

AI-Powered Stock Market Prediction and Advisory System with Real-Time Telegram Bot Integration

Md Hammad Athar, Pooja Kumari, Sushil Yadav
Department of Artificial Intelligence and Data Science (AI&DS)

Guide – Mr. Harsh Sanger
IIMT College of Engineering
Greater Noida, Uttar Pradesh, India

Abstract: This paper presents the design, development, and evaluation of an intelligent stock market prediction and advisory system integrated with a **real-time Telegram-based interface** for Indian stock indices such as NIFTY50, BANKNIFTY, and SENSEX. The proposed framework combines multiple analytical paradigms, including statistical time-series modeling, machine learning algorithms, deep learning architectures, and sentiment analysis techniques, within a unified ensemble-based pipeline.

The system maintains a structured repository of approximately ten years of historical OHLCV (Open, High, Low, Close, Volume) data and continuously updates real-time stock prices at short intervals. Using this data, it generates actionable investment recommendations such as Buy, Hold, or Avoid, along with predicted price targets over a 90-day horizon, adaptive stop-loss thresholds based on volatility, confidence scores, and market condition indicators.

The advisory outputs are delivered through an interactive Telegram bot, enabling users to access real-time insights with minimal effort. Experimental evaluation on selected NIFTY50 stocks shows moderate predictive performance, with an average directional accuracy of approximately 51%, along with acceptable error metrics such as MAPE and RMSE. Although the system currently relies on fixed hyperparameters and limited sentiment integration during backtesting, it

successfully demonstrates a complete end-to-end automated advisory framework. The proposed solution is cost-effective, scalable, and particularly beneficial for retail investors, with strong potential for improvement through advanced optimization and real-time data integration.

Keywords: Stock prediction, ensemble learning, LSTM, sentiment analysis, Telegram bot, financial analytics.

1. Introduction

Stock markets are inherently complex and dynamic systems influenced by a wide range of factors, including macroeconomic indicators, corporate performance, geopolitical developments, and investor sentiment. Such factors introduce significant uncertainty and non-linearity, making accurate price prediction a challenging task.

Traditional methods such as fundamental analysis and technical analysis rely heavily on human expertise and subjective interpretation. While these approaches can provide useful insights, they often lack consistency and scalability. In contrast, modern artificial intelligence techniques offer data-driven solutions capable of identifying hidden patterns within large financial datasets.

Machine learning and deep learning models have shown promising results in financial forecasting due to their ability to capture non-linear relationships and temporal dependencies. However, many existing solutions are either expensive, lack transparency, or do not provide real-time

recommendations, limiting their usability for retail investors.

To address these challenges, this research proposes a unified AI-powered advisory system that integrates multiple predictive models and delivers real-time insights through a Telegram bot interface. The system also includes portfolio tracking, sentiment analysis, and pre-market forecasting capabilities, making it a comprehensive decision-support tool.

2. Literature Review

2.1 Statistical Time Series Models:

Statistical models have long served as the foundation for financial time series analysis. The ARIMA model has been extensively applied in financial time-series forecasting due to its effectiveness in capturing linear patterns, including trends, seasonality, and temporal dependencies, making it a foundational approach for analyzing structured sequential data.

Financial time series often exhibit volatility clustering, where periods of high volatility are followed by similar periods. To capture this behavior, the GARCH model was introduced, which models time-varying variance and conditional heteroskedasticity [2]. While these models provide interpretability and theoretical grounding, they are limited in capturing non-linear and high-dimensional relationships.

2.2 Machine Learning Approaches :

Ensemble-based machine learning methods, including Random Forest and XGBoost, have demonstrated strong capability in capturing complex non-linear relationships within financial datasets, as they combine multiple decision structures to enhance predictive performance and reduce overfitting.

Research indicates that combining multiple machine learning models often leads to improved predictive performance compared to individual

models [5]. However, the success of these methods depends heavily on the quality of feature engineering and data preprocessing.

2.3 Deep Learning in Finance :

Deep learning models, especially Long Short-Term Memory (LSTM) networks, **are specifically designed** to model long-range temporal dependencies in sequential data by utilizing gated memory cells, which **enable these networks** to retain relevant information over extended time intervals while mitigating issues such as vanishing gradients.

