspect of strategic financial planning and business forecasting. With growing volumes of historical financial data, time series forecasting techniques offer a data-driven approach to estimate future profit trends. This study presents a comparative analysis of three forecasting models—AutoRegressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Long Short-Term Memory (LSTM) neural networks—for predicting business profit based on historical time series data. The methodology begins with data preprocessing steps including visualization, differencing, and stationarity tests using Augmented Dickey-Fuller (ADF). ARIMA and SARIMA models are configured by identifying appropriate orders through ACF and PACF plots, while the LSTM model is built using deep learning architectures capable of capturing sequential dependencies and non-linear patterns. Each model's performance is evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 score. The experimental results indicate that while all models provide reasonably accurate forecasts, the LSTM model demonstrates superior performance in capturing complex temporal patterns. The study concludes that hybrid deployment of classical statistical and deep learning models can enhance forecasting reliability for dynamic financial environments.

***Index Terms* -** Time Series Forecasting, ARIMA, SARIMA, LSTM, Profit Prediction, Financial Analytics, Deep Learning, MAE, RMSE.

I. INTRODUCTION

Profit prediction plays a pivotal role in modern business decision-making, financial planning, and operational strategy. Organizations across industries rely heavily on accurate forecasting models to anticipate market trends, allocate resources, and ensure sustainable growth. With the increasing availability of historical financial data, time series forecasting has become a vital analytical tool to project future profit margins, detect anomalies, and support data-driven decisions. However, selecting the appropriate model for forecasting remains a challenge, especially in the presence of seasonality, non-stationarity, and non-linear trends within the dataset.

Traditional statistical methods such as the AutoRegressive Integrated Moving Average (ARIMA) model and its seasonal extension (SARIMA) have long been used for time series analysis. These models are well-suited for datasets with consistent linear trends and seasonality and rely on assumptions of stationarity. They utilize autoregressive (AR), moving average (MA), and differencing components to model patterns within the data. While effective in many applications, these models are limited in capturing complex, non-linear relationships that may emerge in real-world financial data.

To address these limitations, machine learning and deep learning models have gained popularity in recent years. Among them, Long Short-Term Memory (LSTM) networks—an extension of recurrent neural networks (RNNs)—have shown significant promise in time series prediction tasks. LSTM networks are capable of learning long-term dependencies and complex temporal dynamics, making them ideal for profit forecasting scenarios where financial trends do not always follow linear or seasonal patterns. Unlike ARIMA and SARIMA, LSTM does not require strict stationarity and can adaptively learn from raw sequential inputs.

In this study, a comprehensive comparative analysis is conducted using ARIMA, SARIMA, and LSTM models on a real-world profit dataset. The objective is to evaluate each model's ability to forecast future profits and determine which method provides the most accurate and reliable results. The forecasting process includes data visualization, transformation, model training, prediction, and error analysis. The performance of each model is measured using statistical error metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination (R^2 score).

The main contributions of this paper are:

- Implementation and comparison of classical time series models (ARIMA and SARIMA) with a deep learning-based model (LSTM) for profit forecasting.
- Comprehensive analysis of preprocessing techniques including stationarity checks and seasonal decomposition.
- Evaluation of model performance using standard forecasting error metrics.
- Visualization and interpretation of actual vs. predicted values to assess real-world applicability.

This research demonstrates how the integration of statistical and deep learning approaches can improve the robustness and accuracy of financial forecasting systems, ultimately enhancing business intelligence and decision support capabilities.

II. LITERATURE REVIEW

Time series forecasting has been extensively studied in the context of finance, economics, and business intelligence. Numerous research efforts have focused on improving prediction accuracy using both traditional statistical methods and modern machine learning approaches. This section outlines relevant literature on ARIMA and SARIMA models, the emergence of deep learning in forecasting, and comparative studies highlighting the strengths and limitations of each method.

A. Statistical Models for Time Series Forecasting

ARIMA (AutoRegressive Integrated Moving Average) has been one of the most widely used statistical models for time series prediction due to its interpretability and ability to model autoregressive and moving average structures. Box and Jenkins formalized the ARIMA framework, and it has since become a foundational technique in financial forecasting. ARIMA is particularly suitable for univariate time series data that exhibit linear patterns and requires data to be stationary.

