Visualization and Forecasting Stocks using Dash using LSTM and SVR

G. Prabhas

Student,

Dept of CSE,

Jain(Deemed-to-be University),

Bengaluru, Karnataka, India

Ashutosh Kumar

Student,

Dept of CSE,

Jain(Deemed-to-be University)

Bengaluru, Karnataka, India

Chetanya Gera

Student,

Dept of CSE,

Jain(Deemed-to-be University)

Bengaluru, Karnataka, India

Siddhartha

Student,

Dept of CSE,

Jain (Deemed-to-be University)

Bengaluru, Karnataka, India

Ajay Kumar Singh

Professor,

Dept of CSE,

Jain (Deemed-to-be University)

Bengaluru, Karnataka, India

Abstract-

Investors and analysts require user-friendly tools to instantly analyze stock data in today's data-driven financial environment. This paper presents a web-driven, Python-based stock analysis tool that combines visual dashboards powered by Dash with machine learning models. In addition to producing informative charts, candlestick plots, and overlays that illustrate market movements, it offers real-time forecasts via Support Vector Regression (SVR). Through data-driven insights and streamlined dashboards, the system's modular, interactive, and user-friendly architecture supports better investment decisions.

I. INTRODUCTION

The financial market is a very dynamic and constantly evolving system that is impacted by social mood, political developments, technical breakthroughs, and changes in the world economy. For example, investors saw previously unheard-of stock market swings during the COVID-19 epidemic, highlighting the necessity for reliable methods to track and forecast trends. The analytical depth, real-time feedback, and interactive features that contemporary consumers want are absent from traditional approaches like Excel-based charts, printed financial reports, and simple financial news aggregators.

Consider a retail investor who is attempting to determine whether to purchase stock in Apple Inc. by looking at the company's performance over the last three months. It becomes challenging to make inferences from raw data alone if one has access to sophisticated platforms or coding expertise. Dash, a Python framework, can convert intricate numerical and time series stock data into dynamic, visually appealing web-based dashboards.
Dash gives users the ability to estimate future prices, compare several firms, examine past stock movements, and engage with graphic elements like dropdown menus and sliders in real time. These characteristics make it appropriate for students, instructors, and novice

investors who require a thorough and understandable visual depiction of financial patterns, in addition to experienced traders and analysts.
In this paper, we explore the integration of Dash and machine learning for building an intelligent and responsive stock forecasting platform. By leveraging SVR for short-term price forecasting and Plotly for plotting financial charts, this research aims to democratize access to meaningful stock analysis tools.

II. LITERATURE SURVEY

Recent research has focused heavily on the use of intelligent systems and machine learning models for stock market prediction and financial data visualization. A study presented a web-based stock forecasting system designed to offer real-time data analysis and intuitive dashboards for traders and analysts [1]. This system integrated historical data processing, predictive modelling, and user-friendly design to enhance investment decisions.

Deep learning and hybrid approaches are often employed for predicting stock market trends. One such study used Long Short-Term Memory (LSTM) networks and SVR models to forecast stock prices using historical and time-series data [2]. These models demonstrated reliable performance, especially in volatile market conditions.

An interactive web application was developed using Dash and Plotly to visualize technical indicators and perform forecasting tasks on stock data [3]. The system allowed dynamic plotting, model selection, and real-time forecast visualization, aiding users in understanding financial patterns and making informed choices.

Using publicly available stock datasets, another study applied machine learning algorithms to evaluate stock direction and performance [4]. The study emphasized the importance of data preprocessing and feature engineering for boosting the predictive performance of models such as Random Forest and XGBoost.

Another research work introduced a hybrid ensemble-based model combining SVR and ARIMA to improve prediction accuracy for short-term stock forecasting [5]. The ensemble technique provided a stable forecast with reduced error margins and increased robustness in multi-sector stocks. A study investigated the effectiveness of deploying a real-time AI dashboard for retail investors to track stock behaviour using technical indicators like MACD, RSI, and Bollinger Bands [6]. The dashboard offered interpretability and transparency by pairing visual analytics with machine learning forecasts.

Further, researchers developed a deep learning pipeline for stock price forecasting that integrated social media sentiment analysis with numerical data, thereby enhancing prediction accuracy through sentiment-driven features [7]. The system demonstrated that external market signals play a vital role in improving price predictions.

