Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

GrowwUp: A Virtual Stock Market Simulator with LSTM-Based Predictive Analytics for Indian Markets

Hitesh Pawar
Information Technology Department
Pune Institute of Computer Technology

Om Papdiwal
Information Technology Department
Pune Institute of Computer Technology

Devansh Mishra
Information Technology Department
Pune Institute of Computer Technology

Rohit Patil
Information Technology Department
Pune Institute of Computer Technology

Prof. Naman V. Buradkar
Information Technology Department
Pune Institute of Computer Technology

***Abstract*—It can be challenging to learn how stock market works especially to the beginners as it is a very dynamic market and the risks associated with it are financial. A lot of potential traders find it hard to acquire some hands-on experience without fear of losing money. This paper proposes a solution to this problem in the form of GrowwUp, a virtual stock market simulator that will offer a safe and interactive way of learning based on real-time data of the Indian stock market.**

The system enables its users to simulate the purchase and sale of stocks, manage their portfolios and analyze their profit and loss with no financial implications. Along with simulation, the platform incorporates a machine learning model that uses Long Short-Term Memory (LSTM) network to analyze past stock data and provide simple trading information. The model enables the user to know how the market trends change with time by inputting price movements, trading volume, and technical indicators.

Other recent system design practices included in the platform are WebSocket-based real-time updates and caching to facilitate a seamless performance and scalability. All in all, GrowwUp will fill the gap between theory and practice, making the stock market education more available, involving, and risk-free to newcomers.

***Index Terms*—Stock Market Simulation, LSTM, NSE, Time Series Prediction, Algorithmic Trading, Financial Education**

I. Introduction

The increasing availability of financial markets has motivated many people, in particular students and young career workers to consider stock trading as a viable skill and investment [30].With the emergence of digital channels and mobile applications, breaking into the market has never been easier. Yet, it is not necessarily an easy access that is followed by an easy understanding. To the majority of novices, the stock market remains to be a confusing, unpredictable and hard to read environment, especially in the context of analyzing the price trends and making informed conclusions [6].

The absence of a practical learning environment is one of the biggest challenges that new learners struggle with. Although theoretical material is readily available on the internet, it usually lacks the practical aspect of the real world experience of how trading is conducted [21].Order execution, portfolio management and timing in the market are some of the concepts that can only be learnt with practice and not by reading. Simultaneously, a direct application in actual markets is financially risky and will deter experimentation and restrict the chances of learning.

The other factor of significance is the challenge of interpretation of market data. Many technical indicators and charts are usually presented to the beginners, yet it takes time and experience to know how to fully utilize these technical indicators and charts [5]. Unless appropriately instructed, users can struggle to relate these indicators to a real market action. This puts emphasis on the fact that there is a need to have systems that are able to not only avail access to the data but also help the user to make sense of the data in an easy and user friendly way.

To address these issues, the GrowwUp platform is created as a stock market simulator, which is based on the concept of experiential learning. The system is aimed at simulating the true trading environment but eliminating the aspect of financial risk [10], [11]. Individuals are able to communicate with live markets, make simulated trades, and watch their decisions affect their portfolio over time. It will be an active method that will enable users to develop confidence and familiarity with market operations slowly.

Besides simulation, the platform also presents a simple predictive analysis level through machine learning [2].The system can give insights by analyzing historical data patterns to enable users to realize how the past trends can affect the future movements. It is not meant to substitute decision-making, but rather supplement it, helping users to consider their trading strategies in a more critical way.

In general, the work is aimed at establishing a balanced

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

learning space that would allow users to explore, experiment, and enhance their knowledge of stock trading. The system seeks to enhance the learning process by incorporating real time interaction and guided insights so as to make it more meaningful and engaging to the beginners.

