International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1(2) (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

www.ijerst.com
© 2026 IJERST. All Rights Reserved
PeerReviewed, Referred & Indexed Journal

Research Paper

Real Time Stock Market Analysis Using Agentic AI

Mrs. Potnuri Gayatri [1], CheboluSaikumar [2], Padagala Jhansi [3], Jeeru NikhileshReddy [4], Muvvala Devi Kavya [5]

Associate Professor [1], Student [2], Student [3], Student [3], Student [4]

[1,2,3] Department of Artificial Intelligence

Chaitanya Engineering College, Visakhapatnam, Andhra Pradesh, India

{gayatripotnuri119@gmail.com,saikumarchebolu5406@gmail.com, jhansipadagala@gmail.com,

nikhileshreddyj@gmail.com, srikavya58481@gmail.com}

Abstract

Modern financial markets have produced vast amounts of data, the complexity of whichisway beyond the ability of traditional statistical methods to handle. This paper is a stock-market analysis platform in real time, which combines both deep learningforecasting and an autonomous AI agent. The system will retrieve open-high-low-volume live data and historical Open-High-Low-Volume (OHLCV) data in theAlpha Vantage API and augments it with feature-engineering operations to generate four technical indicators: Relative Strength Index (RSI), Average True Range (ATR), Moving Average Convergence Divergence (MACD), and Money Flow Index (MFI). In this study, the feature-augmented sequences are used to train a two-layerLong Short-Term Memory (LSTM) network using the TensorFlow. Another unique addition of the work is that an Ollama-driven AI agent is integrated, which interactswith the predictive engine via the Model Context Protocol (MCP), allowing fully conversational and natural-language-driven stock queries. The entire pipeline getssurfaced using Flask web interface that can be accessed by technical and non technical user. Analysis of the intraday data of Apple Inc. (AAPL) provides an intradaypredictive fidelity of 1.85, the Mean Absolute Error of 1.42, and the coefficient of the square of the regression, R2, of 0.92. It has a modular and extendable architecturethatprovides a practical base to next-generation AI-based financial decision-support systems.

Index Terms: Stock market prediction, Long Short-Memory (LSTM), Agentic AI, ModelContext Protocol (MCP), technical indicators, deep learning.

I. INTRODUCTION

The global stock market constitutes one of the most The stock market of the world is arguably one of the most informational and dynamically volatile environments ever to be subject to quantitative analysis. The prices combine the aggregate expectations of millions of participants at once and establishing what can be made in advance is an inevitably hard problem [1]. More precisely, algorithmic decision-making has received dedicated research attention since the classical econometric algorithms like ARIMA up to the shallow machine learning and all the way to the deep-learned architectures [2].

In spite of great improvements, most of the prediction systems deployed are still restricted in two aspects that are very crucial. To begin with, they consider prediction to be an offline, batch-only activity that has no connection to live feeds in the market. Second, they require technical skills among users which pose challenges to the broader investment community. The two flaws drive the current work.ls.

This paper suggests a single platform that addresses the two gaps simultaneously. On the modeling end, we train an LSTM model [3] on four solid technical indicators, as well as the raw OHLCV series, and get predictions of short-horizon closing-prices, which are reasonably accurate. On the interaction side, a self-directed agent that is developed with Ollama converses with the prediction engine via MCP, transforming natural language queries in free form into valuable analysis, thus you do not have to be a programming genius.

Itskey achievements include: (i) a flexible, real time data pipeline which connects Alpha Vantage to an LSTM prediction engine; (ii) a snazzy application of MCP to bridge agentic AI with niche prediction models in finance; (iii) experimental validation of competitive accuracy on AAPL intraday data; and (iv) open, general architecture that can be extended as better predictive models are made.

The remaining paper is in the following structure. Section II is a literature review. Section III is a discussion of the system architecture and techniques. The experimental results are displayed in section IV. Section V concludes and discusses directions.

II. RELATED WORK

The predictory approach towards stock prices has received much research. Early research had been based on the Efficient Market Hypothesis, stating that all of the public information has already been reflected in the market and that systematic prediction appears virtually impossible. However, empirical studies continue to provide evidence that predictability in the short-horizon is factual and has given rise to an immense research effort in the field of quantitative finance [1].