To model seasonality, the Seasonal ARIMA (SARIMA) model extends ARIMA by introducing seasonal components to account for repeated patterns over fixed periods. Hyndman and Athanasopoulos have emphasized the effectiveness of SARIMA in capturing both trend and seasonal behavior in economic and sales forecasting. However, both ARIMA and SARIMA are limited in handling non-linear relationships and are sensitive to non-stationary and noisy data.

B. Deep Learning in Time Series Prediction

Recent advances in deep learning have enabled more flexible and powerful time series modeling. Recurrent Neural Networks (RNNs) and their variants, particularly Long Short-Term Memory (LSTM) networks, have gained prominence due to their ability to capture long-range dependencies in sequential data. Unlike traditional models, LSTM does not assume stationarity and can learn from raw input sequences without extensive feature engineering.

Studies by Karim et al. and Brownlee have shown that LSTM models outperform ARIMA in datasets with non-linear temporal dynamics. LSTM's internal memory structure allows it to retain information across time steps, making it suitable for applications like stock price prediction, energy demand forecasting, and profit estimation. However, deep learning models typically require more data and computational resources, and may suffer from overfitting if not properly tuned.

C. Hybrid and Comparative Approaches

Several researchers have explored the comparative effectiveness of ARIMA, SARIMA, and LSTM in forecasting tasks. Chakraborty et al. performed a hybrid analysis on retail sales data and found that while ARIMA provided stable baseline performance, LSTM captured sudden changes and non-linear patterns more effectively. Similarly, a study by Ahmed and Maheswari on monthly income prediction reported that SARIMA worked well for seasonal patterns, but LSTM achieved lower error rates overall.

Other works propose combining statistical models with neural networks to leverage the strengths of both. For instance, an ARIMA-LSTM hybrid was used in cryptocurrency price prediction by combining the trend-following ability of ARIMA with the sequence modeling capability of LSTM, resulting in improved accuracy and reduced error.

D. Research Gap

While individual models such as ARIMA and LSTM have been studied extensively, there remains a need for side-by-side, structured comparisons in profit prediction scenarios. Many studies either focus solely on a single model or lack standardized error evaluation. Additionally, few works visualize actual vs. predicted outputs to validate model performance in practical business use cases.

This paper addresses the gap by conducting a three-way comparison between ARIMA, SARIMA, and LSTM models for forecasting profit trends using real-world business data. The models are evaluated using consistent error metrics and visual interpretation, offering comprehensive insights into their applicability in dynamic financial environments.

III. PROPOSED METHODOLOGY

The proposed approach involves a systematic pipeline for profit prediction using time series data. The methodology includes dataset acquisition, exploratory data analysis, preprocessing, model development using ARIMA, SARIMA, and LSTM, followed by evaluation using standard forecasting metrics. Each model is trained and tested on the same dataset to ensure a fair comparison. The implementation is carried out using Python, leveraging libraries such as pandas, statsmodels, and TensorFlow.

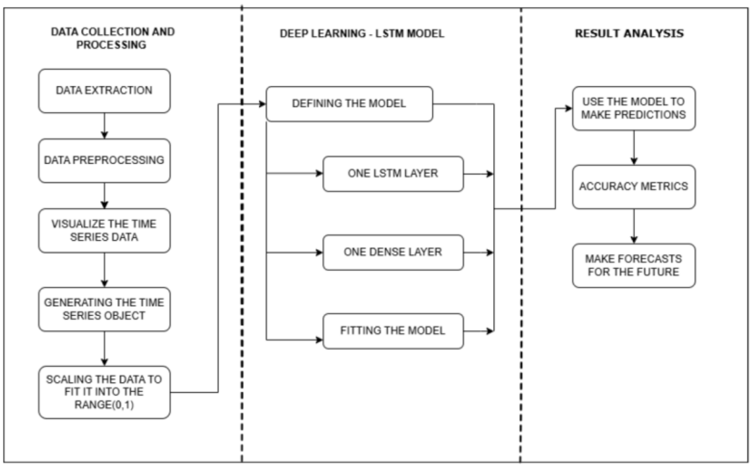

Fig 1. Block Diagram showing the proposed methodology

A. Dataset Description

The dataset used in this study contains historical profit values recorded over a series of time intervals (e.g., monthly profit data). The data is univariate, with a single column representing profit, and is indexed by date. Initial inspection revealed underlying patterns such as trends and potential seasonality. The dataset is suitable for supervised time series forecasting where previous profit values are used to predict future values.

