Real-Time Stock Price Forecasting Web Application: A Deep Learning Approach with LSTM Network

[1st]Rithik Chaudhary, [2nd]Vishal Rana, [3rd]Shubham Kumar Jha, [4th]Saurabh Kumar, [5th]Hariom Upadhyay
[1,2,3,4] B-Tech CSE, R.D Engineering College Ghaziabad, U.P., India
[5]Professor, Department of CSE R.D Engineering College Ghaziabad, U.P., India

Abstract: **Predicting stock prices is very difficult due to the unpredictable behaviour of financial markets. We introduce a web application that uses Long Short-Term Memory (LSTM) networks to forecast stock prices in real time. The system continuously retrains its models using historical Yahoo Finance data which let users set custom prediction horizons through recursive forecasting and displays the results with interactive and animated time-series charts built using Chart.js. The system includes a tool that explains the predictions by comparing them to recent average values. Our tests shows that the app accurately predicts stock prices and update its visuals quickly which makes it a very useful tool for investors.**

1. INTRODUCTION

Predicting stock prices is a difficult job for investors and researchers because the market is very unpredictable and full of complex patterns. Due to this unpredictable nature, traditional techniques like moving averages and autoregressive models many times struggles to capture complex patterns in stock prices. Long Short-Term Memory (LSTM) networks have proved to be a very useful tool for handling sequential data patterns. We have created a real time stock price prediction system that predicts prices and also provides visuals and explanations which makes it easy to use for both researchers and investors.

2. RELATED WORK

Early on researchers have experimented with a wide range of techniques to forecast the stock market. Earlier they used statistical techniques like ARIMA (Nayak, Pai, & Pai, 2016) but these often doesn't work well because financial data is very unpredictable and complex. Many recent studies have started using machine learning and deep learning techniques, mainly neural network architectures such as RNNs and LSTMs (Kolte et al., 2022; Nabipour et al., 2020). Many studies have compared these models to see how well they catch the market's complex behaviour (Oyewole et al., 2024; Akinrinola et al., 2024). Tools like Chart.js make it easier to understand the predictions by showing them in interactive charts (Xu, n.d.). In addition to LSTM networks there are also a variety of models that have been applied to stock market prediction each with its own strengths and mechanisms.

- ARIMA (AutoRegressive Integrated Moving Average): ARIMA is one of the classic methods that uses past values and trends to predict future prices. It works well when the data follows a relatively stable linear pattern but it starts to struggle when the market is very volatile.

- SVM (Support Vector Machines): Although SVMs are best known for classification tasks they can also be adapted for forecasting by finding a line (or hyperplane) that best fits the data. They work well in those cases where you need to separate data into different trends but they might not capture complex time dependencies as naturally.

- Random Forest and XGBoost: These are ensemble methods that builds many decision trees to make predictions. Random Forest averages the results from many trees to remove the errors while XGBoost builds trees one after the other in which each one corrects the mistakes of the previous one. These techniques are decent and can work with various types of data good enough but might not capture the sequential patterns in stock prices as effectively as models that are designed mainly for time series.

- Convolutional Neural Networks (CNNs): CNNs were designed for image recognition. But they have been adapted to predict stock

prices by treating stock charts like images. They excel in identifying patterns in visual data which can be very useful where technical analysis is required.

- Transformer-Based Models: Transformer models use a technique named attention which let them look at all the parts of the input at once. This help them to capture long-term patterns in the data. It also allows them to present a new way of understanding and looking at market trends.

3. METHODOLOGY

The methodologies employed in our system are presented below. Figure 1 gives an overview of the entire pipeline showcasing how each component works together.

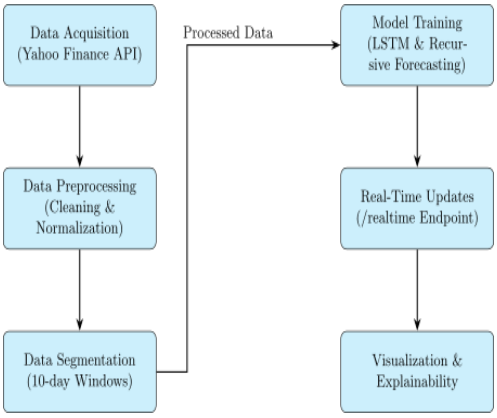

Figure 1: Two-Column System Workflow Diagram

3.1 Data Acquisition and Preprocessing

Data Collection: Historical stock data is collected using the yfinance API which includes features such as closing prices, trading volumes, and common technical indicators. Similar to methods described by Nayak et al. (2016) our system makes sure that a rich dataset is built over an extended period.

