International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Forecasting the Future: A Practical Approach to Stock Price Prediction Using Time Series and Machine Learning Models

Gouri J[1], Jogimol Joseph[2]

[1]Department of Computer Applications, Musaliar College of Engineering & Technology, Pathanamthitta, Kerala, India
Email: *jgouri163[at]gmail.com*

[2]Professor, Department of Computer Applications, Musaliar College of Engineering & Technology, Pathanamthitta, Kerala, India

Abstract: ***Time Series Forecasting is a method used to analyse data points collected at regular intervals for identifying trends, seasonality and future prediction. In machine learning, Time Series Forecasting is used to make future predictions based on historical trends. In financial sector, time series forecasting plays a crucial role in creating accurate financial predictions. By using historical data and economic indicators, companies can estimate future revenues, expenses, and cash flows, enabling informed decision-making in budgeting, investment, and loan assessments. Beyond prediction, the system provides company reviews based on forecasted stock performance, considering factors like price stability, growth potential, and prediction market trends. This project leverages time series forecasting to predict future stock prices using advanced models such as Machine learning models, Logistic Regression, and Random Forest.***

Keywords: Stock market, Market trends**,** Logistic Regression, Random Forest, Time Series

1. Introduction

The stock market forecasting system employs machine learning algorithms to estimate future stock prices using historical data. It features a web-based interface that allows users to choose a company, define a forecast period, and explore projected stock trends through interactive visualizations. The system's main objective is to help users interpret stock price behaviour by analysing past performance. By utilizing previous market data and key economic indicators, businesses can assess their financial stability, gauge profitability, and better handle uncertain market conditions. This type of predictive analysis is especially valuable to the financial sector, as it supports informed decision-making in areas such as budgeting, investment planning, and risk assessment.

2. Literature Survey

H. Tang et al. (2025) offer an insightful examination of the application of Large Language Models (LLMs) in time series forecasting. Their study emphasizes the strengths of LLMs, including their adaptability, strong generalization capabilities, and effectiveness in processing multimodal data. At the same time, the research points out key drawbacks, such as the substantial computational resources required, relince on large datasets, limited interpretability, and difficulties in accurately capturing temporal relationships.[1]

Zaina Saadeddin (2025) highlights ARIMA as a widely adopted and foundational method for time series forecasting, valued for its simplicity and effectiveness with univariate data. Despite its strengths—such as ease of use and noise tolerance—it faces limitations in modelling non-linear patterns, accounting for seasonality, and supporting long-term forecasts.[2]

Wenxiang Li (2024) reviews key techniques for time series analysis, focusing on popular forecasting models and their practical applications. The study outlines the progression of deep learning in this domain, highlighting its benefits and cross-industry use cases. It serves as a valuable guide for researchers and practitioners exploring deep learning in time series forecasting.[3]

Yasika Sharma (2023) explores the importance of time series analysis in meteorology, highlighting its role in uncovering patterns and trends in weather data. The paper evaluates various forecasting techniques, including statistical approaches and machine learning models, to determine their effectiveness in predicting weather variables such as temperature, humidity, and precipitation.[4]

Brendan Artley (2022) highlights that ensemble learning enhances predictive accuracy and reduces overfitting by combining various model strengths. He likens this to real-world problem-solving, where diverse approaches yield better solutions. In time series forecasting, ensemble learning significantly improves prediction accuracy, robustness, and generalization. [5]

Julien Hazeen (2022) emphasizes transfer learning as a powerful technique for overcoming data scarcity and improving the performance of time series forecasting models. By applying knowledge from one domain or task to another, this method reduces the need for extensive labelled datasets, enhances model generalization, and shortens training time. Nevertheless, issues such as domain mismatch, negative transfer, and the complexity of adapting models must be carefully addressed to ensure effective implementation. [6]

Christiene Sinoquet (2021) presents that the paper "Time Series Analysis and Modeling to Forecast: A Survey" offers a valuable and comprehensive review of time series forecasting techniques, from traditional methods to modern machine learning approaches. Its strengths lie in its broad coverage, theoretical insights, and the introduction of hybrid models. [7] Ricardo P. Masini (2021) explores recent advancements in supervised machine learning and high-dimensional models for time series forecasting. The study examines both linear and nonlinear approaches. Among the linear methods, particular emphasis is placed on penalized regression techniques and ensemble models. For nonlinear approaches, the paper covers shallow and deep neural networks—both feed-forward and recurrent architectures—as well as tree-based methods such as random forests and gradient-boosted trees.[8]