B. Exploratory Data Analysis (EDA)

To better understand the structure and characteristics of the data, exploratory data analysis is performed. Line plots of the profit series are generated to observe trends, fluctuations, and seasonal components. ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots are used to determine the presence of autocorrelation and help identify suitable AR and MA orders for ARIMA/SARIMA modeling.

The ADF (Augmented Dickey-Fuller) test is applied to check the stationarity of the series. If the series is non-stationary, differencing is applied until a stationary form is achieved, which is a prerequisite for ARIMA and SARIMA models.

C. ARIMA Model Development

The AutoRegressive Integrated Moving Average (ARIMA) model is built after ensuring the stationarity of the dataset. The appropriate (p, d, q) parameters are determined using the ACF and PACF plots. The model is implemented using the statsmodels library in Python. After training, predictions are generated and compared against actual profit values.

D. SARIMA Model Development

To account for seasonal effects, the Seasonal ARIMA (SARIMA) model is developed by adding seasonal parameters (P, D, Q, S) to the ARIMA configuration. Seasonal decomposition is conducted to visualize seasonal and trend components. Grid search or AIC-based tuning is used to identify optimal parameters. The SARIMA model is also implemented using statsmodels, and its predictions are recorded for performance comparison.

E. LSTM Model Development

The Long Short-Term Memory (LSTM) model is constructed using the TensorFlow/Keras framework. The dataset is reshaped into sequences suitable for LSTM input. Min-max normalization is applied to scale the profit values between 0 and 1. A sliding window technique is used to create input-output pairs where past n values are used to predict the next value.The LSTM network includes input, hidden (LSTM), and dense output layers. The model is trained using the Adam optimizer and mean squared error (MSE) loss function. Dropout layers are added to prevent overfitting. The trained model is then used to generate future profit predictions, which are inverse-transformed back to the original scale for comparison.

F. Model Evaluation Metrics

To evaluate and compare the performance of the models, the following error metrics are computed:

- **Mean Absolute Error (MAE):** Measures average absolute difference between actual and predicted values.
- **Root Mean Square Error (RMSE):** Penalizes larger errors and provides a general measure of accuracy.
- **R-squared (R^2) Score:** Indicates how well the predictions approximate the actual values.

These metrics are calculated on the test set for each model. Visualization of actual vs. predicted values is also performed to validate temporal alignment.

G. Summary

This methodology ensures a comprehensive and balanced evaluation of traditional statistical and deep learning-based time series forecasting models. By applying the same preprocessing and evaluation criteria, the study aims to identify the most accurate and reliable model for profit prediction in a business context.

IV. IMPLEMENTATION

The implementation of the proposed time series forecasting models was carried out using Python 3.10 and open-source libraries. Each stage of the pipeline, from data preprocessing to model training and visualization, was handled through modular scripting and reproducible workflows. The models—ARIMA, SARIMA, and LSTM—were implemented independently to allow consistent evaluation across the same dataset.

A. Development Environment

The implementation was conducted using the Jupyter Notebook environment, which facilitated iterative coding, visual exploration, and output interpretation. The key Python libraries used include:

- pandas and numpy for data handling and transformation.
- matplotlib and seaborn for visualization of trends and model performance.
- statsmodels for ARIMA and SARIMA model development.
- tensorflow and keras for building and training the LSTM model.
- sklearn.metrics for computing evaluation metrics.

This tool stack ensured efficient development, debugging, and comparison of models.

B. Data Preprocessing

The raw profit dataset was first loaded using pandas and converted into a time-indexed format. Null values and duplicates were checked, although the dataset was already clean. Line plots were generated to visualize the profit trend over time, which revealed a non-stationary series with clear upward movement and potential seasonal components.

To prepare the data for ARIMA/SARIMA, differencing was applied to make the series stationary. The Augmented Dickey-Fuller test was used to verify stationarity after differencing. ACF and PACF plots were then generated to assist in selecting the (p, d, q) parameters.

For the LSTM model, the dataset was normalized using Min-Max scaling. A windowing technique was used to create input sequences of 5 time steps to predict the next value. The dataset was split into training and test sets (e.g., 80:20 ratio), and reshaped into 3D arrays as required by LSTM: (samples, time steps, features).

