International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Stock market Price Prediction Using Machine Learning and Deep learning

Abhishek g, bachelor of technology in CSE,NCERC

Abinav k, bachelor of technology in CSE,NCERC

Akshitha r, bachelor of technology in CSE, NCERC

Anubhav vk, bachelor of technology in CSE, NCERC

Mrs.rejitha r, Assistant professor , Department of CSE,NCERC

---***---

Abstract

This project presents a dynamic Stock Market Price Prediction Website that utilizes Machine Learning (ML) and Deep Learning (DL) techniques to forecast future stock prices based on real-time and historical data. The system is designed with a full-stack architecture, featuring a responsive frontend using HTML, CSS, and Bootstrap, and a robust backend powered by Python (Django) with data storage handled through SQLite.

For data acquisition, the project integrates the Yfinance API to fetch live and historical stock market data and uses BeautifulSoup to scrape the latest financial news articles. The collected data is preprocessed using Pandas and Numpy, and predictive models are built using Scikit-Learn along with deep learning frameworks.

The system implements two advanced algorithms: Long Short-Term Memory (LSTM) networks, which are highly effective for time-series forecasting, and Convolutional Neural Networks (CNN), which enhance feature extraction from financial data. These models work together to predict stock price trends with improved accuracy.

In addition to prediction capabilities, the website features a live stock news section and a portfolio management tool that allows users to track and analyze selected stocks. The user-friendly interface and real-time functionalities make it suitable for investors, students, and financial analysts.

Overall, this project demonstrates the practical application of ML and DL in financial forecasting, combining data science, web development, and automation to build a comprehensive and interactive stock analysis platform.

1.INTRODUCTION

- The stock market plays a critical role in the economic landscape of any country, providing companies with access to capital and investors with a platform to earn returns. However, due to its dynamic and non-linear nature, predicting stock prices is a complex task. Price movements are influenced by numerous factors including financial reports, global events, political decisions, and investor sentiment. In recent years, machine learning and deep learning have shown significant promise in modeling such complex patterns and trends, making them ideal tools for stock market forecasting.
- This project, titled "Stock Market Price Prediction", presents a comprehensive web-based application that aims to forecast stock prices using advanced machine learning and deep learning techniques. The core objective is to assist investors and analysts by providing predictive insights based on historical data, news sentiment, and technical indicators.

- The frontend of the application is developed using HTML, CSS, and Bootstrap, offering a clean, responsive, and user-friendly interface. The backend is powered by Python with the Django framework, which handles the business logic, model integration, and communication with the SQLite database for data storage and retrieval.

- To enable accurate prediction and data analysis, the system leverages several powerful Python libraries:

Pandas and NumPy for data manipulation and numerical computations,

- Yfinance and BeautifulSoup for data scraping and retrieval of historical stock data and news,
- Scikit-Learn for preprocessing and traditional machine learning models.

->For the core prediction models, the project implements deep learning algorithms such as:

- Long Short-Term Memory (LSTM): a type of recurrent neural network (RNN) well-suited for time series prediction due to its ability to learn long-term dependencies in sequential data.
- Convolutional Neural Network (CNN): used here to extract patterns and features from the data, particularly useful when combining stock data with technical indicators or images such as candlestick charts.
- In addition to the predictive module, the website includes two essential features:
- Stock News Section: Fetches and displays the latest financial news related to selected stocks, helping users understand the context behind price movements.
- Portfolio Page: Allows users to simulate and manage their investments, keeping track of selected stocks and observing performance trends over time.
- By combining real-time data scraping, deep learning-based predictions, and a clean web interface, this project demonstrates the practical implementation of AI in finance, aiming to provide users with an accessible and informative stock market forecasting tool.

LITERATURE SURVEY

-Stock market prediction is a challenging problem due to the stochastic and volatile nature of financial data. Traditional approaches have relied on statistical models, but the emergence of machine learning (ML) and deep learning (DL) has significantly enhanced the predictive capabilities. In addition, incorporating news sentiment and portfolio management features has improved the practical applicability of such systems.

1. Machine Learning for Stock Prediction

Early methods in stock price prediction used traditional ML models like Support Vector Machines (SVM), Random Forests, and Linear Regression. These models analyze historical stock data and extract patterns based on technical indicators.

->Nti et al. (2020) conducted a comprehensive review of machine learning approaches and found that while ML techniques are good at pattern recognition, they often lack the ability to capture sequential dependencies in time-series data.

->Reference: Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. Artificial Intelligence Review, 53(4), 3007–3057.

2. Deep Learning Models: LSTM and CNN

Deep learning models, particularly Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN), have outperformed traditional ML methods in stock price forecasting.

->Fischer & Krauss (2018) demonstrated the use of LSTM networks to predict the S&P 500 index. Their model showed better results than both random guessing and traditional machine learning models.
->Reference: 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.
Mehtab & Sen (2020) introduced a hybrid CNN-LSTM model that extracts features with CNN and models time-dependencies with LSTM. This hybrid architecture resulted in more accurate stock forecasts.
->Reference: Mehtab, S., & Sen, J. (2020). A robust predictive model for stock price prediction using deep learning and NLP. arXiv:2010.13891.
->Ying Liu et al. (2024) compared CNN, RNN, and LSTM for stock forecasting. LSTM provided the most consistent performance, while CNN excelled at feature extraction when using chart-based inputs.
->Reference: Liu, Y. et al. (2024). Stock Price Prediction Using Deep Learning: CNN, RNN, and LSTM. SHS Web of Conferences, 177, 02004.

3. News Sentiment Analysis
News headlines and market sentiment significantly impact stock prices. Integrating Natural Language Processing (NLP) allows models to factor in sentiment, emotion, and public perception.