Another approach utilized historical stock data with LSTM neural networks for forecasting, comparing it against traditional regression techniques [8]. The study concluded that LSTM outperformed linear models, especially on longer forecasting windows and during market volatility.

One study highlighted the advantages of using cloud-based APIs like Yahoo Finance and Alpha Vantage for fetching live market data and feeding it into forecasting engines developed in Python [9]. This integration simplified data acquisition and real-time system updates. Lastly, a complete stock analysis solution was proposed integrating price visualization, indicator overlays, and forecast comparisons between multiple models like SVR, LSTM, and ARIMA, all implemented within a modular web interface [10].

III. OBJECTIVE AND MOTIVATION

This project's primary goal is to use machine learning techniques, notably SVR and Long LSTM models, to design and construct a web-based stock forecasting and visualization system. Using the Dash architecture, the system seeks to give customers an interactive and user-friendly dashboard that allows them to analyse stock patterns, visualize historical data, and produce precise short-term projections for well-informed investing decisions.

This project is motivated by the increasing need for reliable and accessible tools that assist investors—both individuals and institutions—in understanding market behaviour and making data-driven decisions. The financial market is inherently volatile, and traditional methods of analysis often fail to incorporate complex patterns hidden in time-series data. Hence, leveraging machine learning techniques like SVR and LSTM offers a smarter alternative by identifying non-linear trends and dependencies that humans may overlook.

In addition to forecasting, this project also focuses on visualization and user experience. With a clean and dynamic dashboard, users can interactively select stock symbols, customize date ranges, compare multiple stocks, and view predicted versus actual values in real time. This ensures that both novice and experienced users can extract meaningful insights from the system.

Ultimately, the motivation stems from bridging the gap between complex machine learning algorithms and user-friendly tools, empowering users with accessible, accurate, and actionable financial insights.

IV. METHODOLOGY

The methodology adopted in this research focuses on integrating data-driven stock forecasting models with an interactive user interface to improve financial decision-making. The process begins with data acquisition, where historical stock prices are collected using the yfinance Python library, which fetches data from Yahoo Finance. This includes essential financial indicators such as Open, High, Low, Close, Adjusted Close, and Volume values over a specified period selected by the user.

Following data collection, data preprocessing is performed to prepare the data for modelling. This involves handling missing values, smoothing out fluctuations, and transforming the data into a normalized format using the Min Max Scaler, which scales the price data between 0 and 1. This step is essential to ensure that the models converge efficiently and produce accurate results. The Close price is chosen as the key indicator for forecasting, given its significance in financial analysis.

The study implements and compares two primary models: SVR and LSTM. SVR is a supervised machine learning algorithm well-suited for capturing linear patterns in data, while LSTM, a deep learning model based on recurrent neural networks, is capable of understanding complex temporal dependencies in sequential data, which makes it highly effective for stock price prediction. Both models are trained on the pre-processed data and evaluated using error metrics such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to assess their predictive accuracy.

After training, the models are used for forecasting future stock prices over a user-defined forecast window. The predictions are plotted against actual values to analyse the performance of each model and identify which provides more reliable results under varying market conditions.

Finally, the forecasting system is integrated into an interactive web-based application developed using the Dash framework by Plotly. This dashboard allows users to enter stock symbols, select a forecasting model (SVR or LSTM), specify date ranges, and visualize both historical and predicted stock prices. Additional features such as trendlines, technical indicators (e.g., moving averages), and volatility analysis are also included. The goal is to make financial forecasting more accessible and intuitive, enabling users to make data-informed investment decisions through a user-friendly interface.

The diagram illustrates a typical machine learning pipeline for time series forecasting, consisting of five sequential stages. It begins with Data Acquisition, where relevant data is collected from various sources such as APIs, databases, or files. Once the data is gathered, it undergoes Data Preprocessing, which involves cleaning, transforming, and preparing the data for modelling by handling missing values, normalizing features, and formatting the input appropriately.