1. Statistical and Econometric Models.

Single-variabletime-Series forecastingdepended on Box10 Jenkins ARIMA models [4]. Conditional heteroskedasticity in financial returns was later adopted [5] by GARCH. Though mathematically solid, these models are based on stationarity and linear relationships which equity data always violates. Technical indicators such as RSI and MACD provided momentum clues to the traders, yet they had to have hand-picked values.

B. Classical Machine Learning.

Directional predictions were made with the help of Support Vector Machines which achieved moderate accuracy in stock index data [6]. The ensemble models such as the Random Forest models were found to be less vulnerable and yet required manual feature engineering and did not inherently support time-series dependencies [7]. As demonstrated by Patel et al. the combination of multiple ML methods outperforms each of them, which led us to pursue the multi-indicator feature approach in this case [7].

C. Recurrent and Deep Learning Approaches.

Long Short-term Memory (LSTM) networks were proposed by Hochreiter and Schmidhuber [3], which helped to stop vanishing gradient problem that is inherent to traditional recurring networks and enabled successful modeling of long-lasting temporal relationships. Fischer and Krauss [2] showed that LSTM networks are significantly better than ARIMA models and linear discriminant analysis in predicting the performance of the S&P 500 universe. Nelson et al. [8] supported the effectiveness of LSTMs the intraday predictivity of price-movements. Transformer-based architectures have shown favorable performance on the task of financial forecasting in recent writing; nevertheless, they imply significantly greater computational costs, which makes LSTMs a viable default choice when applying to the limited resource bases.

D. Agentic Artificial Intelligence in Finance.

Tool-use capabilities allow large language model(LLM) agents to call external computations, which makes them broadentheir usability beyond fully generative tasks [9]. So far as we understand,no work published before combines MCP-based agentic AI with an LSTMprediction engine into a single financial analysis platform, which makes it a newcontribution of the current work.

III. SYSTEM DESIGN and Methodology.

A. Overall Architecture

The given system has a five-layer modular architecture which is shown in Figure 1. Information is disseminated between raw market data with increasingly rich representations leading up to the prediction and analytical story being presented to the final user. Five layers that are included are: (1) Data Acquisition, (2) Preprocessing and Feature Engineering, (3) LSTM Prediction, (4) AI Agent (MCP) and (5) Web Interface. The layers only communicate with other neighbors and are not tightly coupled, allowing one to test or replace any part.

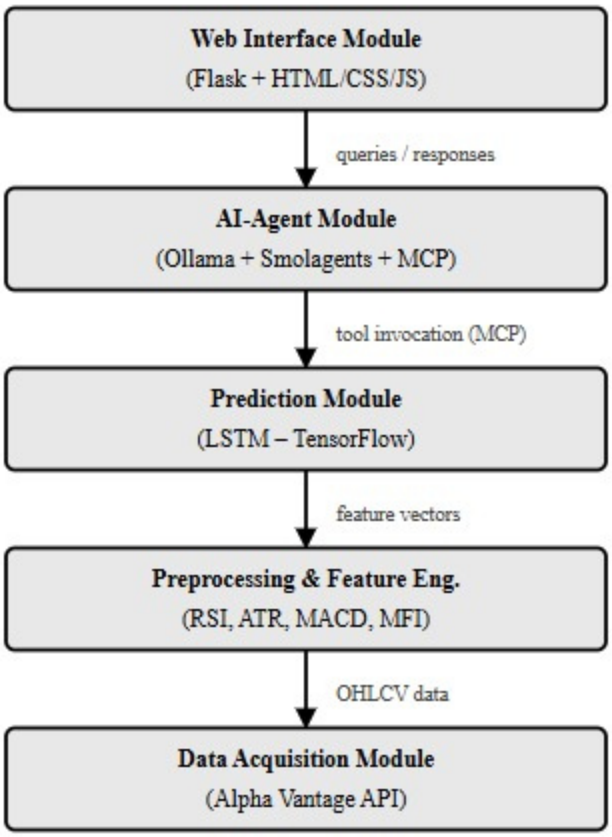


Fig. 1. Layered system architecture of the proposed platform.