Indrajeet Y. Javeri et al. (2021) emphasize that applying data augmentation and AutoML to neural networks can significantly enhance time series forecasting by improving accuracy, efficiency, and accessibility. Synthetic data generation and automated optimization support better generalization, reduced overfitting, and faster development. However, challenges such as data artifacts and high computational demands must be carefully considered.[9]

Christoph Bergmeir et al. (2021) highlight that Recurrent Neural Networks (RNNs) are effective tools for time series forecasting, excelling at capturing temporal dependencies and modelling complex patterns. While RNNs, particularly LSTMs and GRUs, offer high adaptability and can handle non-stationary data, they have limitations, including issues with vanishing gradients, overfitting, computational complexity, and limited interpretability.[10]

Daniel Berberich (2020) explains that hybrid models in time series forecasting combine various techniques to enhance prediction accuracy and overcome the limitations of individual methods. By integrating statistical models (ARIMA, Exponential Smoothing) and machine learning approaches (Neural Networks, Random Forests), these models capture different data characteristics, such as trend, seasonality, and noise, more effectively.[11]

Jorg K. Zink (2019) discusses the advantages of Bayesian Neural Networks (BNNs) for time series prediction, especially in scenarios involving noisy data or when forecasting uncertainty is critical, such as in financial risk prediction. The paper offers a comprehensive framework for implementing BNNs and compares their performance against traditional neural networks and ARIMA models. Recent studies have concentrated on enhancing the scalability and efficiency of BNNs.[12]

Ivan Mendez Jiménez et al. (2018) highlight the growing importance of time series forecasting within the Big Data ecosystem. The increasing availability of large datasets in both industry and research, along with the potential for higher profits from more accurate predictions, has led to the increased use of Deep Learning techniques in time series forecasting. This study evaluates the improvement in forecasting performance of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) when using trend, seasonal, and residual components of time series—generated through Seasonal and Trend decomposition using Loess (STL)—as input, rather than the raw time series data. [13]

Omer Beyca et al. (2015) emphasize that forecasting complex system behaviours is considered one of the major challenges in modern science. Time series data from these systems reflect the dynamic patterns and causal interactions of the underlying processes, making it a useful tool for predicting and monitoring the evolution of system states. However, the nonlinear and non-stationary characteristics of these processes pose significant difficulties for accurate forecasting. In most real-world systems, the state dynamics are governed by a nonlinear relationship between the state variables, their autoregressive components, and external factors. [14]

Ivan Mitov et al. (2011) discuss the limitations of traditional time series models in forecasting financial market crashes. The paper proposes a framework designed to predict both extreme events and highly volatile market conditions. According to the empirical evidence presented, this framework offers a more effective approach than existing models for assessing stock market risk during periods of market distress.[15]

3. Methodology

This study utilizes time series forecasting techniques to predict financial metrics such as stock prices, exchange rates, and commodity prices. The methodology involves the following steps:

a) User Registration

The platform begins by enabling both companies and clients to register securely. Companies contribute by uploading their historical stock price data, which serves as the foundation for generating predictive insights. On the other hand, clients create personalized profiles that grant them access to detailed stock forecasts and analysis. This interaction between data providers and data consumers ensures a comprehensive ecosystem for effective stock market predictions and informed investment decisions.

b) Data Collection & Preprocessing

The data collection process starts with compiling historical financial information, such as stock prices, exchange rates, and commodity prices. To enhance forecasting accuracy, additional external economic indicators—including inflation rates, interest rates, and market sentiment data—are also incorporated. After gathering the data, it undergoes thorough preprocessing, which involves cleaning to eliminate inconsistencies, normalizing to standardize the data, and addressing any missing values. These steps ensure the dataset is accurate, consistent, and well-prepared for effective model training and analysis.

c) Feature engineering

This process includes essential steps designed to improve the accuracy and reliability of forecasting models. It involves selecting meaningful features to boost predictive performance, transforming data to reflect seasonal and cyclical patterns, and applying domain-specific enhancements to optimize model training.

d) Time series forecasting model selection
A range of machine learning and statistical models—such as ARIMA, LSTM, and Prophet—are assessed for their effectiveness. The chosen model is then trained on historical data to identify underlying trends, seasonal fluctuations, and cyclical patterns.