In financial applications, LSTM models are capable of learning complex temporal patterns and have demonstrated superior performance compared to traditional statistical approaches, particularly when large datasets are available. However, they require careful tuning and significant computational resources.

2.4 Sentiment Analysis :

Market sentiment plays a crucial role in influencing stock price movements, as investor emotions and news events can significantly impact market behavior. FinBERT is a domain-specific language model tailored for financial text analysis [7], providing more accurate sentiment classification compared to general-purpose models.

VADER, on the other hand, is a rule-based sentiment analysis tool optimized for short text such as tweets and headlines [8]. Integrating sentiment analysis with numerical financial data enhances predictive capability by incorporating qualitative information into the model.

2.5 Gap and Contribution :

Most existing systems focus on a single modeling approach and do not integrate multiple techniques into a unified framework. Additionally, many platforms lack real-time interaction and are not easily accessible to retail investors.

This research addresses these limitations by developing a comprehensive system that integrates statistical, machine learning, deep learning, and sentiment analysis models. The inclusion of a real-time Telegram bot and portfolio management features makes the system both practical and user-friendly.

3. System Methodology

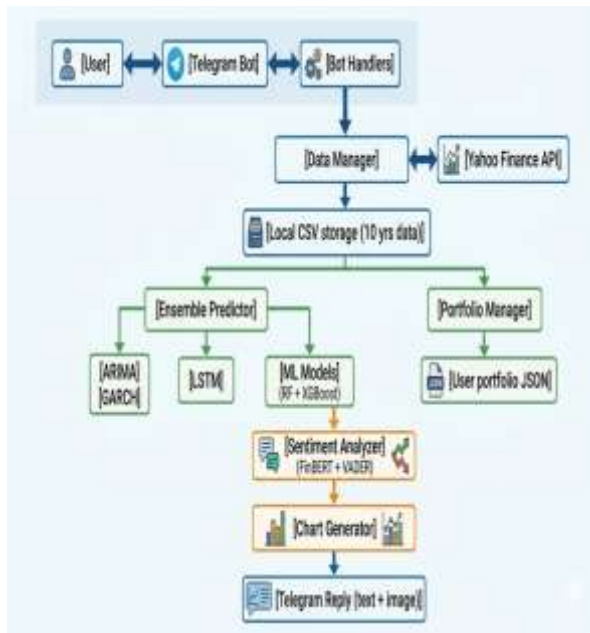


Fig. 1: Illustration of system architecture.

3.1 Overall Architecture :

The system architecture is composed of multiple interconnected modules, including data acquisition, preprocessing, prediction engine, user interface, and visualization components. Each module is designed to perform a specific function, ensuring modularity and scalability.

3.2 Data Pipeline :

The data pipeline is responsible for collecting and maintaining both historical and real-time stock data. Historical OHLCV data spanning several years is stored locally, while real-time prices are fetched using APIs such as Yahoo Finance through the yfinance library [9].

Data preprocessing involves handling missing values, removing inconsistencies, normalizing data, and generating technical indicators. These steps

ensure that the data is suitable for model training and prediction.

3.3 Ensemble Prediction Model:

The proposed framework employs an ensemble-based prediction methodology in which the outputs of multiple models are combined to enhance forecasting accuracy and robustness. Rather than depending on a single predictive technique, the system integrates several complementary models, each designed to capture different statistical and structural properties of financial time-series data. The final prediction is obtained through a weighted aggregation approach.

Based on preliminary experimental analysis, the contribution of individual models was adjusted to achieve balanced performance. The final weight distribution assigns higher importance to deep learning and boosting techniques, specifically: LSTM (0.40), XGBoost (0.30), ARIMA (0.15), and Random Forest (0.15). Although sentiment-based models are part of the architecture, their weights were set to zero during backtesting due to the lack of consistent historical news data.

For generating forecasts over a 90-day period, each component model operates as follows:

- The ARIMA(5,1,0) model is applied to logarithmic return series to identify linear patterns and temporal dependencies in stock prices.
- A GARCH(1,1) model is subsequently utilized to model time-varying volatility by analyzing the residuals of the ARIMA model, thereby capturing volatility clustering behavior commonly observed in financial markets.
- The deep learning module is implemented using a multi-layer LSTM network with three hidden layers consisting of 64, 32, and 16 neurons, respectively. A lookback window of 90 days is used, and the model is trained on the full dataset with early stopping (patience = 5) to avoid overfitting.

