International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 4, July-Aug 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Predicting Stock Market Trends Using Machine Learning Algorithms

Suram Sai*,
*(Department Of Information Technology, University College Of Engineering, Science & Technology, Jawaharlal Nehru Technological University Hyderabad,Telangana, India.
Email: saikumarreddy919@gmail.com)

----------**********----------

Abstract:
The stock market is inherently dynamic and uncertain, making its prediction a challenging task due to constant fluctuations and instability. The primary objective of this study is to analyze and predict future market trends with greater accuracy and reliability. Over the years, numerous researchers have investigated stock market behavior to understand its evolution and improve forecasting methods. Since stock data is highly volatile, it serves as a critical source for evaluating prediction efficiency, where even small variations can significantly influence outcomes.In recent years, machine learning has emerged as a powerful tool in stock market prediction, offering advanced techniques for handling large datasets and building reliable predictive models. By leveraging machine learning algorithms, particularly regression-based approaches, this research aims to predict stock values with improved precision. The proposed work emphasizes the integration of machine learning regression techniques to enhance decision-making in financial forecasting.

Keywords- **Stock Market Prediction, Random Forest, LSTM, Machine Learning, Time Series Analysis, Financial Forecasting**

----------**********----------

I. INTRODUCTION

The stock market is a complex, nonlinear, and dynamic system influenced by numerous factors such as economic indicators, political events, company performance, and even public sentiment. Predicting stock prices is therefore a highly challenging yet crucial task for investors, policymakers, and financial institutions. Traditional statistical methods like ARIMA and linear regression have been widely used for forecasting, but they often fail to capture the nonlinear patterns and temporal dependencies present in stock data.

In recent years, machine learning (ML) and deep learning (DL) techniques have gained popularity in financial forecasting. Ensemble methods like Random Forest (RF) are robust in handling noisy and high-dimensional data, making them suitable for structured financial datasets. On the other hand, Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, are specifically designed to capture sequential dependencies, making them highly effective for time-series data such as stock prices.

This research compares Random Forest and LSTM in stock market prediction, aiming to analyze their effectiveness, strengths, and limitations. The study contributes by providing a comparative framework

for understanding how traditional ML and advanced DL models can complement each other in financial forecasting.

II. LITERATURE REVIEW

Over the past decade, researchers have applied various approaches to improve stock price prediction accuracy. Traditional statistical models such as ARIMA and GARCH provided early attempts at modeling financial time series, but their performance was limited due to assumptions of linearity.

Recent works have explored machine learning algorithms. Patel et al. (2015) showed that ensemble methods, including Random Forest, achieve better prediction accuracy than single models, though at the cost of higher computational complexity. Similarly, Bollen et al. (2011) and Mittal & Goel (2012) introduced sentiment analysis into prediction models, proving that social media platforms like Twitter could provide leading indicators of market movements.

Deep learning models have further advanced the field. Fischer & Krauss (2018) applied LSTM networks to S&P 500 stock data, achieving higher accuracy than logistic regression and random forest. Nelson et al. (2017) demonstrated the effectiveness of LSTM for the Brazilian stock market (Bovespa), showing significant improvements over support vector machines. Chen et al. (2015) also validated the superiority of LSTM over ARIMA in financial time-series forecasting.

More recent studies have moved towards hybrid and attention-based models. Pardeshi et al. (2023) proposed an LSTM with sequential self-attention, showing improved accuracy for Indian banking stocks. Zheng et al. (2024) highlighted Random Forest's ability to forecast stock trends efficiently, though it still struggles with sequential dependencies.

These works demonstrate that while Random Forest provides robustness and stability in noisy environments, LSTM is more suited to sequential learning. This motivates the current research to perform a comparative analysis of both methods and explore their potential complementarity.

Random Forest has been widely applied in stock prediction due to its robustness in handling noisy data. It constructs multiple decision trees and averages predictions, reducing variance and overfitting. Researchers have demonstrated its efficiency in financial data modeling. LSTM, on the other hand, is designed to capture sequential dependencies in time-series data. Studies, such as Fischer & Krauss (2018), show that LSTM outperforms traditional models in stock market forecasting. Hybrid approaches combining sentiment analysis with stock data have also yielded promising results.

III. PROPOSED METHODOLOGY

The proposed methodology involves collecting stock market data, preprocessing it, and applying two prediction models: Random Forest and LSTM.

3.1 Dataset Description

The dataset includes attributes such as Date, Open, High, Low, Close, Volume, and Name, obtained from Yahoo Finance and Kaggle. It covers multiple years of stock price movements.