e) Model training and optimization
Hyperparameter tuning is carried out to enhance model performance, while regularization methods are applied to minimize the risk of overfitting. The dataset is divided into training and validation sets to enable effective evaluation of the model.

f) Forecasting future stock prices
The next step involves forecasting future stock prices for a specific company. The trained model generates predictions for future prices of stocks, commodities, and exchange rates. These predictions are validated using techniques such as cross-validation and evaluated with performance metrics like Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

g) Deployment & real- time forecasting
The following phase focuses on deployment and real-time forecasting. This involves integrating APIs to deliver real-time prediction updates and automating data input and model retraining processes to ensure continuous improvement.

h) Provide insights to client
The forecasted outcomes are shared with clients through visual tools like trend graphs and statistical summaries. These visualizations provide comprehensive insights, enabling clients to make well-informed investment decisions.

i) Client review & feedback
Clients offer feedback on the accuracy and usability of the forecasts, allowing companies to adjust and enhance their financial strategies accordingly.

j) Risk analysis & limitations
Identifying potential risks in forecasting errors and their impact. Implementing strategies to mitigate uncertainties in predictions.

k) Company profile & updating
Companies update their financial profiles based on new market trends and predictions. Continuous refinement of forecasting models occurs with new data inputs.

3.1 Algorithm used in Time Series Forecasting

The algorithms commonly used in Time Series Forecasting are primarily fall into two categories. They are:

3.1.1 ARIMA (Auto Regressive Integrated Moving Average)
The ARIMA model is a widely used statistical technique for forecasting time series data. It is especially effective for univariate datasets that display trends and seasonality. By providing historical data patterns, ARIMA provides a robust framework for generating future predictions.

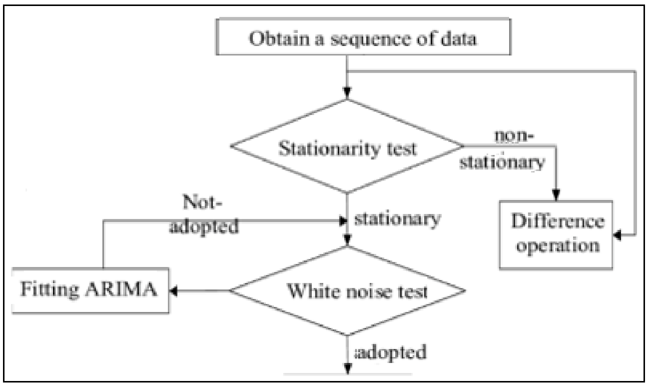

Figure 1: Architecture of ARIMA

Figure 1 shows the architecture of ARIMA model. The ARIMA model works like:

a) Obtain a sequence of data
The process begins with collecting univariate time series data (stock prices).

b) Stationary Test
The next step is to determine whether the data series is stationary or not. If the series is stationary, proceed to the White noise test. If it is not stationary, apply differencing operation to transform it into stationary series.

c) Differencing Operation
Differencing is applied to eliminate trends and achieve stationarity in the data. After performing differencing, the data is re-evaluated for stationarity, repeating the process if it is necessary.

d) White noise test
A White noise test is performed to determine whether the stationary series contain meaningful patterns or if it is just random noise. If the data is identified as white noise, ARIMA is not adopted. If it is not white noise, ARIMA is adopted.

e) Fitting the ARIMA model
If the data is seemed suitable (stationary and non-white noise), an ARIMA model is fitted to it.

3.1.2 Prophet

Prophet is an open-source forecasting tool created by Facebook (Meta). It is specefically designed to handle time series data by capturing trends, seasonality and holiday effects. Prophet is particularly useful for business forecast.