II. LITERATURE SURVEY

The issue of the stock market prediction has attracted a lot of research because of its practical significance and complexity involved in it. Financial markets are very dynamic and affected by various issues including economic indicators, investor sentiment, and world events [30].Consequently, it is difficult to accurately predict the price of stocks. Researchers have over time investigated different methods as far as traditional statistical models to more advanced machine learning and deep learning methods [21], [24].

Initially, statistical techniques like the AutoRegressive Integrated Moving Average (ARIMA) model were used as the main basis of stock prediction [4]. These models presuppose linear relations in data and are good on regular time-series data. Non-linear data, however, is very volatile and is prone to stock market and such approaches cannot be effectively implemented in practice [29]. This drawback prompted researchers to investigate machine learning methods that have the ability to learn complex trends in financial data [20], [21].

As artificial intelligence has developed, machine learning algorithms like Support Vector Machines (SVM), Random Forest, and Artificial Neural Networks (ANN) have been proposed to predict the stock market [20], [21]. These models were shown to be more effective than the traditional methods since they learned the patterns of the past data. They are however not very effective in capturing temporal dependencies which are paramount in time- series forecasting tasks [29].

In order to address these shortcomings, Recurrent Neural Networks (RNNs) have been introduced which are specifically trained on sequential data [8]. Nevertheless, vanishing gradient is a problem of standard RNNs and long-term dependencies are hard to learn [16].To overcome this problem, Long Short-Term Memory (LSTM) networks were proposed that have memory cells and gating mechanisms [3], [18]. The LSTM models have been incredibly successful in predicting financial time-series forecasts because they can store the information that is relevant in predicting forecasts in long sequences [2], [7].

In the recent past, the accuracy of prediction has been further improved by the use of technical indicators like Moving Averages, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD) as input features to improve accuracy [5], [22]. These signals give further background details on market momentum and trends enhancing learning ability of deep learning models [24].

Other than prediction models, there are also virtual trading platforms that are made to assist users to acquire hands-on experience. Simulators like Investopedia Simulator and Neostock allow users to simulate the trading of stocks [10], [11]. Although these systems are practical, they are not integrated with smart prediction models and are not frequently offered to give meaningful analytical results.

Although both prediction models and simulation platforms have been advanced considerably, there is an apparent lack of systems that integrate real-time trading simulation and machine learning-driven insights [6], [27]. The majority of the studies are either predictive modeling or simulation studies. This drawback will drive the proposed system, which will incorporate the two to make the learning process more effective.

TABLE I
LITERATURE SURVEY ON STOCK MARKET PREDICTION AND SIMULATION SYSTEMS

Author(s)	Methodology	Result
G. Box and G. Jenkins (1976) [1]	Proposed ARIMA model for time-series forecasting based on linear statistical assumptions.	Effective for stable datasets but fails to capture non-linear and highly volatile stock market behavior.
T. Fischer and C. Krauss (2018) [2]	Applied LSTM networks for financial time-series prediction using historical stock data.	Demonstrated superior performance compared to traditional machine learning models such as SVM and Random Forest.
S. Hochreiter and J. Schmidhuber (1997) [3]	Introduced Long Short-Term Memory (LSTM) architecture to overcome vanishing gradient problem in RNNs.	Enabled learning of long-term dependencies, making it highly suitable for sequential financial data.
J. Brownlee (2018) [4]	Explored deep learning techniques for time-series forecasting including RNN and LSTM models.	Highlighted the effectiveness of deep learning in capturing non-linear patterns in stock market data.
Y. Zhang et al. (2017) [5]	Combined technical indicators with deep learning models for stock prediction.	Improved prediction accuracy by incorporating domain-specific financial features.
O. Sezer et al. (2020) [6]	Surveyed deep learning approaches for financial time-series forecasting.	Concluded that hybrid models using multiple features provide better real-world performance.
Investopedia Simulator [7]	Developed a virtual trading platform for educational purposes.	Provides risk-free practice but lacks predictive analytics and intelligent insights.
Neostock Platform [8]	Paper trading system with portfolio tracking features.	Limited analytical capabilities and no integration of machine learning models.