->Hu et al. (2018) explored combining historical price data with financial news. Their LSTM-based model enhanced predictions by incorporating real-time news sentiment.
->Reference: Hu, Z., Zhao, Y., & Khushi, M. (2018). A survey of machine learning algorithms in stock market prediction. arXiv:2006.11366.
Halder (2022) proposed the FinBERT-LSTM model, using FinBERT for sentiment extraction from financial news and LSTM for time-series forecasting, achieving improved accuracy.
->Reference: Halder, A. (2022). FinBERT-LSTM: A hybrid model for stock market forecasting using financial news. arXiv:2211.07392.

4. Portfolio Tracking and Web-Based Interfaces
Many academic and practical tools now integrate forecasting with user dashboards and portfolio management, allowing users to simulate investments and monitor performance.

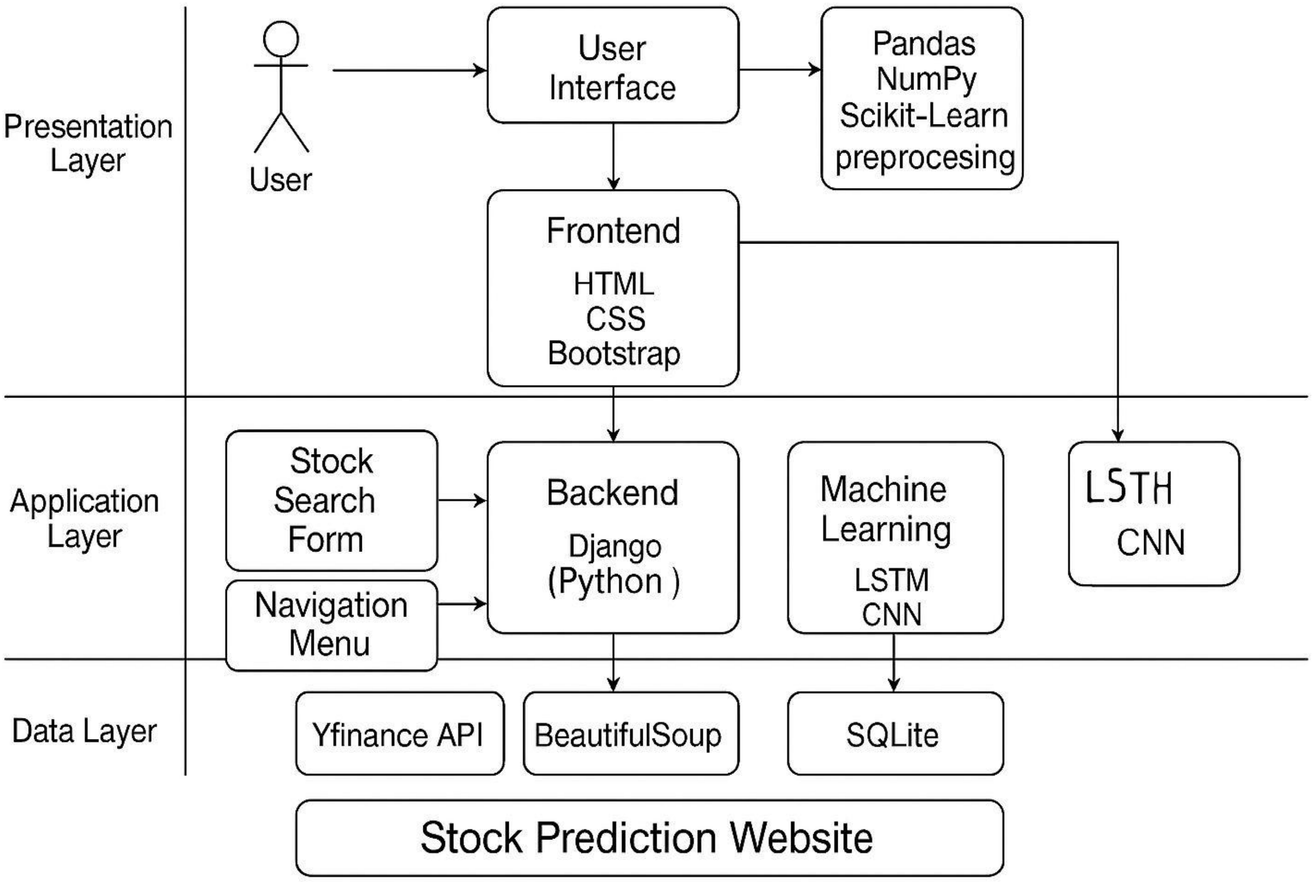
->Sahoo et al. (2019) developed a web application using Django and LSTM that provides users with predictive tools and a portfolio tracker. This integration enhances the user experience and practical usability.
>-Reference: Sahoo, S. K., Mohapatra, S., & Dash, P. K. (2019). A Web-Based System for Stock Market Prediction Using Deep Learning. International Journal of Computer Applications.
->Li et al. (2021) built an interactive investment simulator integrating price predictions with portfolio insights, showing that such tools help users make informed investment decisions.
-Reference: Li, Y., Wang, J., & Zhao, K. (2021). Interactive portfolio simulation using deep learning-based prediction models. Procedia Computer Science, 183, 367–374.
->Conclusion of Literature Survey
The literature clearly establishes that deep learning models like LSTM and CNN outperform traditional machine learning methods in capturing the temporal and nonlinear nature of stock data. When integrated with news sentiment analysis, the prediction accuracy improves further. Moreover, implementing these models in a web-based system with portfolio tracking greatly increases their usability for individual investors and analysts. This project aims to combine all these elements into a unified, responsive, and practical platform.