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

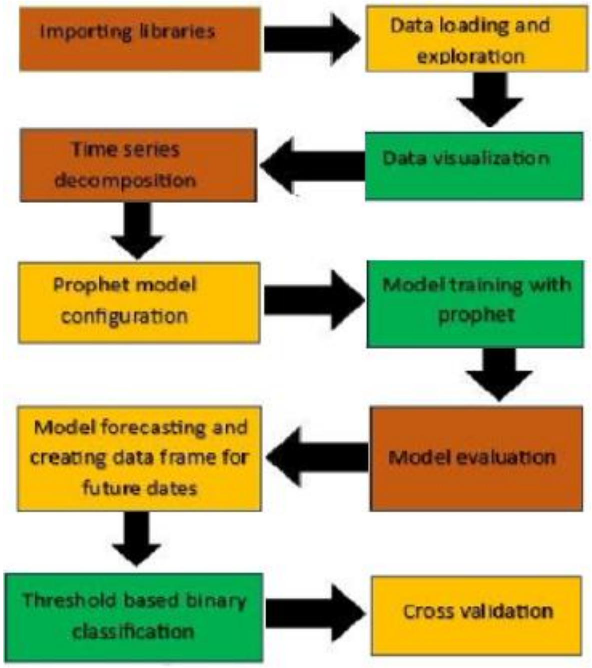

Figure 2: Architecture of Prophet

Figure 2 shows the architecture of Prophet model. The working of Prophet model:

a) Importing Libraries → Data Loading and Exploration
The process begins with importing necessary libraries such as prophet, pandas, NumPy and matplotlib. The next step involves loading time series data and exploring it for trends, missing values and anomalies.

b) Time Series Decomposition → Data Visualisation
Time series decomposition breaks down data into components: trend, seasonality and residual to better understand the patterns. The results are then visualised using graphs to analyse data patterns before applying forecasting techniques.

c) Prophet Model Configuration →Model training with Prophet
Prophet model is initialized with specific setting such as growth type, seasonal component and external regressors if needed. The model is then trained on historical time-series data, learning trends and seasonal patterns.

d) Model Forecasting and Creating Data Frames for Future Dates → Model Evaluation
After training, the model predicts future values based on historical trends. The forecasted results are stored in a new Data Frame, which contains predicted values along with upper and lower confidence intervals. The model's performance is evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or Mean Absolute Percentage Error (MAPE).

e) Threshold-Based Binary Classification → Cross Validation
A threshold-based classification is applied to categorize forecasted results. Cross-validation is used to assess model robustness by splitting the data into training and test sets, ensuring the model generalizes well to unseen data.

3.2 Dataset Description

The dataset used for financial time series forecasting should contain historical financial data, market indicators, and economic variables to enable accurate predictions. Below is a structured description of the dataset:

a) Dataset information
This project makes use of a Financial Time Series Forecasting Dataset obtained from Yahoo Finance. The dataset contains a combination of historical and real-time market data, providing valuable insights into stock trends and price movements. It includes key indicators such as stock prices, trading volumes, and various other market metrics. The data is available at different time intervals—from minutes to days—enabling flexible analysis for both short-term and long-term forecasting purposes.

b) Coverage (financial instruments included)
The dataset provides broad coverage of diverse financial instruments, making it well-suited for various forecasting applications. It includes a wide selection of stocks from major companies such as Apple (AAPL), Tesla (TSLA), and Microsoft (MSFT), enabling in-depth equity analysis. Additionally, it features key market indices like the S&P 500, Dow Jones, and Nasdaq, along with several international indices, offering a global market perspective. The dataset also encompasses major cryptocurrencies, including Bitcoin (BTC) and Ethereum (ETH), addressing the increasing relevance of digital assets. Moreover, it contains forex data with exchange rates for significant currency pairs such as USD/EUR and USD/JPY, supporting comprehensive financial analysis and cross-market forecasting.

c) Data access methods
There are two main ways to access the dataset. First is REST API for developing phase and Python Library (yfinance) to fetch and process Yahoo Finance data efficiently.

d) Key features of Yahoo Finance Api
Historical data covering stocks, indices, forex, and cryptocurrencies is utilized. Real-time price updates are incorporated for timely market analysis. Key market indicators—such as opening and closing prices, trading volume, and moving averages—are also included. Additionally, economic variables are integrated to enhance the accuracy of financial predictions.

4. Result & Discussion

The classification performance was evaluated using common metrics such as accuracy, precision, recall, F1-score, and the ROC curve. The results were visualized using a confusion matrix heatmap and a classification report table. A confusion matrix heatmap visually illustrates the model's performance, allowing for a quick evaluation of how well the model is working, highlighting areas where misclassifications may occur, and guiding future tuning or improvements. The classification report provides a detailed overview of the model's performance across various metrics for each class within the dataset.