III. USE CASE DIAGRAM AND SYSTEM ARCHITECTURE

A. System Overview

The proposed system, GrowwUp, is designed as a virtual stock market simulator that combines real-time data processing, user interaction, and predictive analytics within a unified

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

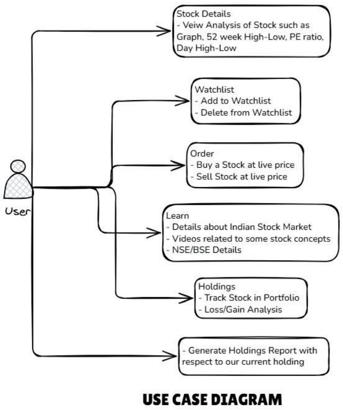

Fig. 1. Use Case Diagram

architecture. The system is structured to provide a smooth and responsive user experience while maintaining scalability and efficiency.

Unlike traditional trading platforms that focus only on execution, this system emphasizes learning and analysis. It enables users to interact with live market data, simulate trading actions, and observe portfolio behavior without financial risk. The architecture is modular in nature, allowing different components such as data fetching, user management, and analytics to operate independently while remaining interconnected.

B. Functional Modules Based on Use Case

The use case diagram shows the communication between the user and the various system functionalities. Every functionality is made to mimic the real-life trading characteristics and the interface is made easy and user-friendly.

1) Stock Details Module: This module enables users to access more information regarding specific stocks. It also contains graphical analysis and daily price movement and indicators like 52-week high and low values along with price to earnings ratios. The idea is to make users aware of the dynamics of stock price movements throughout time and how various measures can be used to make trading decisions.

2) Watchlist Management: Watchlist feature allows the user to monitor the stock(s) of their choice without trading. The users are able to add or remove stocks depending on their interest and hence they can follow the market trends before making a decision. This feature assists novices to see trends and think analytically without engaging in trade right away.

3) Order Execution Module: The system offers an imaginary trading platform in which a user is allowed to make and take buy and sell orders at actual market prices. The process is similar to the real trading systems though there are no real transactions actually performed. This will assist the users to have an intuitive grasp of order execution, price alterations, and market timing.

4) Learning Module: The site has an exclusive learning section to offer guidance to novices, where they get details on the concepts of the stock market, the activities of NSE/BSE, and simple trading strategies. This module also provides the linkage between theory and practice by incorporating the educational content into the system.

5) Portfolio and Holdings Module: This module enables the user to monitor their simulations of investments. It gives information on the profit and loss, the average price of buying, and the performance of the portfolio in general. Through monitoring these metrics, users are able to test their choices and enhance their strategies as time goes by.

6) Report Generation: The system also produces user holding summaries providing a clear picture of current positions and performance. This option is especially helpful to examine the trend and comprehend the results of trading choices in the long term.

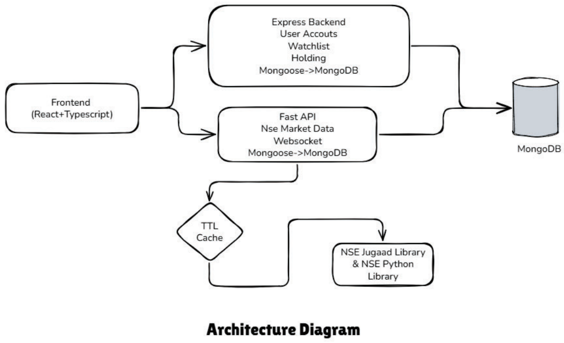

Fig. 2. System Architecture Diagram

C. Architecture Description

The system architecture is broken down into three key layers which include frontend, backend services and data management. The role of each layer is to make sure that the process of communication and efficient processing is carried out.