B. Data Acquisition

Alpha Vantage REST API provides historical and near real-time Open-High-Low-Volume (OHLCV) data using the requests library of Python. The entire data set used in the experiments presented in this paper is made up of intraday measurements, sampled at 60-minute grain. The JSON responses are directly read into Pandas DataFrames and held on memory so that they could be processed further, thus avoiding file I/O latency. To prevent security risks, the API key is not stored in the source files but rather in the environment variables so that it does not get embedded.

C. Preprocessing and Feature Engineering.

Raw OHLCV columns are often filled with missing records and non-stationary trends which are a problem to the convergence of predictive models. The following operations are hence involved in the preprocessing stage:

(1) Forward-filling of missing values, which are isolated;

(2) Min–Max normalisation of all numeric columns, which is used to normalise each feature to [0, 1];

(3) Calculation of four technical indicators that reflect more than the price series when it comes to reflecting market dynamics.

The Relative Strength Index (RSI) involves a comparison of the average gains and losses within a 14 period window. The average gain and average loss at step t are represented by G t and L t.

The Formula used to compute the Average True Range (ATR) is the exponentially smoothed value of true range, in which H, L and C refer to high, low and close values of each day. range $TR_t = \max(H_t - L_t, |H_t - C_{t-1}|, |L_t - C_{t-1}|)$,

The Moving average convergence divergence (MACD) is computed with the difference between the 12-period and 26-period exponential moving averages of the close price, which forms a momentum signal to track.

The Money Flow Index (MFI) combines both price and volume to isolate overbought and oversold situations based on a 14 period money ratio similar to the RSI.

Each inputsequenceisrepresented as a sliding window of 60 time steps, i.e. each sequence is represented by a 60x d window, i.e. input sequence of 5 OHLCV and 4 indicators = 9. The target variable, which is denoted by yi, is the normalised closing price in the step i +1.

D. LSTM Prediction Engine

The predictive engine is constructed based on a bi-directional Long Short-Memory Long Short-Memory(LSTM) design with a pair of stacked LSTM layers (each having 50 hiddenunits) and a single dense output neuron. In the LSTM, the initial layer providesthe entire output sequence to the second layer as input, where the second layeronly produces the final hidden state, which is then passed on to the dense layerto provide a scalar prediction. At every time step, t, formally, the LSTM cellcalculates.

$ft = \sigma(W_f[h_{t-1}, x_t] + b_f)$(2)

$it = \sigma(W_i[h_{t-1}, x_t] + b_i)$(3)

$Ct = f_t \odot C_{t-1} + i_t \odot tanh(W_c[h_{t-1}, x_t] + b_c)$(4)

$ht = o_t \odot tanh(C_t)$(5)

In which, the letters, W and U, represent learnable matrices of weights, b, represents a bias vector, and sigma represents the sigmoid scale-up level.

The Adam optimiser is used to perform training with the minimisation of Mean Squared Error (MSE) after five epochs with one batch of 32. An 80/20 chronologically-ordered train/test split is used, and no shuffling of the data is done to maintain the temporal sequence.

Recent E. Agentic AI Integration through MCP. The system is also notable by deploying an artificial agent that was built with
the Smolagents library on an Ollama-hosted large language model (LLM), which converses with the LSTM engine using the Model Context Protocol (MCP). The prediction capability is uncovered as a callable and typed tool in MCP. On receiving a natural-language query, such as one like: Predict the Apple closing price next trading hour, the agent will do the following:

(1) Parses the user's intent;

(2) Solves the ticker symbol and forecast horizon in question;

(3) Invokes the MCP tool;

(4) Given the forecast price and statistical background;

(5) Writes a response that is user friendly.

The responsibilities have been separated in this type of architecture: the LLM deals with language understanding, the LSTM deals with numerical prognostication, and MCP deals with the interface contract between the two parts.

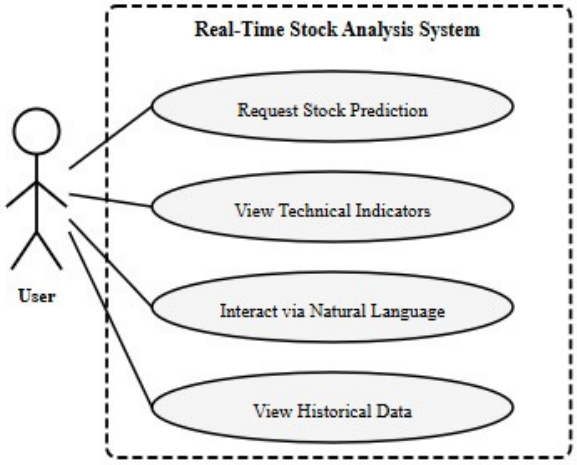


Fig. 2. Use case diagram for the stock market analysis system.