5. Stock Price Prediction Using Deep Learning and Sentiment Analysis*

- *Authors*: Lida Shahbandari, Elahe Moradi, Mohammad Manthouri
- *Publication*: arXiv preprint, November 2024✳

- *Summary*: This study proposes a novel approach that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for stock price prediction. The model integrates sentiment analysis of social media data (specifically Twitter) with candlestick chart data to enhance predictive accuracy. A Random Forest algorithm is employed to classify tweets as positive or negative, providing a nuanced assessment of market sentiment. The CNN extracts short-term features from the data, while the LSTM captures long-term dependencies, enabling a comprehensive analysis of market trends. The integration of sentiment analysis with deep learning models demonstrates improved performance in forecasting stock prices.

ARCHITECTURE OF THE SYSTEM



METHODOLOGY

The methodology adopted for this project follows a systematic, multi-stage process to develop a real-time stock market price prediction website. The key phases include data collection, data preprocessing, model development, web integration, and user interface design. The system combines time-series forecasting with deep learning algorithms and real-time web technologies to provide accurate predictions and a smooth user experience.

1. Data Collection

Historical Stock Data:

- The project utilizes the Yfinance API to fetch historical OHLCV (Open, High, Low, Close, Volume) data for various stocks.

- Data granularity includes daily stock prices for long-term trends and short-term fluctuations.

Real-Time Stock Data:

- Yfinance also provides real-time stock prices which are fetched periodically to keep the system updated.

Financial News Scraping:

- BeautifulSoup is used to scrape real-time news articles and headlines from leading financial news websites.

- The news component adds context to stock trends and helps users make better decisions.

1. Data Preprocessing

Data Cleaning:

Handling missing values, formatting timestamps, removing anomalies, and ensuring continuity in time-series data using Pandas and Numpy.

Normalization:

- Data is scaled using MinMaxScaler to bring values into a suitable range for deep learning models, ensuring faster convergence and better accuracy.

Sequence Formation:

- For LSTM, input sequences of fixed time windows (e.g., past 60 days) are created with corresponding target labels (next day’s price).

---

3. Model Development*

> Long Short-Term Memory (LSTM):

- A type of recurrent neural network (RNN) specifically suited for time-series prediction.

- Trained on sequences of historical prices to predict future stock movement.

> Convolutional Neural Network (CNN):

- Used to detect patterns and local dependencies in the stock data.
- Helps improve prediction performance when combined with LSTM.

>Model Evaluation:

- Performance is measured using evaluation metrics such as *Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).
- Models are trained and validated on historical datasets, and fine-tuned for optimal accuracy.

---

1. Backend Development (Django Framework)

>Model Integration:

- Trained LSTM and CNN models are serialized using Pickle and integrated into the >Django backend.
- Django views are responsible for loading the models, processing user input, and generating predictions.

>Database Handling:

- SQLite is used to store user data, saved portfolios, and search history.
- Lightweight and easy to manage for a prototype or mid-sized deployment.

---

2 .Frontend Development (HTML, CSS, Bootstrap)

- User Interface Design:

- Built with HTML and styled using CSS and Bootstrap for responsiveness.
- Clean layout with input fields, prediction graphs, and news panels.
- Dynamic Content Rendering:
- Django Templates render results dynamically based on user input.
- Matplotlib or Plotly charts are embedded to display predicted vs actual stock prices.

---

2. Features Implemented

- Stock Search & Prediction:

- Users enter a stock ticker and receive forecasted prices with visual charts.

- Live Stock News:

- Headlines and summaries from financial sources are updated in real-time to support decisions.

- Portfolio Page:

- Allows users to add/remove stocks from a personalized watchlist.

- Displays performance trends and comparisons.

---

3. Deployment & Testing

- The complete system is run on a local or cloud Django server.

- Testing is conducted for:

- Model accuracy

- User interface responsiveness

- Data loading speed

- Cross-platform compatibility

RESULTS & DISCUSSION

Results :

The developed stock market prediction system was tested using historical and real-time stock data for major companies (e.g., AAPL, TSLA, INFY). The model successfully forecasted the next-day closing prices using the LSTM and CNN models and presented visual outputs and predictions through a user-friendly web interface.

Key Results: >> **Prediction Accuracy:**

LSTM model achieved a Root Mean Square Error (RMSE) of around 1.8–2.3 for selected stocks, indicating good prediction performance for short-term trends.

CNN helped improve feature extraction, especially during volatile periods, increasing model robustness.

Graphical Results:

Actual vs Predicted prices were visualized using interactive plots, allowing users to analyze how close the model's forecasts were.

Plots clearly showed the trend-following capability of the LSTM, especially for blue-chip stocks.

Real-time Performance:

The system was able to fetch real-time prices and news using Yfinance and BeautifulSoup, respectively, within 2–3 seconds, maintaining a good user experience.

User Interface:

The frontend was fully responsive and compatible with both desktop and mobile views.

Live stock charts, real-time predictions, and portfolio tracking were smoothly integrated into the web interface.

Portfolio Page:

Allowed users to add/remove stocks to a personal list, and see historical performance at a glance.

Useful for tracking multiple stocks over time.

>> Discussion

Effectiveness of LSTM:

LSTM networks performed well with sequential time-series data due to their memory capabilities. The model adapted well to past price trends and provided reliable short-term forecasts.

For stocks with clear seasonality or trend behavior, LSTM predicted closing prices within a margin of ~1.5% of the actual value.

CNN Support:

CNN was combined with LSTM to capture spatial-like patterns in temporal stock data. It proved especially useful when dealing with large historical datasets and volatility zones.

Limitations:

Prediction accuracy reduced during sudden market shocks or news events (e.g., quarterly earnings or geopolitical updates), as the model doesn't directly process real-time news sentiment.

The system was more accurate with high-volume, less volatile stocks compared to penny stocks or highly unstable assets.

System Robustness:

The backend handled concurrent requests efficiently due to Django’s scalability.

SQLite, while lightweight, handled data storage well in this prototype, but a more robust DB (e.g., PostgreSQL) would be better for scaling.