1) Frontend Layer: React and TypeScript are used to build the frontend and offer an interactive interface that is easy to use. It manages user interactions, provides real-time stock information, and presents portfolio data in the form of charts and dashboards. The frontend interacts with the backend services via APIs and Web socket connections.

2) Backend Services: The backend is divided into two major parts:

Express Backend: The component handles operations of users, including authentication, watchlist, and portfolio management. It uses direct interaction with the database to store as well as retrieve user information.

FastAPI Service: This element is in charge of the stock market data retrieval and processing. It connects itself to external NSE data sources and updates in real-time via WebSockets. FastAPI is selected because of its asynchronous features, which make it possible to effectively work with a number of requests.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

3) Data Layer: The major database is MongoDB where user information, portfolio details and watchlist data are stored. The flexibility of the NoSQL database usage is in the possibility to deal with dynamic data structure.

4) Caching Mechanism: A Time-To-Live (TTL) cache is put in place to enhance performance. Temporarily stored is stock data that is being accessed often, instead of calling external APIs to fetch the required data. This will minimize latency, external requests, and improve scalability of the system.

5) External Data Integration: The system is based on NSE information library to retrieve real-time stock data. These libraries serve as a middle ground between the application and the stock exchange data thus allowing the retrieval of the information correctly and up-to-date.

D. System Workflow

The general sequence of work of the system starts when a user communicates with the frontend interface. The requests are dispatched to the backend where they are handled depending on the nature of operation. In the case of stock data, the FastAPI service is used to retrieve and stream updates whereas the Express backend is used to store and retrieve data in the case of user related actions.

The caching layer will make repeated requests efficient without the needless delays. WebSockets are used to push real-time updates to the frontend, and the users are always up-to-date about the latest market information.

None of the architectural designs are complete without their major attributes.

E. Key Characteristics of the Architecture

- **Modular Design:** Each component operates independently, making the system easier to maintain and extend.
- **Real-Time Communication:** WebSockets enable continuous data streaming without repeated polling.
- **Scalability:** The use of asynchronous APIs and caching allows the system to handle multiple users efficiently.
- **Low Latency:** TTL caching significantly reduces response time for frequently requested data.
- **User-Centric Design:** The system focuses on providing a simple and intuitive experience for beginners.

IV. Implementation Details

A. System Development Environment

GrowwUp platform is implemented based on the mix of the modern web technologies and machine learning frameworks to make sure it is both effective and scaled. The system is also designed to be efficient in processing real-time data as well as being user friendly and interactive.

The frontend is implemented in React and TypeScript which allows building dynamic user interfaces and responsive components. In the case of backend services, the hybrid approach is implemented, and user-related operations are performed with the combination of Node.js and Express, whereas the real-time stock data and machine learning integration are implemented with FastAPI. Python is mostly applied in data processing and model development because it has a wide range of support of scientific computing and machine learning libraries.

1) Software Requirements:

- Programming Languages: Python, JavaScript (TypeScript)
- Frontend Framework: React.js
- Backend Frameworks: FastAPI, Node.js with Express
- Database: MongoDB
- Machine Learning Libraries: TensorFlow / Keras, NumPy, Pandas
- Real-Time Communication: WebSockets
- Visualization Tools: Chart.js / Recharts
- Development Tools: VS Code, Postman

2) Hardware Requirements:

- Processor: Intel i5/i7 or equivalent
- RAM: Minimum 8 GB (16 GB recommended)
- Storage: At least 10 GB free space
- GPU: Optional (for faster model training)

B. Data Acquisition and Processing

The system relies on real-time stock data obtained from NSE through open-source libraries such as nsepython and NSE Jugaad. These libraries provide access to stock prices, trading volume, and historical data without requiring paid APIs.

1) Data Collection: Historical stock data is collected in the form of:

- Open, High, Low, Close (OHLC) prices
- Trading volume
- Historical price trends

This data serves as the foundation for both visualization and machine learning tasks.

