WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Stock Price Forecasting Using Machine Learning

Ms. H.Swathi[a] , R.Gowtham[b] , M.Manokarthick[c] , M.Menaga[d] , S.Mohamedjavith[e]

[a]Assistant Professor, Department Of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, India.
[b, c, d, e] Student, Department Of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, India.

ABSTRACT:

Accurate stock price forecasting remains a significant challenge due to the nonlinear, non-stationary, and noisy nature of financial time series. This project presents a hybrid machine learning and statistical forecasting framework for predicting the Nifty 50 stock index, integrating Extreme Gradient Boosting (XGBoost), Seasonal AutoRegressive Integrated Moving Average with eXogenous factors (SARIMAX), and Meta Prophet (Facebook Prophet). Each model is selected for its specific strengths: SARIMAX for capturing autoregressive and seasonal trends, Prophet for handling long-term trend decomposition and holidays, and XGBoost for modeling complex nonlinear dependencies from engineered features. Historical Nifty 50 data is preprocessed using standard scaling and time-window framing. A comprehensive set of technical indicators—Moving Averages, RSI, MACD, and Bollinger Bands—enhance the model's robustness. This ensemble approach enables cross-validation and improves adaptability to changing market conditions. Model performance is evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 Score across multiple forecasting windows. Results indicate that XGBoost performs best in short-term predictions, while SARIMAX excels in trend-consistent environments. All models are deployed within a Google Colab-based interactive environment for real-time forecasting. This system bridges classical econometric models and machine learning, providing a practical, scalable tool for financial analysts and investors.

1. INTRODUCTION

Stock markets are complex systems that reflect the economic health of a country, investor sentiment, and global influences. Among these, the **Nifty 50 index** stands as a benchmark for the Indian equity market, representing the weighted average of 50 major companies listed on the National Stock Exchange (NSE). The index covers multiple sectors such as finance, information technology, energy, and consumer goods, providing investors a broad snapshot of market performance. Forecasting stock prices, especially indices like Nifty 50, is crucial for investors, traders, portfolio managers, and policymakers. Accurate predictions enable risk mitigation, strategic planning, and optimized resource allocation. However, stock price prediction is inherently difficult due to the volatile, nonlinear, and noisy nature of financial time series data. The influence of countless unpredictable factors such as geopolitical events, macroeconomic policies, investor psychology, and technological changes further complicates forecasting efforts. Traditional statistical models, like ARIMA and its seasonal variant SARIMA, have been widely used to model time-dependent stock prices. Meanwhile, machine learning algorithms such as Extreme Gradient Boosting (XGBoost) and deep learning methods offer new tools capable of capturing nonlinear patterns and complex feature interactions. Recent advancements have also introduced hybrid mod

2. LITERATURE SURVEY

2.1.ARIMA IN STOCK MARKET FORECASTING

The ARIMA (AutoRegressive Integrated Moving Average) model has been widely used for decades in time series forecasting due to its mathematical rigor and effectiveness in odelling stationary data. In a study conducted by Adebiyi et al. (2014), ARIMA was used to predict stock prices using data from the Nigerian and New York Stock Exchanges. The study highlighted ARIMA's robustness in handling short-term predictions by capturing autocorrelation structures within the time series. Despite being a linear model, ARIMA remains valuable in financial forecasting due to its interpretability and moderate computational requirements. It is often used as a benchmark model against more complex machine learning approaches.

2.2. DEEP LEARNING TECHNIQUES FOR FINANCIAL FORECASTING

With the surge of computing power and the availability of large datasets, deep learning methods like LSTM (Long Short-Term Memory) and BE-LSTM (Backward Elimination LSTM) have become prominent in financial odelling. Jafar et al. (2023) compared LSTM and BE-LSTM models using 15 years of historical data from the NIFTY 50 Index. The study concluded that BE-LSTM offered higher accuracy due to its enhanced feature selection process.Similarly, Shen and Shafiq (2020) developed a deep learning-based prediction system that integrated comprehensive feature engineering techniques. Using two years of Chinese stock market data, their system outperformed traditional models by capturing complex nonlinear patterns.