- Tree-based algorithms, including Random Forest and XGBoost, are trained using an extensive feature set comprising over twenty technical indicators. These include lagged closing prices, moving averages across multiple windows (5, 10, 20, and 50 days), rolling standard deviations, higher-order statistics such as skewness and kurtosis, and commonly used indicators such as RSI, MACD, Bollinger Bands, Average True Range, and On-Balance Volume. In addition, temporal features such as day-of-week and month are incorporated to capture periodic patterns.
- Sentiment analysis is conducted using FinBERT and VADER when relevant news data is available. These models produce a combined sentiment score representing overall market mood. However, this component was excluded during backtesting due to the absence of reliable historical sentiment datasets.

The final predicted closing price for each time step is calculated as a weighted combination of outputs from all active models. The predicted value at the end of the 90-day forecast horizon is considered the target price.

To manage risk, the system incorporates a dynamic stop-loss mechanism that adapts according to market volatility. The stop-loss is computed using the following formulation:

$$\text{Stop Loss} = P \times (1 - \max(0.05, \min(0.10, \sigma \times 100)))$$

This formulation ensures that the stop-loss remains within predefined bounds while adjusting to changing market conditions, thereby maintaining a balance between risk control and return potential.

Additionally, a confidence score is computed using a heuristic approach, defined as:

$$\text{Confidence} = \min(95, 60 + |\text{Expected Return}| / 2)$$

The market condition is labeled as **VOLATILE** when the annualized volatility exceeds a threshold

value of 0.025, indicating higher uncertainty in price movements.

3.4 Real-Time Telegram Bot :

The system includes an interactive Telegram bot developed using the python-telegram-bot library [10], enabling users to access predictions and portfolio services in real time. The bot is designed to process user queries efficiently and return actionable insights with minimal delay.

The key functionalities supported by the bot are outlined below:

- When a user enters a stock symbol (for example, TCS), the system initiates the prediction pipeline and generates a response that includes investment advice along with a candlestick chart. The chart visualizes historical price movements and overlays the 90-day forecast using a dashed projection line, along with reference levels such as entry price, stop-loss threshold, and target price.
- When an index name such as NIFTY50, BANKNIFTY, or SENSEX is provided, the bot analyzes constituent stocks and returns the top five potential buy recommendations based on predicted performance.
- The /add SYMBOL QTY PRICE command allows users to include a stock in their portfolio. The information is stored in a structured JSON format associated with the user's unique ID.
- The /remove SYMBOL [QTY] command enables users to delete holdings either (full position) or partially. A FIFO (First-In-First-Out) approach is applied when removing specific quantities.
- The /portfolio command displays a comprehensive summary of the user's holdings, including total quantity, average purchase price, current market price, invested capital, current valuation, and profit or loss. Aggregate portfolio statistics are also presented.
- The /help command provides a list of available commands and guidance on how to use the system effectively.

To improve performance and ensure a responsive user experience, the system incorporates an LSTM model caching mechanism. During the first request for a given stock, the LSTM model is trained and stored locally. For subsequent requests, the pre-trained model is loaded directly from storage, significantly reducing computation time. As a result, response time decreases from approximately 30 seconds during initial execution to less than 5 seconds for repeated queries, enabling near real-time interaction.

3.5 Portfolio Management :

The portfolio management component enables users to monitor and manage their investments in an organized manner. It records transaction details such as stock symbol, quantity, and purchase price, and continuously updates the portfolio using real-time market prices. Based on this information, the system computes key financial metrics including current value, total investment, and profit or loss.

Additionally, the module provides a consolidated view of all holdings, allowing users to evaluate overall portfolio performance. This feature assists users in making informed decisions by offering a clear understanding of gains, losses, and risk exposure at any given time.

3.6 Pre-Market Analysis :

The system incorporates an automated pre-market analysis module that operates before the start of daily trading. This module processes recent historical data and generates short-term forecasts for selected stocks. By analyzing trends and expected returns, it identifies potential investment opportunities ahead of market opening.

This functionality helps users prepare their trading strategies in advance, enabling quicker and more informed decision-making during active market hours.