2) Data Preprocessing: Before feeding the data into the model, several preprocessing steps are applied:

- Handling missing or inconsistent values
- Normalization of numerical features
- Creation of sliding windows for time-series modeling
- Feature scaling for stable training

These steps ensure that the data is consistent and suitable for training the predictive model.

C. LSTM Model Implementation

The predictive component of the system is implemented using a Long Short-Term Memory (LSTM) network, which is well-suited for sequential data such as stock prices.

1) Feature Selection: The model uses multiple input features, including:

- OHLC prices
- Trading volume
- Moving Averages
- Relative Strength Index (RSI)
- Moving Average Convergence Divergence (MACD)

These features help the model capture both price trends and market momentum.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

2) Model Architecture: The LSTM network consists of multiple layers that process sequential data and learn temporal dependencies.

$$h_t = f(W \cdot x_t + U \cdot h_{t-1} + b)$$

$$y_t = W_y \cdot h_t + b_y$$

The gating mechanism within LSTM allows the model to retain relevant information and discard noise, making it effective for financial prediction.

3) Training Process: The model is trained on historical data using backpropagation through time (BPTT). The training process includes:

- Splitting data into training and testing sets
- Using loss functions such as Mean Squared Error (MSE)
- Optimizing weights using Adam optimizer
- Evaluating performance on unseen data

4) Prediction Output: The trained model generates predictions for future stock prices and basic trading signals such as Buy, Sell, or Hold. These predictions are used to assist users in understanding market trends.

D. Real-Time Data Handling using WebSockets

Initially, the system relied on periodic API calls to fetch stock updates, which resulted in increased latency and server load. To address this issue, WebSockets were implemented for real-time communication.

1) Working Mechanism:

- The backend establishes a persistent connection with the frontend
- Stock price updates are pushed continuously without repeated requests
- Multiple users can receive updates simultaneously

This approach significantly improves responsiveness and reduces unnecessary network overhead.

E. Caching Mechanism (TTL Cache)

To further optimize performance, a Time-To-Live (TTL) cache is implemented.

1) Functionality:

- Frequently requested stock data is stored temporarily
- Cached data is reused within a defined time window
- Reduces repeated API calls to external data sources

This improves system efficiency while maintaining near real-time accuracy.

F. Portfolio Management Implementation

The portfolio module tracks user investments and calculates performance metrics.

1) Core Features:

- Tracking buy and sell transactions
- Calculating profit and loss
- Computing average purchase price
- Generating portfolio summaries

All data is stored in MongoDB and updated dynamically as users perform actions.

G. Integrated Execution Flow

The overall execution process of the system follows a structured workflow:

1) Fetch real-time stock data from NSE sources
2) Process and cache data for efficient access
3) Stream updates to frontend using WebSockets
4) Allow users to perform trading actions
5) Store user data and portfolio updates in database
6) Generate predictions using LSTM model
7) Display insights and analytics on dashboard

H. Implementation Challenges

- Managing real-time data updates without performance degradation
- Handling large volumes of concurrent user requests
- Ensuring accuracy of predictive model in volatile market conditions
- Balancing system responsiveness with data freshness

V. Acknowledgment

It is with much gratitude that the authors would like to acknowledge our project guide and faculty members who constantly guided us, offered feedback and support at all points in the development of the GrowwUp system. Their knowledge and support contributed greatly to the choice of the direction of this project and the quality of the entire work.

We also wish to acknowledge the Department of Information Technology who has been able to offer the infrastructure and resources that were necessary to implement and test this work successfully. The smooth running of the project was enhanced by the academic environment and technical support.

We would like to thank our peers and colleagues who supported us, discussed our works and motivated on various development stages. Their proposals aided in the polishing of ideas and solving many problems.

Lastly, we owe a great debt of gratitude to our families because they supported and encouraged us all the way through the process of this project completion.