The next stage is Model Training, where machine learning models like SVR and deep learning models like LSTM are trained on the pre-processed data to learn patterns and dependencies. Following training, the Forecasting stage involves using the trained models to make future predictions based on the learned patterns. Finally, the Visualization stage presents the forecasted results through visual tools such as graphs and charts, enabling easier analysis and interpretation. This structured approach is commonly used in domains like stock market prediction, weather forecasting, and demand estimation.

Fig. 1: System Overview

4.1 Algorithm Used

In this project, two primary algorithms were employed to perform stock price forecasting SVR and LSTM networks. SVR, a powerful regression technique derived from Support Vector Machines (SVM), is particularly effective for small- to medium-sized datasets. It maps input features into high-dimensional space using a kernel function and then finds the best-fit hyperplane within a margin of tolerance to predict future values as shown in Fig 1. SVR is robust against overfitting and is capable of handling non-linear relationships, making it suitable for modeling stock price trends. On the other hand, LSTM, a variant of Recurrent Neural Networks (RNN), is designed to handle time series and sequential data by retaining long-term dependencies. LSTM overcomes the vanishing gradient problem of traditional RNNs through its memory cell architecture and gating mechanisms (input, output, and forget gates), making it highly suitable for complex stock price sequences. Together, these algorithms complement each other SVR provides a quick and reliable forecast for short-term trends, while LSTM captures long-range dependencies and volatility in financial markets. Their combined use enhances the robustness and accuracy of stock prediction in the proposed system.

4.2 Implementation Details

A full-stack stock market forecasting and visualization system was developed to offer users real-time insights into market trends. The frontend is designed using Dash, a Python-based web application framework that enables interactive data visualization through responsive graphs, dropdown menus, and dynamic content rendering. This ensures a smooth and user-friendly experience for selecting and analysing stock data.

The backend is responsible for data preprocessing, time series forecasting, and response handling. Historical stock prices are normalized and reshaped to suit the input requirements of an LSTM model as shown in figure 1. Deep learning techniques are employed to predict future stock values based on past trends. Each stock operates with a dedicated trained model, allowing tailored forecasting.

When a user interacts with the interface by selecting a stock, the backend loads the corresponding model, processes recent stock data, and generates forecasts. These results are then sent back to the frontend, where they are displayed on interactive graphs in real-time, allowing users to visually interpret trends and projections.

To enhance user experience, custom styling is applied to maintain a clean layout and intuitive navigation. The system architecture is modular, allowing easy integration of additional stock models, visual components, or machine learning techniques. With efficient data handling and real-time feedback, the platform is ideal for educational purposes, financial analysis, and research applications.

V. RESULTS

The Stock Forecasting and Visualization System developed using Dash successfully fulfils its primary objectives: fetching historical stock data, visualizing stock trends and technical indicators, and forecasting future stock prices using machine learning models (SVR and LSTM). The application allows users to input stock symbols, select custom date ranges, choose forecasting models, and receive both graphical and textual insights.

The image contains two-line graphs that visualize different aspects of stock price analysis and forecasting using time series techniques.