3.2 Data Preprocessing

Data preprocessing involved handling missing values, normalization, feature extraction (Moving

Average, Momentum, Bollinger Bands), and splitting into training and test sets. StandardScaler was used to normalize values.

3.3 Algorithms Used

• Random Forest Regressor: Ensemble learning method combining decision trees for robust predictions.
• LSTM: Recurrent neural network that captures temporal dependencies in stock price sequences.
• Time Series Split: Ensures chronological order in train-test splits, preventing data leakage.

IV. IMPLEMENTATION

The proposed system was implemented as a modular pipeline that integrates data collection, preprocessing, model training, prediction, and result visualization. The implementation was carried out using **Python** due to its rich ecosystem of machine learning and deep learning libraries such as **Scikit-learn, TensorFlow, Keras, Pandas, NumPy, and Matplotlib**. The system was deployed as a **Flask-based web application** to provide an interactive interface for end users.

4.1 Data Collection and Storage

Historical stock data was collected from publicly available sources such as **Yahoo Finance and Kaggle**. The dataset consisted of attributes including Date, Open, High, Low, Close, Volume, and Stock Name. The data was stored in a structured CSV format and then imported into Python for analysis.

4.2 Preprocessing and Feature Engineering

Data preprocessing included handling missing values, removing outliers, and normalizing numerical features to ensure consistency. Feature engineering was performed by extracting **technical indicators** such as Moving Averages (MA), Exponential Moving Averages (EMA), Relative Strength Index (RSI), and Bollinger Bands to enrich the dataset and improve prediction accuracy. The dataset was then split into training (80%) and testing (20%) sets using **time-series split** to preserve chronological order.

4.3 Random Forest Model Implementation

The **Random Forest Regressor** from Scikit-learn was used. Hyperparameters such as the number of trees, maximum depth, and minimum samples per split were tuned to optimize model performance. The model was trained on historical features and used to predict closing prices. Random Forest provided robustness against noisy and high-dimensional features but required feature importance analysis to interpret results.

4.4 LSTM Model Implementation

The **LSTM model** was implemented using **Keras with TensorFlow backend**. Input sequences were reshaped into 3D arrays (samples × timesteps × features) to fit LSTM's sequential input requirements. The architecture consisted of:

- Input Layer
- Two stacked LSTM layers with dropout to prevent overfitting
- Dense (fully connected) layer for output prediction

The model was trained using the **Adam optimizer** and **Mean Squared Error (MSE)** as the loss function. Training was performed for 100 epochs with a batch size of 64.

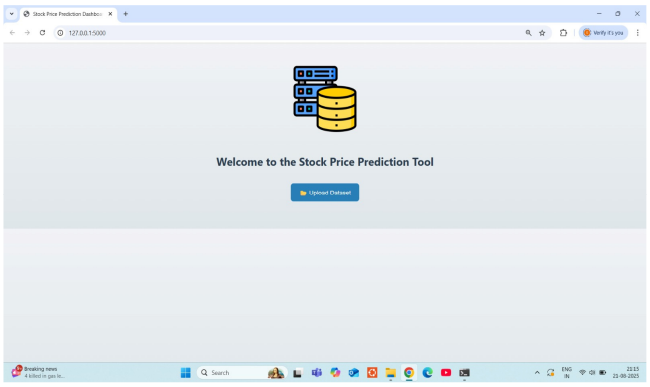
4.5 Model Evaluation and Visualization

Both models were evaluated using **MSE, RMSE, and MAE**. Visualization was performed by plotting actual vs. predicted closing prices to assess the performance of Random Forest and LSTM models. The results were presented using **Matplotlib and Seaborn** plots.

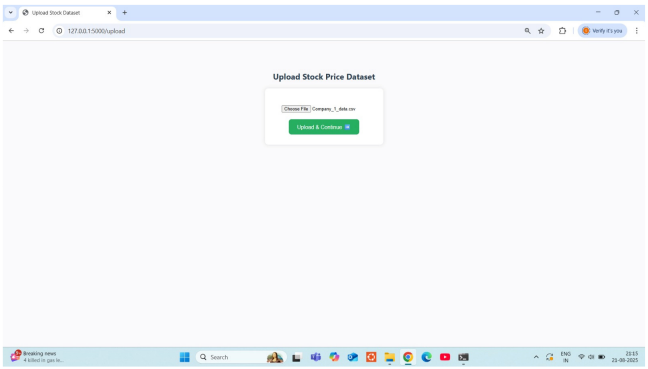
4.6 Web Application Integration

The final system was deployed as a **Flask web application**, allowing users to enter a stock symbol and receive predictions in both graphical and numerical form. The backend handled data preprocessing, model inference, and result generation, while the frontend displayed interactive plots of historical and predicted values.