Data Normalization and Segmentation: We process the raw data using a MinMaxScaler that scales all features between 0 and 1. This helps the LSTM network to learn from the inputs more efficiently. And we break the processes data into overlapping chunks like 10-day sequences which we use as training samples. This step is necessary for capturing short-term trends as mentioned in several deep learning studies (Kolte et al., 2022).

3.2 Model Training and Forecasting

On-Demand LSTM Model Training: The machine learning module which we implemented using TensorFlow/Keras builds a LSTM model on demand. Each time the user requests predictions for a particular stock ticker the system checks for an existing model if it is not present it starts training a new model using the preprocessed data. Our design allows the model to adapt quickly to new information which is inspired from a technique supported by recent research on dynamic model training in financial forecasting (Nabipour et al., 2020).

Recursive Forecasting: To allow the predictions to be made for much further in the future the system uses recursive forecasting. This method generates predictions by using the previous prediction as part of the input for the next predictions. This approach increases the usefulness of the predictions for investors whether they are making short term or long term decisions.

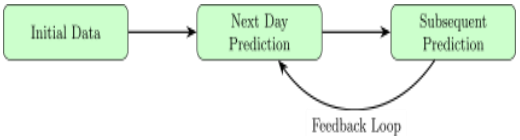

Figure 2: Recursive Forecasting Process

3.2.1 LSTM Model Formulation

The LSTM cell is designed to capture long-term dependencies in time-series data. Its internal operations are defined by the following equations:

Forget Gate:

$$f_t = \sigma(Wf \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate:

$$i_t = \sigma(Wi \cdot [h_{t-1}, x_t] + bi)$$

Candidate Memory Cell:

$$C_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cell State Update:

$$Ct = f_t \cdot C_{t-1} + i_t \cdot C_t$$

Output Gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Hidden State:

$$h_t = o_t \cdot \tanh(C_t)$$

Here, σ denotes the sigmoid function, tanh is the hyperbolic tangent function, x_t represents the input at time t, h_{t-1} is the hidden state from the previous time step, and W and b denote weight matrices and bias vectors, respectively.

3.2.3 Performance Evaluation Metrics

The performance of the predictive model is evaluated using several standard metrics:

Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\text{MSE}}$$

Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

In these equations, yi represents the actual stock price, $y\ i$ the predicted price, and n the number of predictions.

3.3 Real-Time Data Integration and Visualization

Real-Time Updates: The Flask backend includes a /realtime endpoint fetches the latest intraday data every 30 seconds and changes the timestamps to Indian Standard Time (IST). This ensures that the predictions are always made with the latest market conditions.

Interactive Visualization: The frontend built with HTML, CSS and JavaScript uses Chart.js to display animated and interactive time-series charts. These visuals display historical and forecasted data and also updates in real time. This approach is inspired from the methods recommended in recent literature for enhancing interpretability and user experience (Xu, n.d.).

3.4 Explainability and Interpretability

Explainability Module: Complex models are hard to understand so our system includes an explainability module that helps simplifying the predictions. This module calculates the average stock price over a recent period (e.g. past 10 days) and then compares that average price with the predicted price. This percentage difference provides a clear market sentiment, a positive value means a bullish trend and a negative value indicates a bearish trend.

This method offers an initial level of interpretability and supports the need for more transparent deep learning models as mentioned in recent research (Oyewole et al., 2024; Akinrinola et al., 2024). In future, we might add more advanced techniques like SHAP or LIME to show even more details. . The percentage difference is computed as follows:

$$Percentage\ Difference = \frac{Forecasted\ Price - 10\ day\ Average}{10\ day\ Average} \times 100$$

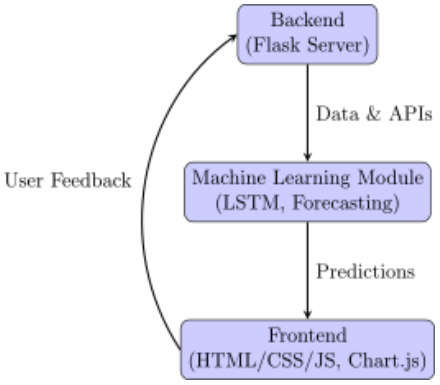

Figure 4: System Architecture Diagram

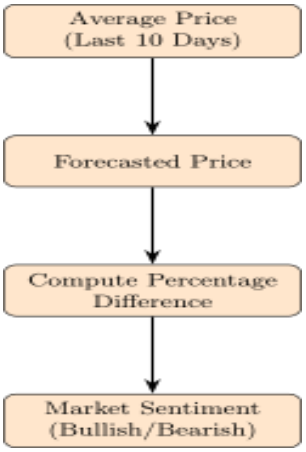

Figure 3: Explainability Module Diagram

3.5 System Architecture and Integration

Backend Implementation: The system works on a Flask web server which handles and manages HTTP requests, model training and data collection and deliver realtime updates to the frontend.