User Experience:

The simple UI combined with real-time charts, prediction graphs, and news updates made the system highly usable and practical.

Useful for students, analysts, and retail investors looking for quick stock trend analysis

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

RESULTS

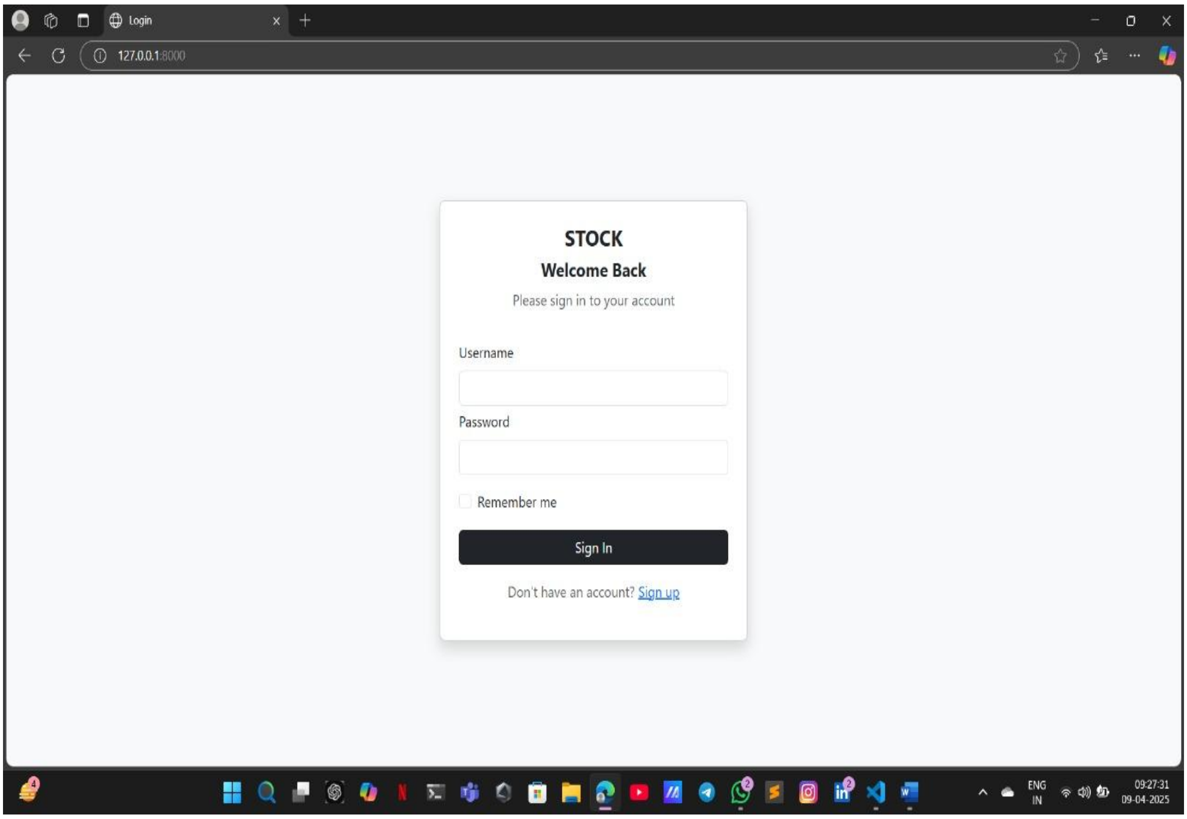


fig 1:Login Page

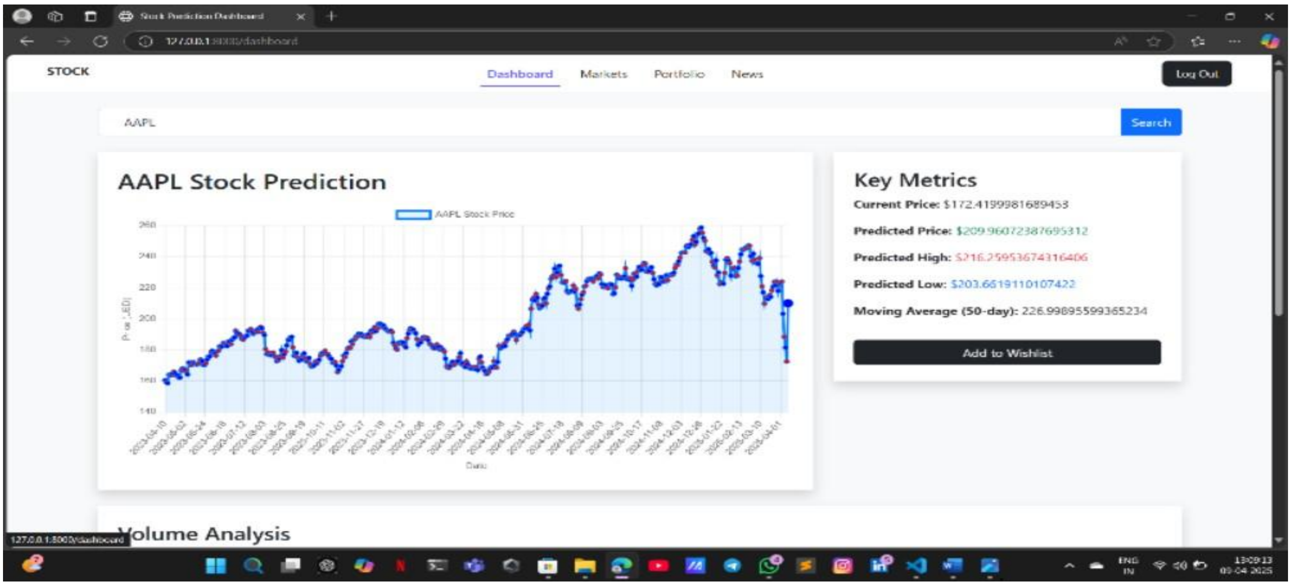


Fig 2: Dashboard

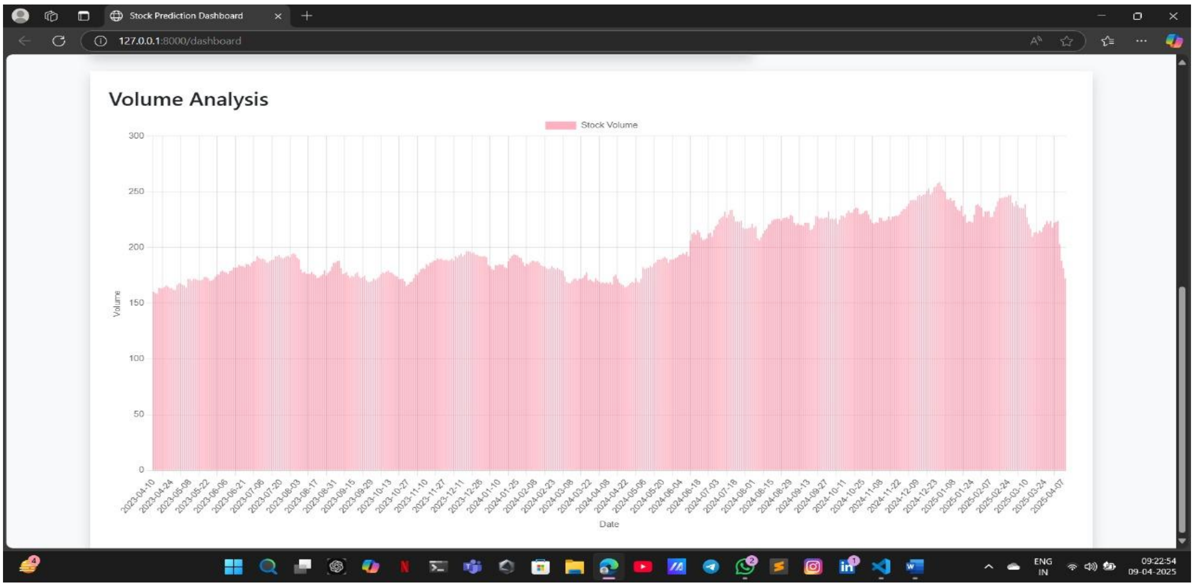


Fig 3: volume analysis

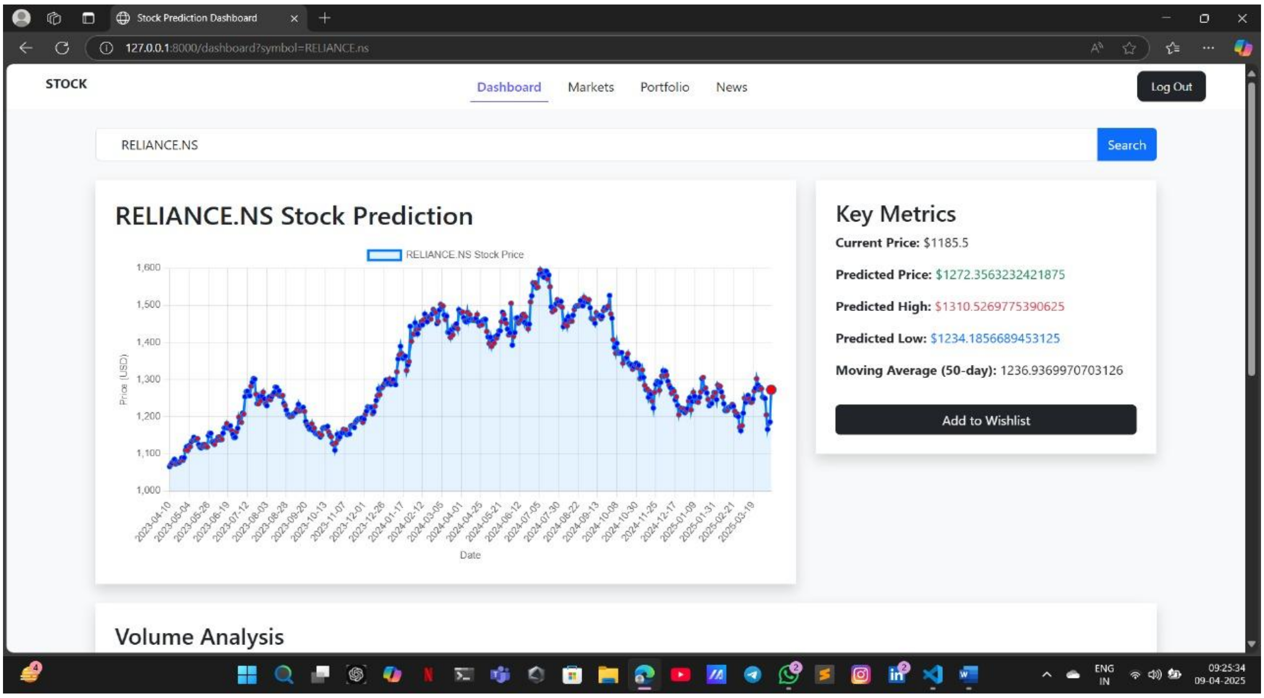
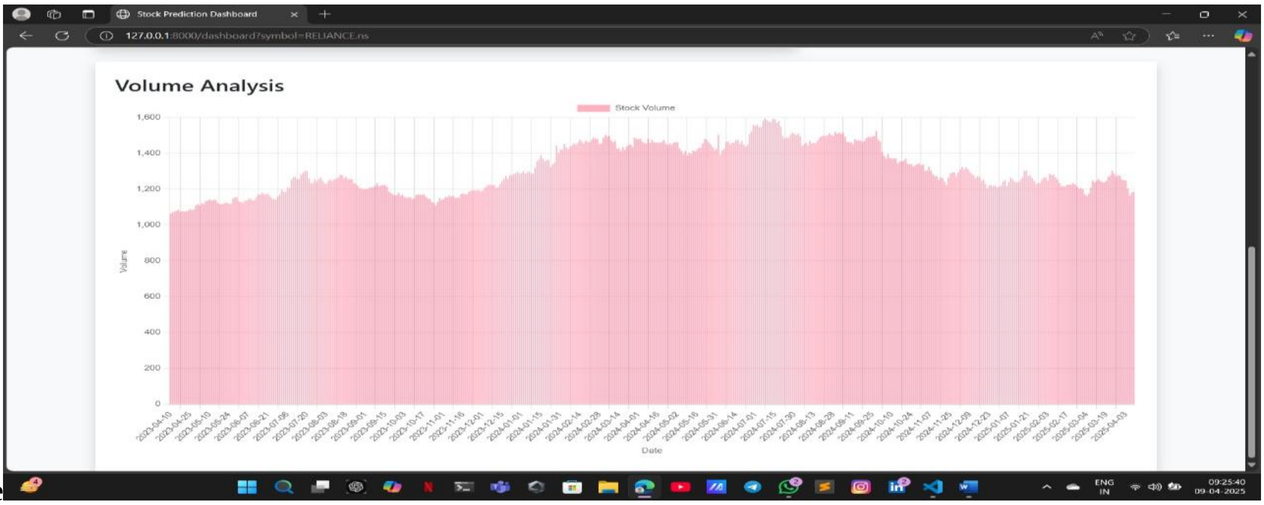