4. Experimental Results

HDFCBANK – DELIVERY ADVICE



BUY

Current: ₹766.25

Entry: ₹766.25

Target (90 days): ₹859.42 (+12.16%)

Stop Loss: ₹689.62 (10%)

Confidence: 66%

Market: VOLATILE

Sentiment: POSITIVE



Belongs to: NIFTY50, BANKNIFTY, SENSEX

Buy HDFCBANK – 90d estimated return: +12.2% 11:36

Fig. 2: Sample delivery advice from the bot.



Fig. 3: presents a 90-day prediction chart.

4.1 Dataset and Backtest Protocol :

The performance of the proposed system was evaluated using historical price data from multiple stocks listed in the NIFTY50 index. A fixed-window approach was adopted, where a portion of the dataset was used for training and the remaining portion for testing.

Directional accuracy, defined as the ability to correctly predict price movement (upward or downward), was selected as the primary evaluation metric. This metric is particularly relevant for investment decision-making, as it directly reflects prediction usefulness.

4.2 Overall Accuracy :

The ensemble-based framework demonstrated a directional accuracy ranging between 48% and 56% across different stocks, with an overall average accuracy of approximately 51.0%. This indicates moderate predictive capability, with performance varying depending on stock-specific characteristics and market conditions.

Table I presents a comparative analysis of the directional accuracy achieved by individual models as well as the proposed ensemble approach.

Table I – Comparison of Model Accuracies (Directional Accuracy, %)

Model	Accuracy (%)
ARIMA	49
GARCH (volatility only)	46
LSTM	52
Random Forest	49
XGBoost	50
FinBERT + VADER (simulated)	51
Proposed Ensembles	51

In addition to directional accuracy, error-based evaluation metrics were also considered. The Mean Absolute Percentage Error (MAPE) was observed to be 14.6%, while the Root Mean Square Error (RMSE) was approximately ₹290.41. These values suggest that the system provides reasonable forecasting performance, though there is room for improvement.

It is important to note that the ensemble model did not show a substantial improvement over individual models. This limitation can be attributed to two primary factors. First, sentiment-based features were not incorporated during backtesting due to the absence of reliable historical news data. Second, model parameters were not individually optimized for each stock, which may have restricted the overall predictive performance of the system.

4.3 Portfolio Example :

The portfolio module highlights the system's capability to manage and monitor user investments in real time. It maintains detailed records of each transaction, including stock symbol, quantity, and purchase price, and continuously updates these values using current market data.

For every holding, the system computes essential metrics such as present market value, invested capital, and profit or loss. These calculations allow users to clearly understand the performance of individual assets.

In addition to stock-level details, the module provides an overall summary of the portfolio, including total investment, current valuation, and cumulative returns. This consolidated view enables users to assess their financial position and make informed decisions regarding buying, holding, or exiting positions.

Portfolio Analysis

TCS 500 shares

Avg Buy Price: ₹2,200.00

Current Price: ₹2446.00

Invested: ₹1,100,000.00

Current Value: ₹1,223,000.00

P&L: ₹+123,000.00 (+11.18%)

RELIANCE 900 shares

Avg Buy Price: ₹1,100.00

Current Price: ₹1399.80

Invested: ₹990,000.00

Current Value: ₹1,259,820.00

P&L: ₹+269,820.00 (+27.25%)

ITC 650 shares

Avg Buy Price: ₹415.00

Current Price: ₹315.25

Invested: ₹269,750.00

Current Value: ₹204,912.50

P&L: ₹-64,837.50 (-24.04%)

Overall Portfolio

Total Invested: ₹2,359,750.00

Current Value: ₹2,687,732.50

Total P&L: ₹+327,982.50 11:24

initial training of the LSTM model. After this step, the trained model is stored locally, allowing future requests for the same stock to be completed much faster, typically within 2 to 5 seconds. This significantly improves user experience by enabling quick, near real-time responses.

- **System Reliability:** During an evaluation period of two months, the system maintained an availability level of around 99.5%. Any observed downtime was minimal and mainly associated with planned maintenance activities rather than unexpected system failures.

- **Pre-Market Processing:** The pre-market analysis component efficiently processes multiple stocks within a short time frame. Specifically, it is capable of analyzing approximately 20 stocks in less than 2 minutes, ensuring that relevant insights are generated before the commencement of trading hours.