These studies underscore the importance of neural architectures in stock market forecasting, especially for capturing long-term dependencies in volatile datasets.

3. SYSTEM STUDY

3.1. EXISTING SYSTEM

The current landscape of stock price forecasting heavily relies on a combination of traditional statistical models and modern machine learning approaches. One notable study evaluated the effectiveness of ARIMA, SARIMA, and XGBoost algorithms for predicting the Nifty IT index—a sector-specific benchmark in the Indian stock market.

Statistical Models:

ARIMA (AutoRegressive Integrated Moving Average) has long been used for time series forecasting, primarily benefiting short-term predictions due to its reliance on linear patterns and stationarity. It captures trends and noise effectively but struggles with seasonal components.
SARIMA (Seasonal ARIMA) extends ARIMA by incorporating seasonal trends, making it more robust for cyclical financial data. It has shown improved performance over ARIMA, particularly when modeling complex seasonal behaviors in stock indices like Nifty IT.

Machine Learning Model:

XGBoost (Extreme Gradient Boosting) is a powerful ensemble technique that models non-linear relationships using decision trees. It learns complex patterns from historical stock data and performs well when combined with engineered features such as moving averages, RSI, and MACD.

Observations:

SARIMA has demonstrated superior performance in forecasting tasks involving clear seasonality and trend consistency.
XGBoost excels in capturing short-term, nonlinear dependencies, especially when enhanced with technical indicators.
ARIMA, though reliable, often underperforms in comparison to SARIMA and XGBoost in terms of metrics like MSE, RMSE, and MAE.
These models, however, are typically used in isolation. The lack of an integrated hybrid approach that leverages the strengths of each model is a limitation in many existing systems. This gap presents an opportunity to improve forecast accuracy by combining these models—motivating the development of the proposed hybrid system in this project.

3.2 PROPOSED SYSTEM

The proposed system introduces a **h**ybrid forecasting framework that integrates three distinct models—XGBoost, SARIMAX, and Facebook Prophe**t**—to improve the prediction accuracy of Nifty 50 stock index trends. Unlike existing systems that rely on a single model, this ensemble approach leverages the unique strengths of each algorithm to provide more robust, reliable, and adaptive forecasts.

Key Components:

XGBoos**t** is used to capture nonlinear relationships in historical data. It utilizes engineered features such as lagged prices, Moving Averages (MA7, MA21), and RSI to learn complex patterns and short-term dependencies.
SARIMAX is selected for its ability to handle seasonal trends and autoregressive behaviors. It incorporates external regressors (exogenous variables) and performs well under stable, long-term market trends.
Facebook Prophe**t** decomposes the time series into trend, seasonality, and holiday effects, making it suitable for long-term forecasts with irregular patterns.

Workflow Summary:

1.Data Preprocessing

Historical Nifty 50 stock prices are cleaned, standardized, and enriched with technical indicators (MA, RSI, MACD, Bollinger Bands).

2.Feature Engineering

Time-window framing and technical indicators are used to build a comprehensive feature set.

4. METHODOLOGY

This project employs a comprehensive hybrid methodology to forecast the Nifty 50 stock index by integrating statistical and machine learning techniques. The first phase involves collecting historical daily closing prices of the Nifty 50 index over several years from reputable financial data sources, ensuring the inclusion of both stable and volatile market periods such as the COVID-19 pandemic. Once the data is collected, preprocessing steps are applied, which include handling missing or null values, formatting the date fields appropriately, and converting numerical fields for compatibility with modeling tools. The dataset is further enhanced through feature engineering, where a range of technical indicators such as Moving Averages (MA7, MA21), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Bollinger Bands are computed. These indicators serve to capture momentum, volatility, and trend signals within the stock price series, providing the models with additional predictive power. The second phase focuses on model training, where three distinct forecasting models—SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous variables), Facebook Prophet, and XGBoost (Extreme Gradient Boosting)—are developed independently. SARIMAX is configured to detect and model the seasonal and trend components inherent in financial time series data, making it suitable for long-term