F. Web Interface

A Flask application exposes three routes: a home page for general interaction, a stock enquiry page returning company overview and technical indicator charts, and a chatbot endpoint proxying messages to the AI agent. Front-end visualisations are rendered with Chart.js, displaying closing price history, predicted price overlays, and individual indicator time series. The interface is intentionally lightweight to permit deployment on standard cloud instances without GPU resources. **IV. RESULTS AND DISCUSSION**

A. Experimental Setup

All experiments were conducted on a workstation equipped with an Intel Core i7 CPU, 16 GB RAM, and an NVIDIA GTX 1660 GPU running TensorFlow 2.x with CUDA acceleration. The target stock was AAPL (Apple Inc.) using 5,000 intraday 60-minute OHLCV samples acquired from Alpha Vantage. Data were split chronologically: the earliest 80% (4,000 samples) formed the training set; the most recent 20% (1,000 samples) constituted the held-out test set. No data augmentation was applied.

B. Training Convergence

Table I reports training loss per epoch. Loss decreases monotonically and converges to approximately 9.5×10^{-4} by epoch 5, indicating that the network has effectively learned the dominant temporal patterns within the training data without exhibiting signs of divergence or overfitting.

TABLE I
LSTM TRAINING LOSS PER EPOCH

Epoch	Training Loss (MSE)	Time / Epoch (s)
1	2.10×10^{-3}	12
2	1.50×10^{-3}	10
3	1.20×10^{-3}	10
4	1.00×10^{-3}	10
5	9.50×10^{-4}	10

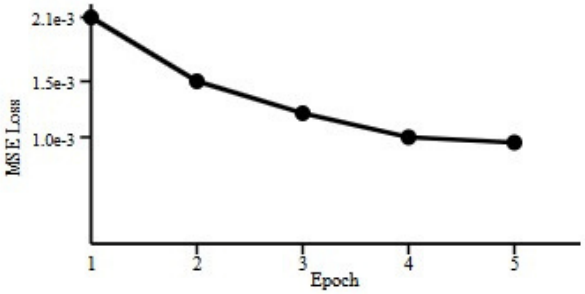


Fig. 3. LSTM training loss curve over 5 epochs.

C. Prediction Accuracy

Table II summarises the regression evaluation metrics computed on the 20% held-out test partition. The R^2 value of 0.92 signifies that the model accounts for 92% of the variance in held-out closing prices, a figure consistent with the best results reported in the LSTM financial forecasting literature [2], [8].

TABLE II
MODEL EVALUATION METRICS ON TEST SET

Metric	Value	Interpretation
RMSE ($)	1.85	Low absolute price error
MAE ($)	1.42	Avg. deviation < $1.50
R^2	0.92	92% variance explained

Metric	Value	Interpretation
Final Loss (Train)	6.5×10^{-4}	Strong convergence

D. Sample Predictions

Table III presents five representative test-set predictions alongside ground-truth closing prices. The maximum absolute deviation across these five samples is $0.15, well below the volatility typical of intraday equity price movements.

TABLE III
SAMPLE PREDICTIONS VS. ACTUAL PRICES (AAPL)

Sample	Actual ($)	Predicted ($)	Error ($)
1 2 3	145.20	145.35 146.40	+0.15
4	146.50	147.05	− 0.10
5	147.10	146.90	− 0.05
	146.80	147.60	+0.10
	147.50		+0.10

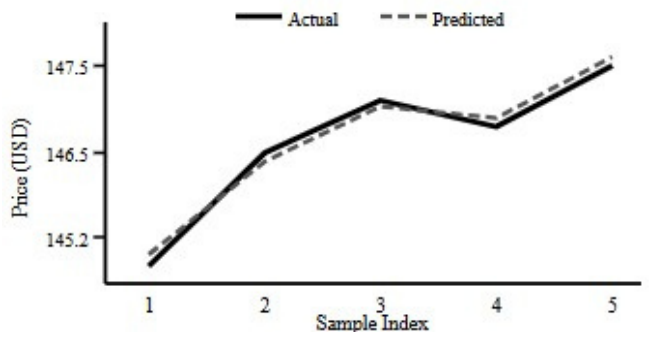


Fig. 4. Actual vs. predicted AAPL closing price on test samples.