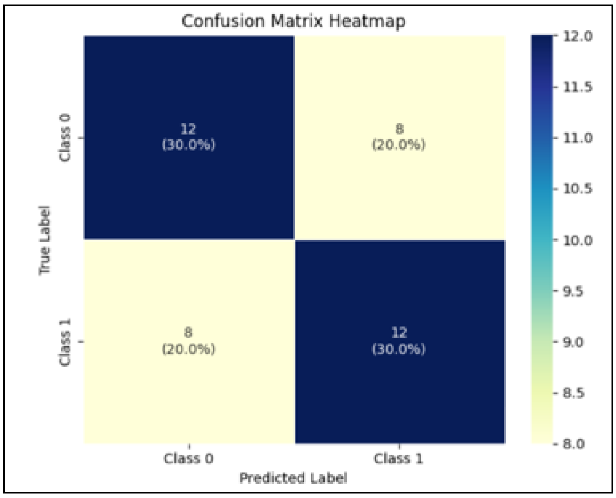

Figure 3: Confusion matrix heat map

Table 1: Classification Report Table

	Precision	Recall	F1- Score	Support
Class 0	0.6	0.6	0.6	20.0
Class 1	0.6	0.6	0.6	20.0
Accuracy	0.6	0.6	0.6	0.6
Macro avg	0.6	0.6	0.6	40.0
Weighted avg	0.6	0.6	0.6	40.0

a) Accuracy
Accuracy describes whether the true or actual values predicted class or not. It is calculated by taking the difference between the actual class and the predicted class. Accuracy is the ratio of truly predicted class to the total number of instances. The trained model achieved on overall accuracy of 100%.

b) Precision
Precision refers to the proportion of correctly predicted positive instances out of all instances predicted as positive. The trained model achieved a macro-averaged precision of 100%.

c) Recall
Recall is calculated by taking the ratio of true positive instances to the sum of true positive and false negative instances. The model achieved a recall of 100%.

d) F1-score
F1-score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. The model achieved an F1-score of 100%.

5. Conclusion

In conclusion, the comparative analysis highlights that no single model is universally optimal for all aspects of stock price forecasting. While Random Forest and XGBoost are efficient and effective for short-term predictions due to their speed and simplicity, LSTM demonstrates superior performance in capturing long-term patterns with higher accuracy, albeit at the cost of greater computational demand. Additionally, the AI-driven company rating system proves to be valuable for investors by aligning well with actual market trends. Therefore, a hybrid approach that leverages the strengths of both traditional machine learning and deep learning models offers a comprehensive and practical solution for accurate and insightful stock forecasting.

References

[1] Time Series Forecasting with LLMs: Understanding and Enhancing Model Capabilities by H Tang, C Zhang, M Jin, Q Yu, Z Wang and X Jin (2025)
[2] ARIMA for Time Series Forecasting: A Complete Guide by Zaina Saadeddin (2025)
[3] Deep Learning Model for Time Series Forecasting: A Review by Wenxiang Li (2024)
[4] Time Series Forecasting for Weather Prediction by Yashika Sharma (2023)
[5] Time Series Forecasting using Ensemble Learning by Brendan Artley (2022)
[6] Transfer Learning for Time Series Forecasting by Julien Hazeen (2022)
[7] Time Series Analysis and Modelling to Forecast: A Survey by Christiene Sinoquet (2021)
[8] Machine Learning Advances for Time Series Forecasting by Ricardo P Masini (2021)
[9] Improving Neural Networks for Time-Series Forecasting using Data Augmentation and AutoML by Indrajeet Y Javeri, Ismailcen B Arpinar (2021)
[10] Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions by Christoph Bergmeir, Hansika Hewamalagae (2021)
[11] Hybrid Models for Time Series Forecasting by Daniel Berberich (2020)
[12] Time Series Forecasting with Bayesian Neural Networks by Jorg K Zink (2019)
[13] Time Series Decomposition for Improving the Forecasting Performance of Convolutional Neural Networks by Ivan Mendez Jimnez, Miguel Cerdenas Montes (2018)
[14] Time series forecasting for nonlinear and non-stationary processes: A review and comparative study by Omer Beyca, Hui Yang (2015)
[15] Time Series Forecasting in Financial Markets by Ivan Mitov, Frank J Fabozzi (2011)

Paper ID: SR25417110001 DOI: https://dx.doi.org/10.21275/SR25417110001