Modular Design: The web application is structured into three interconnected components. The backend

manages data and API endpoints. The machine learning module trains LSTM models and generates a prediction and the frontend provides visualizations and user interaction. This design makes the system scalable and easy to maintain.

4 RESULTS AND DISCUSSION

Our system performs well in terms of responsiveness and user experience. The web application retrieves the latest stock data every 30 seconds making sure that the displayed information is always up to date. The system processes various stock related data that includes closing prices, trading volume and technical indicators which allows users to analyze trends effectively. The frontend built with Chart.js present this data through interactive animated time-series charts that makes it easier to understand stock movements.

The explainability module enhances usability by comparing forecasted prices with the recent 10-day average offering a simple way to understand market trends. This feature allows the user to know whether a stock may be experiencing an upward or downward trend which makes it useful for both casual observers and more experienced investors.

Challenges

- Market Volatility: Stock prices can be highly unpredictable making it very difficult for any model to make accurate predictions every time.

- Data Variability: Stock prices are influenced by multiple factors such as global events, economic policies and company performance which can cause unexpected fluctuations.

- Computational Efficiency: Training LSTM models on demand requires significant processing power and optimizing performance for real-time forecasting still remains an ongoing challenge.

- Scalability: As more users interact with the system, ensuring smooth performance and quick response times will require additional optimizations.

Despite the given challenges, the system demonstrates its potential as a very useful tool for both inexperienced and experienced investors by providing real- time stock analysis with a user-friendly interface.

5 CONCLUSION AND FUTURE WORK

This paper shows the potential of LSTM networks for real-time stock price forecasting implemented within a web application framework. The system achieves high predictive accuracy and also enhances user understanding through interactive graphs and an explainability module. Future research will focus on implementing additional technical indicators, asynchronous model training to reduce latency and advanced interpretability techniques such as SHAP or LIME.

REFERENCES

[1] Akinrinola, O., Addy, W. A., Ajayi- Nifise, A. O., Odeyemi, O., & Falaiye, T. (2024). Predicting stock market movements using neural networks: A review and application study. *GSC Advanced Research and Reviews, 18*(2), 297–311. https://doi.org/10.30574/gscarr.2024.1 8.2.0072

[2] Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. C. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications, 9*, 100320. https://doi.org/10.1016/j.mlwa.2022.10 0320

[3] Kolte, G., Kini, V., Nair, H., & S. Babu, K. (2022). Stock market prediction using deep learning. *International Journal for Research in Applied Science & Engineering Technology, 10*(IV). https://doi.org/10.22214/ijraset.2022.4 1159

[4] Nayak, A., Pai, M. M., & Pai, R. M. (2016). Prediction models for Indian stock market. *Procedia Computer Science, 89*, 441–449. https://doi.org/10.1016/j.procs.2016.06.096

[5] Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E., & Shahab, S. (2020). Deep learning for stock market prediction. *Entropy, 22*, 840. https://doi.org/10.3390/e22080840

[6] Paliwal, S., & Sharma, S. (2022). Stock prediction using neural networks and evolution algorithm. *International Journal for Research in Applied Science & Engineering Technology, 10*(IV). https://doi.org/10.22214/ijraset.2022.4 1331

[7] Oyewole, A. T., Adeoye, O. B., Addy, W. A.,

Okoye, C. C., Ofodile, O. C., & Ugochukwu, C. E. (2024). Predicting stock market movements using neural networks: A review and application study. *Computer Science & IT Research Journal*, *5*(3), 651–670. https://doi.org/10.51594/csitrj.v5i3.912

[8] Sen, J., Waghela, H., & Rakshit, S. (n.d.). Exploring sectoral profitability in the Indian stock market using deep learning. Praxis Business School, Kolkata, India.

[9] Xu, S. Y. (n.d.). Stock price forecasting using information from Yahoo Finance and Google Trend. [Manuscript, UC Berkeley].

[10] Bhanja, S., & Das, A. (2024). A Black Swan event-based hybrid model for Indian stock markets' trends prediction. *Innovations in Systems and Software Engineering*, *20*, 121–135. https://doi.org/10.1007/s11334-021- 00428-0