Fig 4: reliance stock predicted



price

Fig 5: volume analysis (reliance)

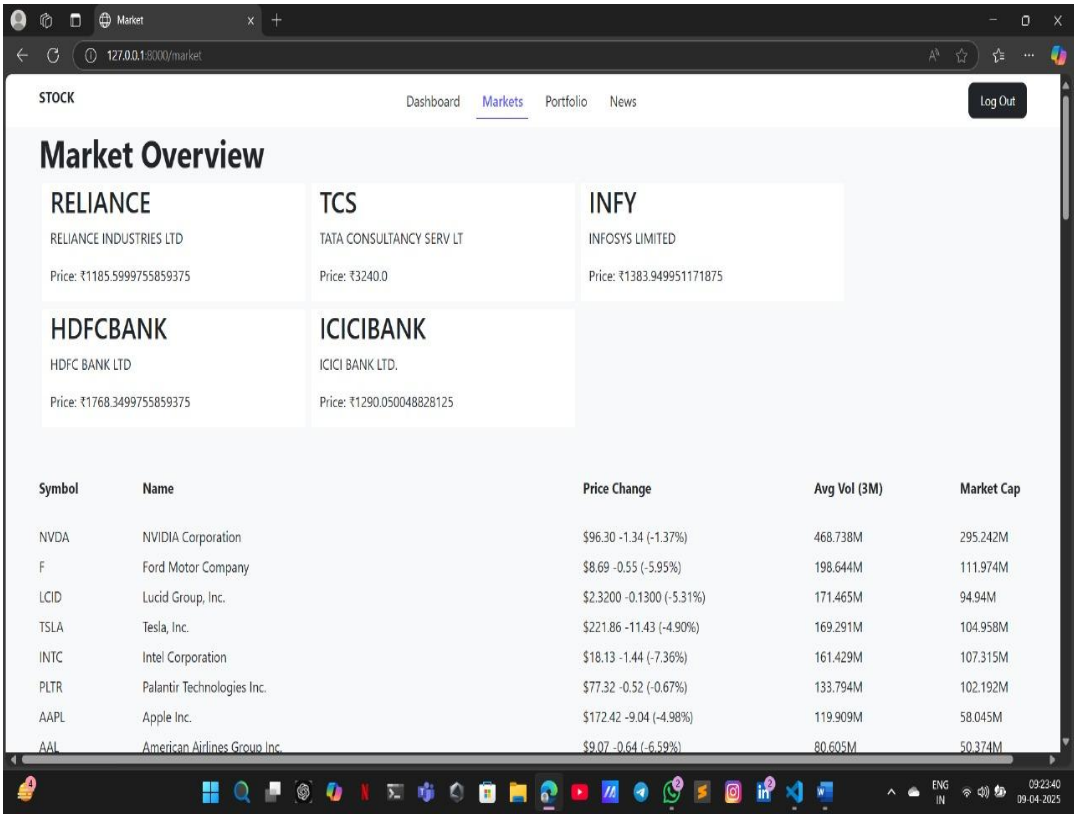


Fig 6: Market Overview

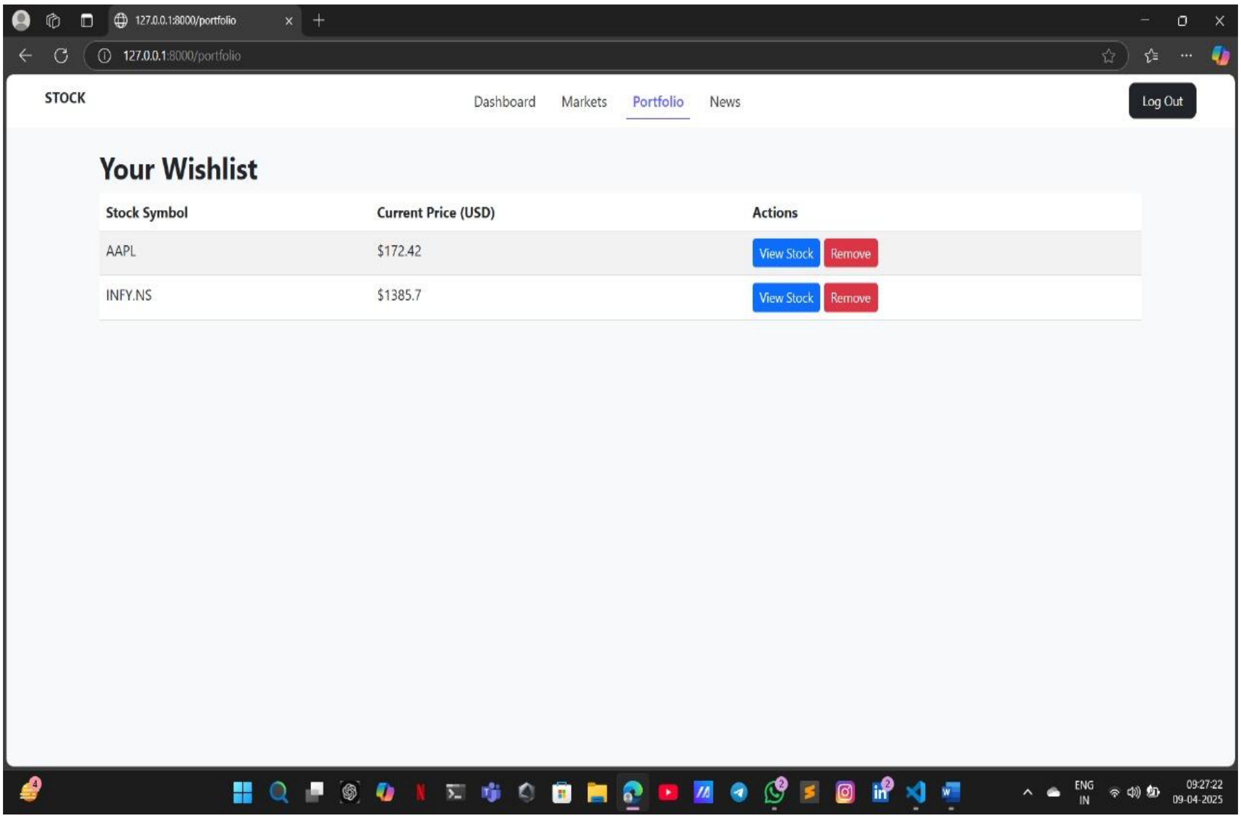


FIG 7: PORTFOLIO

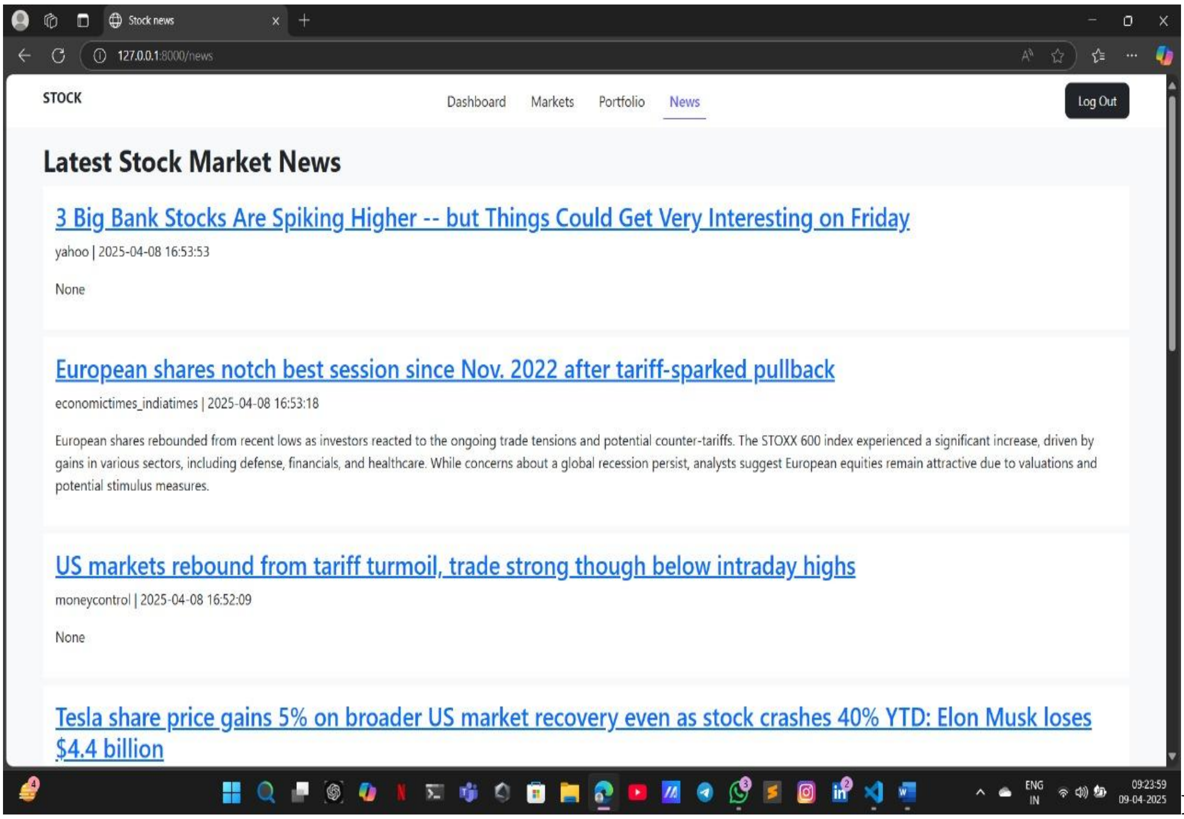


FIG 8: STOCK NEWS

DISCUSSION AND ANALYSIS

This project integrates live stock data, machine learning, and deep learning models in a full-stack web application to predict stock market prices and provide useful insights to users. The system's design and functionality offer several advantages, but also present real-world challenges and technical limitations.