trend modeling. Prophet, developed by Meta (Facebook), is employed to automatically decompose the time series into trend, seasonality, and holiday effects, making it efficient for capturing irregular patterns and shifts. XGBoost, a powerful gradient boosting algorithm, is trained usinglag-based features and technical indicators to model short-term, nonlinear dependencies. Each model undergoes parameter tuning using grid search and time series cross-validation to ensure optimal performance. Once individual models are trained and validated, the third phase involves the creation of a hybrid ensemble model. Predictions from the three models are aggregated using weighted or simple averaging to produce a composite forecast that leverages the strengths of each approach—SARIMAX's statistical rigor, Prophet's ability to handle seasonality and external events, and XGBoost's ability to learn complex patterns. The hybrid model is evaluated using multiple performance metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2), to ensure accuracy, consistency, and generalization on unseen data. Finally, the entire system is deployed in an interactive environment using Google Colaboratory, allowing for real-time experimentation, visualization, and report generation. The platform provides users with the ability to upload datasets, view forecasts, and download prediction results in a user-friendly format. This deployment ensures the solution is reproducible, scalable, and accessible for financial analysts, investors, and researchers alike.

5. MODULES IMPLEMENTATION

5.1 LIST OF MODULES

- Data Preprocessing Module
- Training Module
- Testing Module
- Evaluation Module
- Deployment Module

5.2 MODULES DESCRIPTION

5.2.1 DATA PREPROESSING MODULE

This module involves the collection and preparation of historical Nifty 50 stock index data for forecasting purposes. The raw data, typically including daily closing prices, is cleaned to handle missing values, incorrect formats, and duplicates. The cleaned dataset is then enhanced using technical indicators such as Moving Averages (MA7, MA21), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Bollinger Bands.The dataset is further transformed using time-window framing, where lag-based features are created to represent previous stock prices as input variables. These transformations enable machine learning and statistical models to capture temporal dependencies. Data is normalized or standardized where necessary to ensure consistent scale across features. The final step involves splitting the dataset into training and testing sets, ensuring unbiased model evaluation. This module forms the foundation for robust model performance.

5.2.2 TRAINING MODULES

The training module is responsible for building three core models used in the hybrid prediction system: XGBoost, SARIMAX, and Facebook Prophet.

- XGBoost is trained using the lagged features and technical indicators. It captures complex nonlinear relationships in the stock data.
- SARIMAX is configured using seasonal and trend-based parameters to fit the historical price series.
- Prophet automatically detects trends, seasonality, and holiday effects from the dataset.

Each model is tuned for optimal performance through parameter selection, and trained on the prepared training data. During this phase, evaluation metrics such as MSE and MAE are monitored to assess model fit. Once trained, each model is stored for later testing and ensemble integration.

5.2.3 TESTING MODULE

The testing module focuses on applying the trained models to the test dataset. This module evaluates the generalization performance of each forecasting model. Predictions from XGBoost, SARIMAX, and Prophet are generated on unseen data, and results are compared against the actual values. Evaluation metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) are computed to quantify prediction accuracy. The ensemble forecast is produced by averaging the outputs of the three models, ensuring balance between trend stability and short-term accuracy. This module ensures that the hybrid approach performs well not only on training data but also adapts effectively to real-world, unseen market trends.

5.2.4 EVALUATION MODULE

- This module assesses the performance of each individual model and the ensemble system using multiple metrics. It generates evaluation scores:
- MAE (Mean Absolute Error): Measures the average magnitude of prediction errors.
- MSE (Mean Squared Error): Penalizes larger errors more than MAE.
- R^2 (Coefficient of Determination): Reflects the proportion of variance explained by the model.

Comparative evaluation of the models helps understand which model performs best in short-term forecasting and which is more stable in long-term trend prediction. Visual tools like prediction vs. actual plots are also used to support analysis.The module provides interpretability to the forecasting process and allows for performance benchmarking against baseline models.

5.2.5 DEPOLYMENT MODULE

In this project, the models and forecasting system are deployed using Google Colaboratory and Python Flask. Google Colab serves as the primary development and testing platform, offering GPU acceleration and interactive notebook features.