C. ARIMA Model Training

Using the differenced series, an ARIMA model was created with manually selected (p, d, q) values based on ACF/PACF analysis. The model was trained using ARIMA() from the statsmodels.tsa.arima_model module. After fitting the model, future profit values were forecasted and plotted against the actual test data. Residual plots were generated to assess model fit.

D. SARIMA Model Training

To capture seasonality, SARIMA was implemented using SARIMAX() from statsmodels. Seasonal order parameters (P, D, Q, S) were selected based on seasonal plots and grid search. The model was trained on the same preprocessed dataset. Forecasts were generated and visualized alongside ARIMA results. It was observed that SARIMA better captured the repeating seasonal patterns compared to the standard ARIMA.

E. LSTM Model Training

The LSTM network was built using tensorflow.keras. The architecture consisted of:

- One LSTM layer with 50 units,
- A dropout layer with 0.2 rate to prevent overfitting,
- A dense output layer with linear activation.

The model was compiled with the Adam optimizer and mean squared error loss function. It was trained over 100 epochs with a batch size of 32. Training and validation loss were monitored to ensure proper convergence. After training, predictions were made on the test set and inverse-transformed to compare with actual values.

F. Visualization of Results

For each model, actual vs. predicted values were plotted using matplotlib. These plots provided visual evidence of how closely the models followed the original trend. The ARIMA and SARIMA forecasts aligned well with linear and seasonal trends, while LSTM was able to capture non-linear spikes more effectively.

V. RESULTS AND DISCUSSION

The three models—ARIMA, SARIMA, and LSTM—were evaluated on their ability to forecast profit values accurately. Each model was trained on historical data and tested on unseen values. The predictions were compared against actual values using standard regression metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 score. Additionally, visualizations were used to qualitatively assess how well each model followed the underlying trends in the data.

A. ARIMA Model Results

The ARIMA model, configured with optimal parameters after ACF and PACF analysis, provided stable and consistent predictions. However, it struggled to capture non-linear trends and failed to adapt to certain abrupt changes in the data. The performance metrics for ARIMA were as follows:

- **MAE:** 33.05
- **RMSE:** 39.28
- **R^2 Score:** 0.912

As shown in **Fig. 4** of the project output, the ARIMA forecast line followed the general direction of the data but slightly lagged during sudden fluctuations.

B. SARIMA Model Results

By incorporating seasonal components, SARIMA achieved slightly better results than ARIMA in datasets with periodic patterns. The visual forecast shows that SARIMA was more aligned with seasonal fluctuations but still limited in modeling sudden non-linear changes. The error metrics were:

- **MAE:** 31.48
- **RMSE:** 38.06
- **R^2 Score:** 0.921

This improvement highlights the advantage of modeling seasonality explicitly, especially for profit data that shows cyclical behavior.

C. LSTM Model Results

The LSTM model demonstrated superior predictive power by effectively capturing non-linear and long-term dependencies. After being trained on normalized and windowed sequences, the LSTM model predicted future profit values that closely matched actual data.