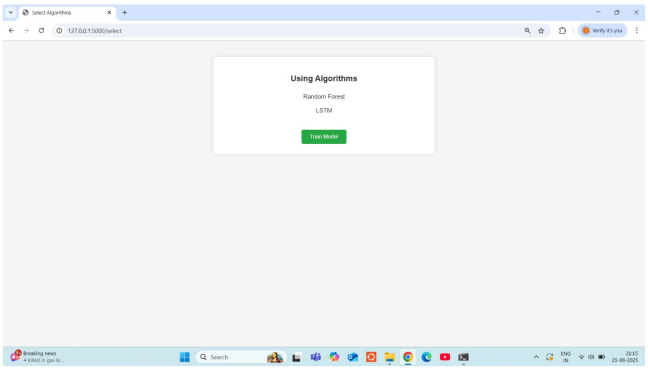
V. SCREENS



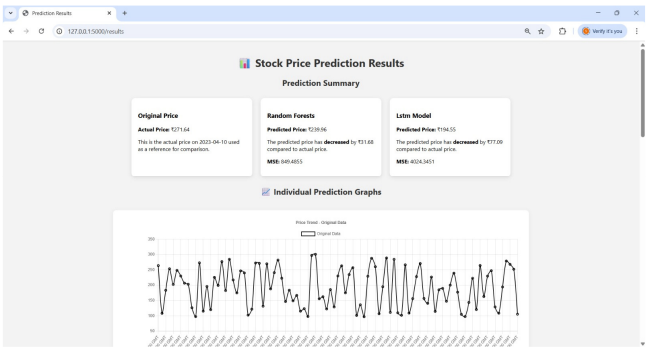
This is the entry page of the stock prediction tool, where the user uploads the dataset in CSV format. It acts as the starting point of the prediction workflow.



This screen allows users to choose and upload their stock market dataset. Once uploaded, the system preprocesses the data for model training.

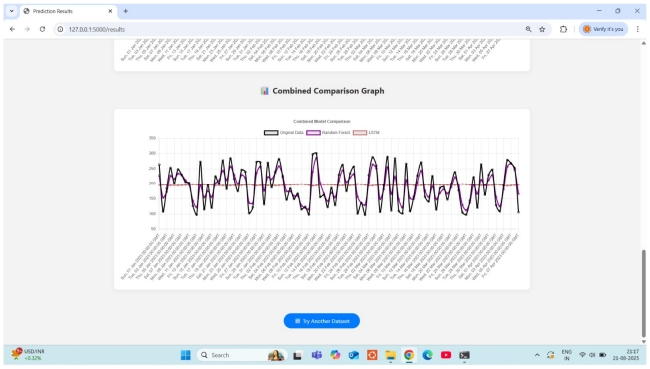


After uploading the dataset, the user selects the prediction algorithm (Random Forest or LSTM). The chosen model is then trained for stock price forecasting.



This output provides actual vs predicted stock prices along with Mean Squared Error (MSE). It shows that Random Forest achieved lower error compared to LSTM, making it more reliable on this dataset.

Here, Random Forest predictions capture short-term volatility well, while LSTM predictions follow smoother long-term trends. Both models demonstrate different strengths in forecasting.



This graph shows the comparison of predicted values from Random Forest and LSTM models against the actual stock prices. It highlights how closely each model follows real market fluctuations.

VI. RESULTS AND DISCUSSION

The models were evaluated using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Random Forest produced reliable results for short-term predictions, while LSTM demonstrated higher accuracy in capturing long-term sequential trends. Graphs of actual vs predicted stock prices illustrate the comparative performance of both models.

VII. CONCLUSION AND FUTURE SCOPE

This paper concludes that while Random Forest offers robustness against noisy data, LSTM is more effective in handling sequential dependencies, making it suitable for time-series stock forecasting. A hybrid approach may enhance prediction accuracy further. Future work could integrate sentiment analysis from social media and news data, expand datasets, and optimize deep learning architectures for improved forecasting.

REFERENCES

[1] J. Bollen, H. Mao, and X. Zeng, 'Twitter mood predicts the stock market,' Journal of Computational Science, vol. 2, no. 1, pp. 1–8,2011.
[2] 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.
[3] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques.Expert Systems with Applications,42(4),162–2172.
[4] Mittal, A., & Goel, A. (2012). Stock prediction using Twitter sentiment analysis. StanfordUniversity.