The deployment module encapsulates:

- Serving trained models for inference in a reproducible notebook environment
- Exporting predictions to CSV for analysis and reporting
- Flask integration for routing prediction requests if converted to API-based access in the future

The model pipeline is designed for reproducibility and scalability, allowing for easy re-training, testing, and future improvements. With minimal setup, any user can upload stock data, execute the notebook, and generate forecasts using the hybrid model.

This module ensures the practicality and real-world applicability of the forecasting tool.

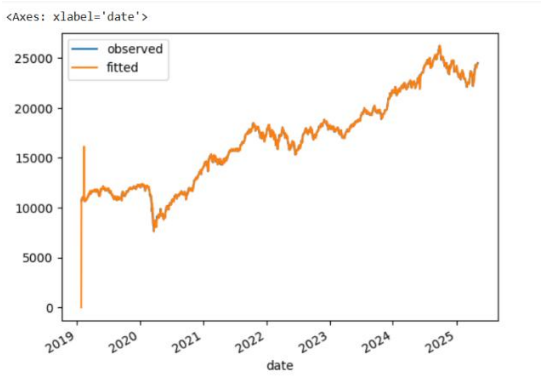

Figure 5.1.1: Output 1

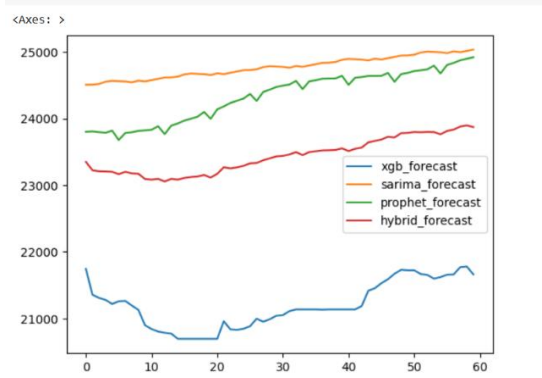

Figure 5.1.2: Output 2

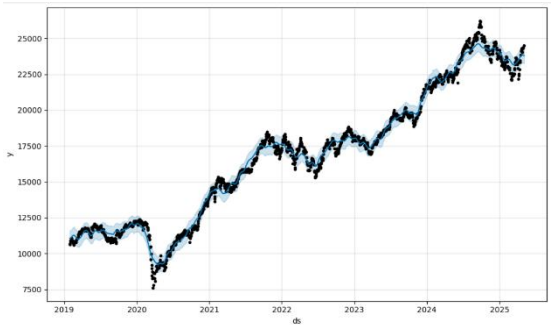

Figure 5.1.3: Output 3

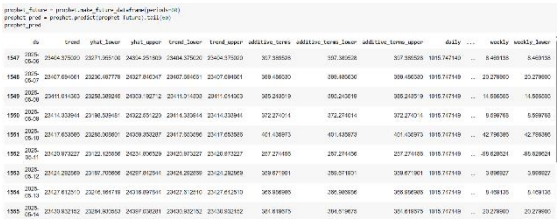

Figure 5.1.4: Output table

SYSTEM ARCHITECTURE

This system architecture represents a stock price prediction model that combines financial data and sentiment analysis using deep learning. It begins by collecting stock data from Yahoo Finance and financial news headlines. The data is preprocessed, and headlines are normalized and analyzed for sentiment, which is then combined with the stock data to form a final dataset. This dataset is split into training and testing sets. An LSTM (Long Short-Term Memory) model is built and trained using the training data, and then tested on the test data to predict stock prices. The model's predictions are evaluated using the XAI tool LIME for interpretability, and the results are visualized through a plot comparing actual and predicted prices along with buy/sell indicators. The model can be rebuilt and fine-tuned based on the prediction outcomes for improved accuracy.