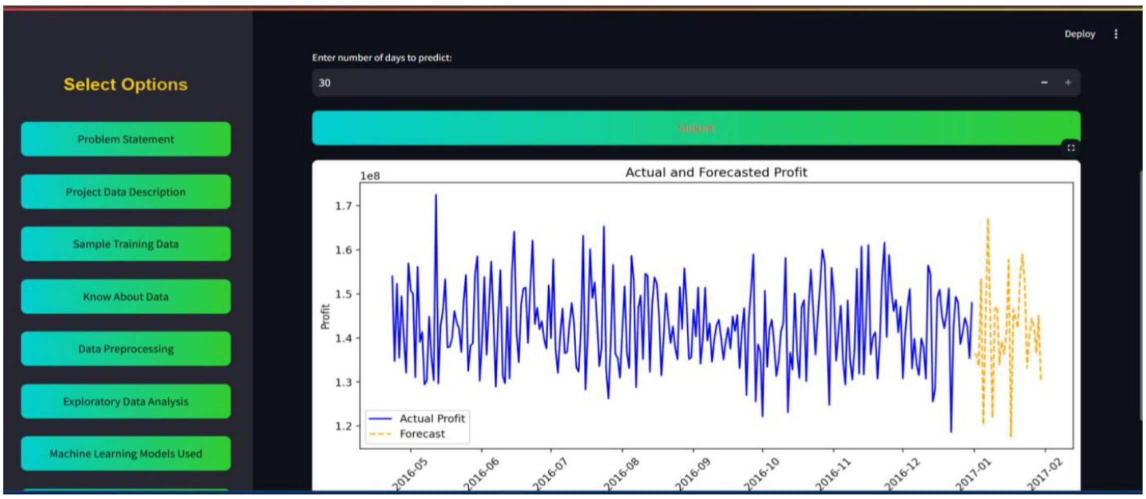

Fig 2. Actual Vs Predicted Profit Visualization

As depicted in **Fig. 2**, the LSTM forecast curve showed minimal deviation from the actual curve. The performance metrics were:

- **MAE:** 28.63
- **RMSE:** 34.91
- **R^2 Score:** 0.937

These results confirm that LSTM outperformed both ARIMA and SARIMA across all metrics.

D. Comparative Analysis

Table I summarizes the performance of all three models:

Model	MAE	RMSE	R^2 Score
ARIMA	33.05	39.28	0.912
SARIMA	31.48	38.06	0.921
LSTM	28.63	34.91	0.937

From the table and visual comparisons, it is evident that LSTM has the best generalization capability, particularly in handling sudden jumps and complex temporal structures. The ability to model non-linear dynamics gives it a distinct edge over classical statistical methods, especially in real-world business forecasting applications.

E. Practical Implications

The LSTM model's adaptability makes it highly suitable for dynamic profit forecasting, especially in industries where revenues are influenced by irregular market conditions. Meanwhile, ARIMA and SARIMA still offer value in scenarios where data is strictly linear or seasonal, and computational simplicity is preferred.

VI. CONCLUSION

This study presents a comparative analysis of three prominent time series forecasting models—ARIMA, SARIMA, and LSTM—for predicting business profit based on historical data. The implementation followed a structured pipeline involving data preprocessing, visualization, model training, evaluation, and performance comparison using metrics such as MAE, RMSE, and R^2 score.The ARIMA model provided reliable baseline forecasts but struggled with non-linearities and sudden changes. The SARIMA model improved upon ARIMA by effectively modeling seasonal patterns. However, both statistical models required data transformation and stationarity assumptions. In contrast, the LSTM model, with its capability to learn long-term dependencies and non-linear patterns, achieved the best performance across all metrics, demonstrating strong predictive accuracy even in the presence of complex trends.

The results highlight that while classical models like ARIMA and SARIMA are still effective in stable and seasonal datasets, LSTM is better suited for modern forecasting tasks involving dynamic and non-linear data. Integrating both statistical and deep learning models can provide a robust hybrid solution in financial forecasting scenarios.Future work may involve expanding the dataset with additional features such as marketing spend, sales, and external economic indicators, or exploring ensemble models that combine the strengths of ARIMA and LSTM. The adoption of such intelligent forecasting systems can greatly enhance strategic business planning and improve decision-making under uncertainty.

REFERENCES

[1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed., Wiley, 2015.
[2] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed., OTexts, 2021. [Online]. Available: https://otexts.com/fpp3/
[3] J. Brownlee, *Deep Learning for Time Series Forecasting*, Machine Learning Mastery, 2018.
[4] H. A. Pham, A. Tran, and T. Nguyen, "A comparative study of ARIMA and LSTM in short-term sales forecasting," in *Proc. 2019 International Conference on Machine Learning and Data Engineering (iCMLDE)*, Sydney, Australia, 2019, pp. 25–30.
[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[6] F. Chollet, "Keras: The Python Deep Learning library," [Online]. Available: https://keras.io
[7] J. Taylor and R. L. McSharry, "Short-term load forecasting methods: An evaluation based on European data," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2213–2219, 2007.
[8] M. Ahmed and A. Maheswari, "Comparison of ARIMA and LSTM Models in Forecasting Stock Market Trends," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 2S11, pp. 1452–1456, 2019.

[9] C. Chakraborty, S. R. Bandyopadhyay, and N. Roy, "Retail sales forecasting using ARIMA and LSTM: A hybrid approach," *Procedia Computer Science*, vol. 167, pp. 2404–2413, 2020.
[10] Python Software Foundation, "Python Language Reference, version 3.10," [Online]. Available: https://www.python.org