VI. Conclusion

GrowwUp system proves to be an efficient way of making stock market learning more realistic and reachable. The platform provides users with hands-on experience with simulated trading environment and real-time data to experience this type of environment without risking their funds. The system is enhanced further by the integration of a time-series prediction model that gives the system meaningful insights into the behavior of the market.

The design in general is aimed at simplicity, responsiveness, and usability, so that the beginners will be able to use the system conveniently, developing their knowledge step by step. The deployment also underscores how the modern technologies can be utilized in a combination to develop scalable and effective financial applications.

This article demonstrates that a properly developed platform of simulation can be of significance in making financial

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

awareness and assisting the users in building confidence prior to entering the actual markets.

VII. Future Work

More complex deep learning techniques may also be employed to refine the existing system and ensure a higher accuracy of prediction and adaptation to the changing market conditions. The extension of the system to international stock markets would give the users more exposure and learning. Also, it can be helpful to include intelligent recommendation capabilities that will help users make more informed trading choices based on information-driven insights. Additional modifications can also be oriented at the improvement of real-time performance and the establishment of individual learning modules.

References

[1] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, 1997.
[2] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” European Journal of Operational Research, 2018.
[3] J. Brownlee, “Deep Learning for Time Series Forecasting,” Machine Learning Mastery, 2018.
[4] G. Box and G. Jenkins, “Time Series Analysis: Forecasting and Control,” 1976.
[5] Y. Zhang, Y. Wang, and X. Wang, “Stock Market Prediction Using Deep Learning,” 2017.
[6] O. Sezer, M. Gudelek, and A. Ozbayoglu, “Financial Time Series Forecasting with Deep Learning: A Survey,” Applied Soft Computing, 2020.
[7] M. Nelson, A. Pereira, and R. de Oliveira, “Stock Market’s Price Movement Prediction with LSTM Neural Networks,” 2017.
[8] A. Graves, “Supervised Sequence Labelling with Recurrent Neural Networks,” Springer, 2012.
[9] NSE India, “Official Market Data and Reports,” National Stock Exchange of India.
[10] Investopedia Simulator, Online Trading Platform.
[11] Neostock, Virtual Trading Application.
[12] TensorFlow, “An End-to-End Open Source Machine Learning Platform,” Google, 2015.
[13] MongoDB Inc., “MongoDB Documentation,” 2023.
[14] FastAPI, “High Performance Web Framework for APIs,” 2023.
[15] Node.js Foundation, “Node.js Documentation,” 2023.
[16] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” IEEE Transactions on Neural Networks, 1994.
[17] K. Cho et al., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” 2014.
[18] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” Neural Computation, 2000.
[19] J. Yao and C. L. Tan, “A case study on using neural networks to perform technical forecasting of forex,” Neurocomputing, 2000.
[20] K. Kim, “Financial time series forecasting using support vector machines,” Neurocomputing, 2003.
[21] H. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock market index using fusion of machine learning techniques,” Expert Systems with Applications, 2015.
[22] S. Selvin et al., “Stock price prediction using LSTM, RNN and CNN-sliding window model,” International Conference on Advances in Computing, 2017.
[23] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for event-driven stock prediction,” IJCAI, 2015.
[24] W. Bao, J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and LSTM,” PLoS ONE, 2017.
[25] Z. Hu, Y. Zhao, and M. Khushi, “A survey of forex and stock price prediction using deep learning,” Applied System Innovation, 2021.
[26] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques,” Expert Systems with Applications, 2015.
[27] A. Hiransha, E. Gopalakrishnan, V. Menon, and K. Soman, “NSE stock market prediction using deep-learning models,” Procedia Computer Science, 2018.
[28] M. Krauss, S. Do, and N. Huck, “Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the SP 500,” European Journal of Operational Research, 2017.
[29] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A comparison of ARIMA and LSTM in forecasting time series,” IEEE Big Data Conference, 2018.
[30] R. J. Hyndman and G. Athanasopoulos, “Forecasting: Principles and Practice,” 2018.