WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

Stock Sage

Riya Pillai[1], Sakshi Pachlaniya[2], Rashi Malviya[3], Niharika Jat[4]

[1,2,3,4] Computer Science and Engineering, Acropolis Institute of Technology & Research, Indore, India
[1]riyapillai210806@acropolis.in, [2] sakshipachlaniya210114@acropolis.in, [4]rashimalviya210210@acropolis.in, [4]niharikajat210152@acropolis.in
DOI : https://doi.org/10.55248/gengpi.6.0425.14122

ABSTRACT

This research presents a stock prediction system utilizing machine learning models, including Long Short-Term. Memory (LSTM), Random Forest, and Autoregressive Integrated Moving Average (ARIMA), along with technical indicators such as Moving Averages, Relative Strength Index (RSI), and Bollinger Bands. The system employs data preprocessing techniques and evaluates model accuracy using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). An interactive user interface (UI) with advanced visualizations enables dynamic exploration of stock trends, enhancing user experience and decision-making. By combining deep learning with traditional models.

***Keywords*— StockSage, stock prediction, machine learning, LSTM, Random Forest, ARIMA, technical indicators, Moving Averages, RSI, Bollinger Bands, financial forecasting, data preprocessing, predictive analytics, investment strategies, market trend analysis, and mainly based on Indian stock market.**

I. INTRODUCTION

StockSage is a stock prediction system utilizing machine learning models (LSTM, ARIMA) and technical indicators (Moving Averages, RSI, Bollinger Bands) to enhance market trend forecasting. It features data preprocessing, evaluates models using RMSE and MAE, and offers an interactive UI with visualizations for informed investment decisions and risk management. The system combines deep learning with traditional models to capture complex market patterns. Its intuitive interface empowers users to explore stock trends dynamically. StockSage aims to support robust investment strategies and effective financial planning.

II. PROBLEM FORMULATION

Predicting stock prices is a challenging task due to the volatile and dynamic nature of financial markets. Traditional forecasting methods often struggle to capture complex market patterns influenced by historical prices, market sentiment, and economic indicators. Investors and traders require reliable tools that not only provide accurate predictions but also present data in an accessible and insightful manner. StockSage addresses this need by formulating the problem as a time series prediction task, leveraging machine learning models such as Long Short- Term Memory (LSTM), and AutoRegressive Integrated Moving Average (ARIMA).

The system integrates technical indicators—including Moving Averages, Relative Strength Index (RSI), and Bollinger Bands— to enrich the feature set and improve prediction accuracy. StockSage evaluates model performance using metrics like Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), ensuring robust model selection. Additionally, the system features an interactive user interface (UI) with dynamic visualizations, enabling users to analyze stock trends effectively and make informed investment decisions. The system bridges the gap between raw market data and actionable insights.

III. LITERATURE REVIEW

A review of existing stock prediction systems highlights gaps in their ability to provide accurate forecasts and user-friendly insights:

A. Yahoo Finance:

Strengths: Free access to historical stock data and basic analytics.

Weaknesses: Limited predictive analytics; lacks advanced machine learning model integration.

B. Bloomberg Terminal:

Strengths: Comprehensive market data, financial analytics, and professional tools.

Weaknesses: High cost; complex interface, making it less accessible to casual or small-scale investors.

C. TradingView:

Strengths: Advanced charting tools and a vibrant community for trading ideas.

Weaknesses: Primarily focuses on technical analysis; lacks machine learning-based predictive models.

D. MetaStock:

Strengths: Strong technical analysis and forecasting tools.

Weaknesses: Limited AI-driven insights; requires steep learning curve for new users.

E. Google Finance:

Strengths: Simple interface with basic stock tracking and news aggregation.

Weaknesses: Minimal analytical tools; no support for predictive modeling or trend analysis.

StockSage combines advanced machine learning models with technical indicators to provide accurate stock predictions and dynamic visualizations.