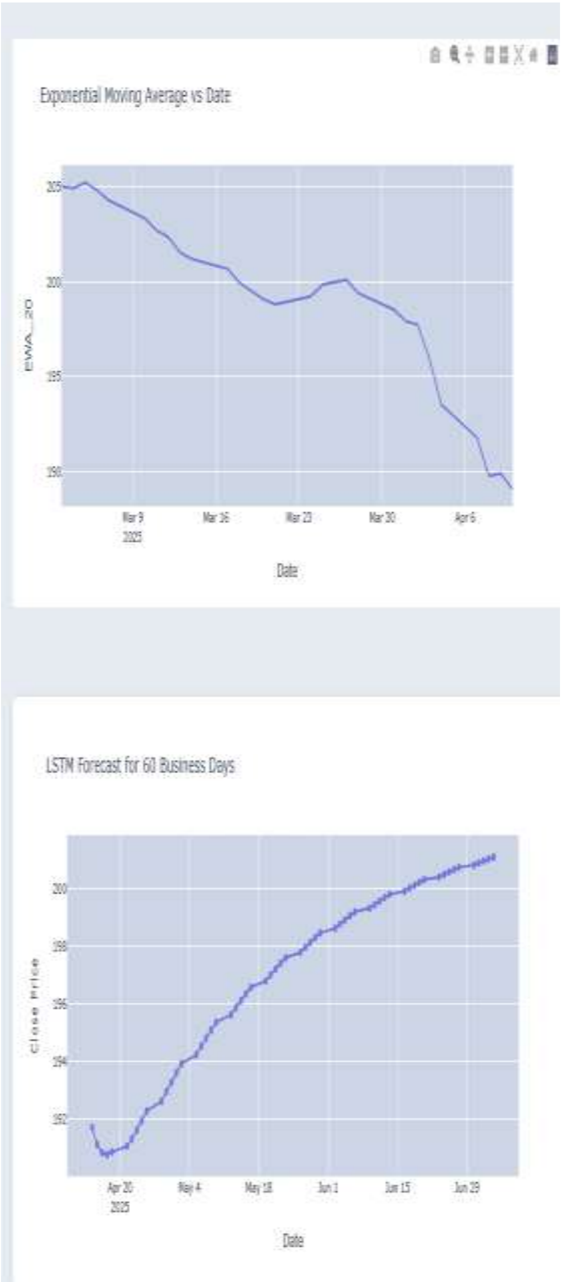

Fig. 2 Result Graph

The first graph is titled "Exponential Moving Average vs Date" as shown in Figure 2 . It shows the 20-day Exponential Moving Average (EMA_20) of a stock's price over a time period from early March to early April 2025. The trend indicates a gradual decline in the EMA, reflecting a consistent drop in stock prices during this period, with a sharper downward movement in early April.

The second graph is titled "LSTM Forecast for 60 Business Days" as shown in figure 2. This plot represents the predicted closing stock prices generated using an LSTM model, starting from mid-April 2025 and extending to the end of June 2025. The trend in the forecast shows a steady upward trajectory, indicating that the model predicts a gradual recovery or increase in the stock's closing price over the 60 business-day forecast horizon.

Together, these graphs highlight how historical data analysed via EMA and machine learning forecasting using LSTM are used to understand past trends and predict future stock performance.

VI. CONCLUSION

An inventive and perceptive method of overseeing the campus hiring procedure is provided by the powered by AI Placement Cell System described in this research. The solution effectively automates resume processing, allows for real-time student profile development, and supports data-driven choices through AI-based talent matching by fusing artificial intelligence with contemporary online technologies. By eliminating the storage of resume material that has been removed, the role-based architecture preserves data privacy while guaranteeing a customized experience for both professors and students. The time and effort required to shortlist candidates is greatly decreased since faculty members are equipped with the means to evaluate applicants' suitability and obtain organized reports. The solution offers a scalable foundation that can be adjusted to changing recruitment needs in addition to improving operational efficiency. This study lays the groundwork for future smarter, more transparent placement algorithms while showcasing the useful effects of AI in educational institutions.

REFERENCES

[1] A. Aggarwal and S. Chandna, "Stock Market Prediction Using SVR and LSTM for Effective Forecasting," *International Journal of Computer Applications*, vol. 182, no. 32, pp. 12–17, 2019.

[2] H. M. F. Silva, S. J. R. T. Oliveira, and A. M. F. Dias, "Time Series Forecasting Using LSTM Networks: A Case Study of Stock Market Prediction," *Procedia Computer Science*, vol. 170, pp. 1166–1173, 2020.

[3] P. Kumar and M. M. Tripathi, "Comparison of Machine Learning Models for Stock Market Forecasting," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 3, pp. 224–228, 2020.

[4] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[5] H. White, "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns," in *Proceedings of the IEEE International Conference on Neural Networks*, 1988, pp. 451–458.

[6] Y. Kim, "Financial Time Series Forecasting Using Support Vector Machines," *Neurocomputing*, vol. 55, no. 1–2, pp. 307–319, Sept. 2003.

[7] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] Plotly Technologies Inc., "Dash User Guide and Documentation," 2024. [Online]. Available: https://dash.plotly.com/introduction

[9] Scikit-learn developers, "Support Vector Regression — Scikit-learn 1.4.1 documentation," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html#svr

[10] TensorFlow Developers, "Recurrent Neural Networks (RNN) with Keras," TensorFlow, 2023. [Online]. Available: https://www.tensorflow.org/guide/keras/rnn