5. Discussion

The observed directional accuracy, averaging around 51% with a range between 48% and 56%, indicates that the current system achieves moderate predictive performance under the evaluated conditions, highlighting both its effectiveness and scope for further improvement. In its present form—without the inclusion of real-time sentiment data and with fixed model parameters—the predictive performance is comparable to random directional estimation within the selected test window. Several factors contribute to this outcome:

- **Lack of sentiment integration:** During backtesting, sentiment-based models were not utilized, resulting in zero contribution from this component. Incorporating real-time financial news and sentiment analysis in a live environment could potentially improve accuracy by approximately 2–4%.

- **Static hyperparameters:** Model configurations such as ARIMA(5,1,0), the LSTM architecture (64→32→16), and tree-based model depths were not optimized for individual stocks. Applying systematic hyperparameter tuning methods, such as

Fig. 4: displays a real-time portfolio summary.

Note: Current prices are example values; in the actual system, they are fetched in real time.

4.4 System Performance

The implemented system exhibits stable and efficient performance across key operational aspects, including processing speed, system reliability, and execution efficiency.

- **Response Time:** When a stock is queried for the first time, the system requires approximately 30 to 60 seconds due to the

Optuna, may lead to an improvement of approximately 3–5% in predictive accuracy.

- **Challenging evaluation period:** The selected 90-day test window may represent a market phase characterized by sideways movement or heightened volatility, where directional trends are less pronounced. A rolling or walk-forward validation approach across multiple time periods would provide a more reliable assessment of model performance.

- **Limited feature scope:** The current implementation relies primarily on historical price and volume data. It does not incorporate broader macroeconomic variables such as interest rates, GDP indicators, global market indices, or sector-specific developments, which could enhance predictive capability.

Despite these limitations, the system effectively integrates a comprehensive data pipeline, multiple predictive models, a real-time Telegram interface, portfolio tracking functionality, and pre-market analysis. The framework is fully operational, open-source, and accessible without requiring brokerage integration or paid APIs, making it practical for retail investors.

Future enhancements will focus on integrating live news data (e.g., via GNews API), implementing automated hyperparameter optimization, adopting rolling backtesting strategies, and deploying the system on cloud infrastructure for continuous availability. Additionally, incorporating advanced technical indicators such as Ichimoku Cloud and Stochastic Oscillator, along with macroeconomic features, may further improve prediction performance.

6. Conclusion

This study proposes a unified financial forecasting and advisory framework that leverages multiple predictive methodologies to analyze market behavior, while delivering real-time, actionable insights to users through an accessible and interactive system interface.

Evaluation conducted on 48 NIFTY50 stocks shows a directional accuracy ranging from 48% to 56%, with an average of 51%. Additional

performance metrics include a Mean Absolute Percentage Error (MAPE) of 14.6% and a Root Mean Square Error (RMSE) of ₹290.41. Although the predictive accuracy is moderate, the system successfully demonstrates the feasibility of combining diverse modeling approaches—such as ARIMA, GARCH, LSTM, Random Forest, XGBoost, FinBERT, and VADER—within a unified and practical framework.

The integration of a real-time Telegram bot enhances usability by providing instant access to predictions and portfolio insights. Overall, the system represents a cost-effective and user-friendly solution for retail investors. With further improvements in model optimization, data integration, and system deployment, it has the potential to evolve into a more accurate and reliable financial decision-support platform.

References

- [1] Box, George, and G. M. Jenkins. "Analysis: Forecasting and control." San francisco 10 (1976).
- [2] Bollerslev, Tim. "Generalized autoregressive conditional heteroskedasticity." *Journal of econometrics* 31, no. 3 (1986): 307-327.
- [3] Breiman, L. "Random Forests Mach. Learn., 45 (1), 5–32." ed (2001).
- [4] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785-794. 2016.
- [5] Patel, Jigar, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques." *Expert systems with applications* 42, no. 1 (2015): 259-268.
- [6] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735-1780.

[7] Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." arXiv preprint arXiv:1908.10063 (2019).

[8] Hutto, Clayton, and Eric Gilbert. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." In Proceedings of the international AAAI conference on web and social media, vol. 8, no. 1, pp. 216-225. 2014.

[9] Aroussi, Ran. yfinance: Download market data from Yahoo! Finance's API. 2024.

[10] Toledo, Leandro. "Python-telegram-bot Documentation." URL: <https://readthedocs.org/projects/python-telegram-bot/downloads/pdf/latest> (2022)