IV. METHODOLOGY

The development of StockSage follows a systematic and user centric approach to ensure accuracy and effectiveness. This comprehensive process is designed to meet the diverse needs of investors while maintaining scalability, security, and innovation. By adopting a phased methodology, every aspect of the project is carefully planned and executed to align with user expectations and technical feasibility.

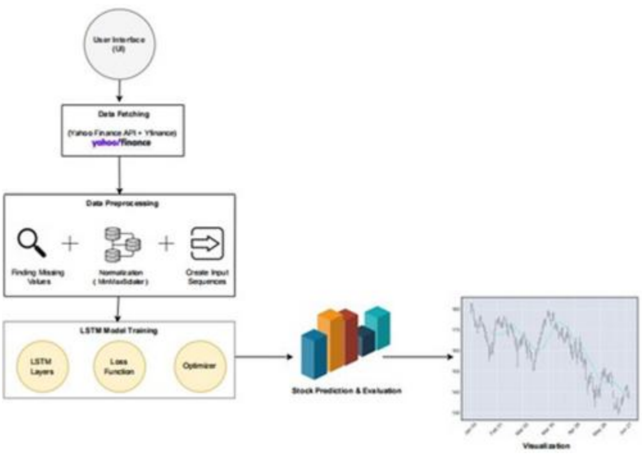
A. Requirement Gathering

The process of building StockSage begins with an in-depth analysis of user requirements. This involves conducting interviews, surveys, and feedback sessions with target audiences, includingtraders, investors, and financial analysts. Functionalrequirements are identified, including integration of machine learning models (LSTM, Random Forest, ARIMA), incorporation of technical indicators (Moving Averages, RSI, Bollinger Bands), an interactive UI. Non- functional requirements such as system scalability, real-time data processing, and robust security measures are also prioritized. The insights gathered during this phase help shape the project’s objectives and ensure alignment

with user needs.

B. System Design

StockSage is designed as a stock price prediction system with Streamlit powering the interactive front-end interface. It fetches 20 years of historical stock data using the yfinance API, enabling robust financial trend analysis. Data is cleaned and visualized using Pandas and Matplotlib, while Scikit-learn handles scaling for model training. An LSTM neural network, built with Keras and TensorFlow, learns temporal patterns to forecast future stock prices. The app visualizes predictions versus actual prices and evaluates performance using Root Mean Squared Error (RMSE)



C. AI Integration

Artificial intelligence powers StockSage’s core, using advanced models like LSTM and ARIMA for accurate stock predictions. It integrates technical indicators such as Moving Averages, RSI, and Bollinger Bands to deepen market insight. A robust analytics engine enables real-time data processing for efficient trend analysis and forecasting. The AI-driven system automates complex tasks, boosting accuracy and saving users time. By leveraging these tools, StockSage helps investors make smarter, more informed decisions.

D. Testing

Testing is an integral part of StockSage's development cycle, ensuring that the platform meets high standards of accuracy and reliability. Unit testing focuses on validating individual components, such as the data preprocessing pipeline, prediction models, and visualization modules. Each component is rigorously tested in isolation to identify and resolve issues early in the development process.

Cross-Browser Compatibility Testing

Test ID	Objective	Test Scenario	Expected Outcome
CBC - 01	Verify design consistency	Load the on Chrome, Firefox, Safari	Consistent layout and functionalit-y
CBC - 02	Validate feature functionali-ty	Interact with buttons, links, and visualize	Features perform identically across browsers

Database Testing:

Test ID	Objective	Test Scenario	Expected Outcome
DBT - 01	Validate data integrity	Retrieve historical stock data	Accurate and consistent data
DBT - 02	Test data flow	Store processed data for model use	Proper format and usability

Security Testing

Test ID	Objective	Test Scenario	Expected Outcome
ST - 01	Verify end-to-end flow	Run full stock prediction	All steps run successfully
ST - 02	Check UI-model integration	Select stock and view prediction in UI	Prediction displays properly