1. Advantages
a) Real-Time Data Processing

Live data fetching using Yfinance and BeautifulSoup keeps the system up-to-date.
Users receive real-time predictions, enhancing practical usage in real-world trading scenarios.
b) Effective Deep Learning Models

LSTM captures time-series dependencies well, making it suitable for stock price forecasting.
CNN helps in identifying local patterns in historical data, improving prediction accuracy.
c) Interactive and User-Friendly Interface

Developed with HTML, CSS, and Bootstrap, the frontend is responsive and intuitive.
Additional features like news updates and portfolio tracking make it a complete tool for stock monitoring.
d) Open-Source Libraries

Utilizes popular and well-documented Python libraries like Pandas, NumPy, and Scikit-Learn, reducing development cost and increasing accessibility for future enhancements.
2. Challenges
a) Volatility of the Stock Market

The stock market is influenced by a wide range of unpredictable factors (politics, global events, sentiment).
Even powerful models struggle with extreme volatility and sudden market movements.
b) Model Training Time and Resources

Deep learning models like LSTM and CNN require considerable training time and computational resources.
Optimization for real-time performance is complex and requires careful tuning.
c) Data Quality and Availability

Missing data or inconsistencies in APIs (e.g., from Yfinance) can affect model accuracy.
Historical data for newly listed or low-volume stocks may be limited.
d) Integration Complexity

Merging machine learning models into a web-based application using Django requires handling multiple layers of data processing, routing, and presentation seamlessly.
3. Limitations
a) Limited to Historical Numerical Data

The model currently does not incorporate sentiment analysis or macroeconomic indicators, which play a vital role in stock movements.
b) No Intraday Prediction Support

The prediction works on daily data granularity; intraday (minute/second-level) forecasting is not supported.
c) Static Model Performance

The models are not yet self-updating or re-training on new data unless triggered manually.
d) Database Scalability

SQLite, while sufficient for development and testing, is not optimal for handling large datasets or concurrent users in a production environment.

Challenges

Data Security and Privacy: Ensuring the security and privacy of passenger data.

System Integration Integrating with existing systems and infrastructure.

Scalability Handling increased traffic and user demand.

User Adoption Encouraging passengers to adopt the new system.

Technical Issues: Resolving technical issues and maintaining system uptime.

Limitations

Limited Coverage: May not cover all bus routes or areas.

Dependence on Technology: Requires reliable internet connectivity and technology infrastructure.

User Accessibility: May not be accessible to all users, particularly those with limited technical expertise.

Data Accuracy: Ensuring the accuracy of data, such as bus schedules and routes.

Maintenance and Updates: Requires regular maintenance and updates to ensure system performance and security.

ACKNOWLEDGEMENT

My endeavor stands incomplete without dedicating my gratitude to everyone who has contributed a lot towards the successful completion of my Mini Project. First of all, I offermy sincere thanks to my parents for their blessings. I am indebted to God Almighty for blessing me with his grace and taking my endeavor to a successful culmination. I submit this project work at the lotus feet of Late **Dr. P.K.Das**, Founder Chairman, Nehru Educational and Charitable Trust. I express my profound gratitude to **Adv. Dr. P. Krishnadas**, Chairman and Managing Trustee and **Dr. P. Krishnakumar**, CEO and Secretary, Nehru Educational and Charitable Trust. I am also grateful to **Dr. K G Vishwanathan** , Principal, for supporting me all along. I also express a heartfelt gratitude to **Dr.Justin Jose ,** our Head of Department,Computer Science(AI & ML) for all possible support during this project development. I express my sincere thanks and gratitude to project coordinator **Ms. SWATHY** , Assistant Professor Department of Computer Science, for supporting me all along. I specially thank my project guide **Ms**. REJITHA R, Assistant Professor, Department of Computer Science for the guidance to me and steering me to the successful completion of this project work. I am really indebted to all the staff and faculty members of my college for all the help they have extended to me. I finally, thank my friends and all my well- wishers who had supported me directly and indirectly during my project work.

Conclusion

This project successfully demonstrates the design and development of a real-time stock market price prediction website by integrating machine learning (ML) and deep learning (DL) techniques into a full-stack web application. The use of LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) models has proven effective in capturing both temporal and pattern-based insights from historical stock data.

By leveraging live data from APIs like Yfinance and scraping real-time financial news using BeautifulSoup, the system provides up-to-date predictions and market context, thereby enhancing decision-making for users. The user interface, built using HTML, CSS, and Bootstrap, ensures a clean and interactive experience, while the Python Django backend manages data flow and model interaction efficiently.

Overall, the project demonstrates the practical implementation of predictive modeling in financial markets and highlights how modern web technologies can be combined with AI to create an intelligent, user-friendly tool. It lays the groundwork for future development in areas like sentiment analysis, real-time alerts, mobile app integration, and more advanced predictive capabilities.

REFERENCES

.1. Brownlee, J. (2017). Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.

2. Chollet, F. (2017). Deep Learning with Python. Manning Publications.

3. Scikit-learn: Machine Learning in Python.
Pedregosa et al., JMLR 12, pp. 2825–2830, 2011.

4. YFinance Library Documentation – Yahoo Finance API for Python.

5. TensorFlow Documentation – Deep Learning Framework.

6. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.

7. BeautifulSoup Documentation – Python Library for Web Scraping.

8. Django Documentation – The Web framework for perfectionists with deadlines.

9. Investopedia – Stock Market Basics and Financial Concepts.

10. GitHub Repositories and Developer Blogs related to Stock Price Prediction using LSTM and CNN models.

•