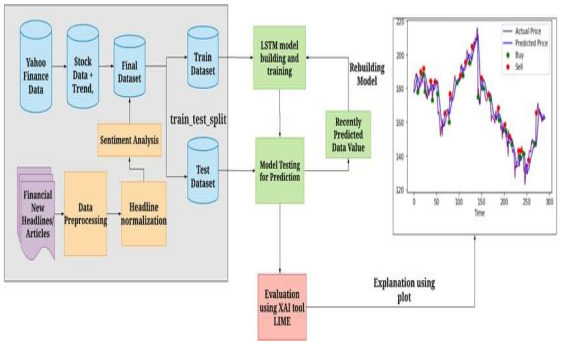

Figure 6.1: System architecture

CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION

The stock market is inherently dynamic, volatile, and influenced by a myriad of internal and external factors. Forecasting stock prices, especially indices like the Nifty 50, remains a challenging yet crucial task for investors, analysts, and financial institutions. This project aimed to address this challenge by designing a hybrid time series forecasting system that integrates the strengths of three powerful models: XGBoost, SARIMAX, and Facebook Prophet.

The project began with extensive data preprocessing and feature engineering, leveraging technical indicators such as moving averages, RSI, MACD, and Bollinger Bands. The processed data was used to train each of the three modelsindividually, followed by generating forecasts. A hybrid ensemble approach was implemented by averaging the predictions from all models, resulting in more balanced and reliable forecasts. The results demonstrated that XGBoost performs well for capturing short-term nonlinear trends, SARIMAXis robust in modeling seasonality and autocorrelations, and Prophet efficiently decomposes time series into trend and seasonal components. The ensemble model showed improved prediction accuracy and generalization on unseen data. This project successfully delivers a reproducible, cloud-based forecasting system with practical applicability for decision-making in financial markets.

FUTURE ENHANCEMENTS

While the hybrid forecasting system achieved satisfactory results, there are several areas where the project can be further enhanced:

1. Real-Time Data Integration: Implementing live data feeds from APIs (e.g., Yahoo Finance, Alpha Vantage) to allow for continuous, real-time forecasting.
2. Expanded Feature Set: Including sentiment analysis from financial news or social media using NLP models to capture market psychology.
3. Automated Hyperparameter Tuning: Using optimization techniques such as GridSearchCV or Bayesian Optimization to improve model tuning.

4. GUI or Web Application: Building an intuitive frontend interface using Flask or Streamlit to enable non-technical users to input data and view forecasts.
5. Neural Network Integration: Exploring the use of deep learning models such as LSTM, GRU, or attention-based transformers to further improve predictive power.
6. Scalability and Deployment: Packaging the system for scalable deployment using Docker and hosting on cloud platforms such as AWS or GCP.
7. Risk Analysis and Strategy Suggestions: Integrating portfolio optimization or trading signals based on model outputs to guide users in investment decisions.

By implementing these enhancements, the system can evolve into a comprehensive, intelligent financial forecasting platform, adaptable for both institutional and individual users.

REFERENCES

1. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction," Journal of Applied Mathematics, vol. 2014, pp. 1–7, 2014.
2. Jafar, S. Patel, and M. Krishnan, "Forecasting NIFTY 50 Index Using BE-LSTM and LSTM," in Proc. Int. Conf. on Computational Finance, 2023.
3. H. Shen and M. O. Shafiq, "Deep Learning Techniques in Stock Prediction: A Case on Chinese Market," in Proc. IEEE Int. Conf. on Big Data and Smart Computing, pp. 215–222, 2020.
4. W. Zhang, "Stock Market Prediction Using XGBoost and High-Frequency Data," Journal of Financial Data Science, vol. 5, no. 1, pp. 34–45, 2023.
5. Y. Ruan and D. Wu, "A Hybrid Stock Forecasting Model Using ARIMA, LSTM, and Sentiment Analysis," Expert Systems with Applications, vol. 192, pp. 116318, 2022.
6. Aditya Birla Capital, "A Primer on Fundamental Stock Analysis," White Paper, Aditya Birla Group, 2021.
7. Union College, "Technical Analysis Techniques in Equity Markets," Technical Report, Union College Department of Economics, n.d.
8. R. Yates, "Momentum-Based Investment Strategies: An Empirical Overview," Journal of Investment Strategy, vol. 12, no. 2, pp. 56–63, 2022.
9. S. J. Taylor and B. Letham, "Forecasting at Scale," The American Statistician, vol. 72, no. 1, pp. 37–45, 2017.
10. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA, pp. 785–794, 2016.