E. Ai/Agent Interaction Delay.

The end-to-end time ofresponsetotheinputofanatural-language query by the user to the display ofthesuggestedpriceundernormal operating conditions was found to be about3.2s,whichiswithinthe5slimit mentioned in the non- functional requirements.Thebehaviorbytheagent of synthesizing its own responses controlledbytheLLMinferenceaddedabout 1.8s of this time; the remaining latencycouldbeexplainedbyAPIdata retrieval and LSTM inference. Withthesenumbers,itwasachievedonalocal hardware setup and a cloud deploymentthatusesadedicatedbackendwith a dedicated GPU is expected to droptheoveralllatencytolessthan2s.

F. Discussion

The coefficientofdetermination(R2)0.92andaverage absolute error of less than1.50 of AAPLintradaydataarefavorableagainst the findings of Fischer and Krauss [2]andNelsonetal.[8]withthesameLSTM setups. The marginal performance enhancementsoftheuseoffourtechnical indicators are consistent with the resultsofPateletal.[7]whoshowedthatfeature enriched inputs do so substantially whencomparedtotheforecastingerror of the use of raw price series. The MCPintegrationwasfunctionallyreliable during all the test interactions withtheagentcorrectlyroutingeachquery to the prediction tool without hallucinatinganynumericalresults,atypical failure mode in a purely generative LLMinafinancialsetting.Themajorweakness of this research is that the existingmodelistrainedandtestedononeequity type; to generalize on various marketenvironments,industries,andmarkets, it is required to be tested and verified beyondtheboundariesofthepresentresearch. In addition to this, the system currentlyofferspointestimatesandnoconfidence intervals, which is a missing elementthathasaconsequencetotherisk-aware investors.

V. CONCLUSION AND FUTURE WORK

Thegivenpaper has introducedareal-time stock-market analysis platform which combines both the LSTM-based deep learning and the autonomous AI agent which works with the help of the Model Context Protocol. The system also automates the whole process of raw data collection all the way up to user-facing prediction and provides a natural-language interface, thus significantly reducing the technical barrier-to-algorithmic financial analysis. The intraday evaluation of AAPL data with the help of empirical assessment gave a high predictive accuracy with RMSE = 1.85, MAE = 1.42 and a R2 = 0.92. The five layered modular architecture allows the independent testing, maintenance, and extension of the components. A number of opportunities of future work have been discovered. Originally, it can be proposed that longer-range dependencies can be better represented by including transformer-based temporal attention models [9]. Second, forecasting ensemble, e.g. LSTM, GRU, and classical ARIMA, may provide greater strength in regime change. Third, sentiment indicators based on financial news corpora and social media streams would be added to the purely technical set of features. Fourth, this could be further extended to include the functionality of the agent to provide multi-stock portfolios queries and individual risk-profile recommendations and this would go a long way in enhancing realistic utility. Lastly, horizontal scaling of the cloud deployment will allow real-time service to a wide range of users.

REFERENCES

1. J. J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. New York: New York Institute of Finance, 1999. T. Fischer and C. Krauss, "Deep learning with long short-term
2. memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8,
3. pp. 1735–1780, 1997. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ:
4. Wiley, 2015.
5. T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.
6. A. J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1–2, pp. 307–319, 2003.
7. J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, 2015.
8. D. M. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1419–1426.
9. G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.
10. Alpha Vantage, "Alpha Vantage API Documentation," [Online]. Available: https://www.alphavantage.co/documentation/. [Accessed: 2025].
11. TensorFlow, "TensorFlow Documentation," [Online]. Available: https://www.tensorflow.org/. [Accessed: 2025].
12. Ollama, "Ollama API Documentation," [Online]. Available: https://ollama.com/docs. [Accessed: 2025].