Usability and User Experience Testing:

Test ID	Objective	Test Scenario	Expected Outcome
UX-01	Check UI usability	Select stock and date	Smooth navigation
UX-02	Check visual clarity	View graphs and predictions	Clear visuals displayed

System testing in StockSage ensures smooth integration of the UI, ML models, and data pipelines. End-to-end tests simulate real use, like fetching stock data, preprocessing, and generating predictions. These assess system stability under different inputs and conditions. Scenarios include stock selection, date filtering, and result visualization. Usability testing collects feedback to improve clarity, navigation, and user experience.

E. Deployment

StockSage is deployed using Streamlit, allowing it to run as a web-based application accessible from any modern browser. The model and data preprocessing scripts are integrated within the Streamlit app, enabling end-to-end execution without separate server configurations. Deployment involves hosting the application through platforms like Streamlit Cloud, which manages the backend infrastructure. This approach ensures seamless updates, scalability, and ease of access for users. The streamlined deployment process enables real-time predictions and interactive visualizations with minimal latency. Here, are some points that describes deployment of StockSage better:

Streamlit-based web app Real-time execution Auto env setup

Live data via yfinance GitHub for updates

All components run seamlessly within the Streamlit environment, requiring no external server setup. Dependencies are auto-installed using a requirements.txt file for easy deployment. Live data fetching via (yfinance) removes the need for manual data updates or storage.

V. RESULT DISCUSSIONS

Advanced AI Features:

StockSage uses LSTM models to analyze historical stock data and generate accurate price predictions, outperforming basic statistical methods.

User-Centric Design:

The Streamlit-based interface enables users to input stock symbols, set date ranges, and view predictions through clear and interactive graphs.

Lightweight Architecture:

The app performs data fetching, preprocessing, and prediction in real time using yfinance, with no need for external databases or complex backend setup.

Comparative Advantages:

Unlike traditional tools, StockSage offers integrated technical indicators and real-time AI forecasting, making it a more dynamic and insightful solution.

Practical Impact:

The platform reduces manual analysis and helps investors make quicker, data-driven decisions with confidence and ease.

VI. CONCLUSION

StockSage offers a powerful, AI-driven solution for stock p.rice prediction, combining LSTM models with technical indicators to deliver accurate insights. Its Streamlit-based interface ensures ease of use, while real-time data processing enhances decision-making. With a lightweight, scalable architecture and no reliance on external databases, StockSage is both efficient and accessible. Overall, it empowers investors with reliable forecasts and interactive tools to make informed, data-driven investment strategies.

ACKNOWLEDGMENT

We extend our sincere gratitude to our supervisor, Prof. Ritika Bhatt, whose guidance and expertise were instrumental in the success of this project. We also acknowledge the collaborative efforts of our development team and the invaluable feedback from our peers and testers. Their contributions have shaped StockSage into a robust and user-friendly platform.We are thankful for the learning environment provided by our institution.Special thanks to the creators of open-source tools like Streamlit and yfinance.

REFERENCES

Brownlee, J. (2017). Long Short-Term Memory Networks With Python. Machine Learning Mastery. [i]

Chollet, F. (2015). Keras: Deep Learning Library for Theano and TensorFlow. GitHub Repository. [ii]

Abadi, M. et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems https://www.tensorflow.org [iii]

yfinance – Yahoo! Finance Market Data Downloader. https://py pi.org/project/yfinance/ [iv]

Streamlit – The fastest way to build data apps.https://streamlit.io/ [v]

Pandas: Python Data Analysis Library. https://pandas.pydata.org/[vi]

Matplotlib– Visualization with Python. https://matplotlib.org/[vii]

NumPy– Fundamental package for scientific computing with Python. https://numpy.org/ [viii]

Scikit-learn: Machine Learning in Python. https://scikit-learn.org/[ix]