International Journal of Science and Technology (IJST) | ISSN(O):2049-7318 |2(3):66-69, 2026

STOCK MARKET PREDICTION USING MACHINE LEARNING

Mr. P. Bhaskar[1], S. Sania Mehek[2], C.P. Nanditha[3], K. Navya[4], M. Navya[5]

[1]Asst. Professor, [2345]UG Students, Department of Artificial Intelligence and Data Science,
Aditya College of Engineering, Madanapalle – 517325, Annamayya Dist., A.P., India
E-mail: shaiksaniamehek1004@gmail.com

Abstract: Stock markets are a fundamental pillar of the global financial ecosystem, enabling capital formation and wealth growth. Predicting stock prices is inherently challenging owing to their highly volatile, non-linear, and time-dependent nature. This paper presents a stock price prediction system developed using machine learning and deep learning techniques. The system dynamically retrieves historical stock market data via the Yahoo Finance API and applies multiple predictive models—including Linear Regression, Decision Tree, Random Forest, Support Vector Machine, K-nearest neighbours, XGBoost, Extra Trees, and Long Short-term memory networks—to forecast future stock prices. A robust data preprocessing pipeline involving missing value treatment, normalization using Standard Scalar, and feature engineering (50-day and 200-day SMAs) is implemented to ensure data quality. Model performance is assessed using Mean Absolute Error and the R-squared score. Experimental results confirm that LSTM achieves the highest accuracy, and the system is deployed via an interactive Streamlet web interface enabling real-time stock analysis accessible to both technical and non-technical users.

Index Terms: Stock price prediction, Machine learning, deep learning, LSTM, Time-series forecasting, financial analytics, XG Boost, Extra Trees, python, Streamlit

I. INTRODUCTION

The stock market is one of the most critical components of the global financial system, providing a platform through which individuals, institutions, and corporations trade equity instruments, thereby facilitating capital allocation, investment and wealth creation. Stock price behaviour is governed by a complex interplay of internal factors—such as corporate earnings, management strategy, and financial performance—and external macroeconomic variables including inflation, interest rates, GDP growth, geopolitical events, and global economic sentiment. Psychological forces such as investor speculation and herd behaviour further compound the difficulty of prediction.

Historically, two analytical schools have guided market analysis: fundamental analysis, which evaluates intrinsic company value through financial statements and industry indicators, and technical analysis, which identifies patterns in historical price and volume data. While both approaches yield insights, they are limited in processing high-dimensional data or adapting to rapidly evolving market dynamics.

With advance in computation and large-scale financial datasets, machine learning has emerged as a dominant paradigm for automated, data-driven prediction. Algorithms such as Random Forest, XG Boost, and Support vector regression have demonstrated substantial accuracy gains over classical statistical baselines. More recently, deep learning architectures—particularly LSTM networks—have excelled at modelling temporal dependencies intrinsic to financial time-series data.

This paper presents the design and implementation of an intelligent Stock Price Prediction System integrating eight ML and DL models within a Streamlit-based interactive deployment framework.

A. Problem Statement

Despite advances in computational finance, accurate stock price prediction remains an open challenge. Markets exhibit extreme volatility driven by unpredictable events and display non-linear, time-dependent patterns that defeat simplistic modelling approaches. Investors and analysts lack accessible, automated paltforms that can simultaneously benchmark multiple ML models, visualize historical trends, and generate predictions from live data—motivating the development of the unified system presented in this paper.

B. Objectives

The primary objectives are: (i) To collect and preprocess historical stock market data from reliable financial sources; (ii) To implement and evaluate multiple ML and DL algorithms for stock price prediction; (iii) To develop a deep learning LSTM model optimized for sequential time-series forecasting; (iv) To compare model performance using standardized evaluation metrics and (v) To deploy an interactive Streamlit web interface for real-time stock visualization and prediction.

II. LITERATURE REVIEW

Early research on stock price prediction relied on statistical models such as ARIMA(Auto Regressive Integrated Moving Average), which perform adequately for stationary, linear time series but fail to generalize to the volatile and non-linear nature of financial markets[1]. ARIMA assumes linearity and stationarity—assumptions that rarely hold in practice, making it inadequate for capturing sudden market regime changes or complex nonlinear price dynamics.

Subsequent work investigated ML-based approaches. Kim(2003) [2] applied Support Vector Machines to financial time-series, demonstrating improved performance over ARIMA. Ensemble methods such as Random Forest(Breiman, 2001) [3] and Xg Boost (Chen and Guestrin, 2016) [4] have been widely adopted for structured financial data, as they reduce overfitting and effectively model non-linear feature interactions.

Fischer and Krauss (2018) [5] demonstrated that LSTM networks significantly outperform classical statistical and

ML baselines in predicting stock returns, attributing the improvement to the architectures ability to retain long-range historical context. Sezer et al. (2020) [6] confirmed LSTM as the dominant architecture through a systematic literature review.

Zhang (2003) [7] proposed a hybrid ARIMA-neural network model demonstrating that combining linear and non-linear components improves forecasting accuracy. A key limitation across reviewed works is reliance on static datasets. The present study addresses this gap with an open, multi-model prediction platform with a dynamic data acquisition pipeline and interactive Streamlit architecture.

III. METHODOLOGY AND PROPOSED SYSTEM

The proposed Stock price prediction System follows a multi-stage pipeline encompassing data acquisition, preprocessing, feature engineering, model training, evaluation, and interactive deployment. The methodology adopts a data-driven approach in which historical stock price data serves as the primary source of information. Fig.1 illustrates the complete system architecture.

A. Data Collection

Historical daily stock price data is dynamically retrieved from Yahoo Finance using the y finance python library. The system retrieves six primary attributes per trading day: Date, open, High, Low, Close, and Volume. The AAPL dataset spans 2015-2024 (-2,265 trading days), exposing trained models to diverse market conditions including both bull and bear cycles. The API-based approach supports equities, market indices, ETFs, and cryptocurrencies.

B. Data Preprocessing

The preprocessing pipeline operates through four sequential stages (Fig. 3). (1) Data Cleaning: duplicates removed; missing values imputed via forward-fill. (2) Feature Engineering: SMA50 and SMA200 appended along with lag features and rolling statistics. (3) Normalization: Standard Scaler transforms all features to zero mean and unit variance. (4) Train-Test Split: 80/20 chronological split prevents data leakage.

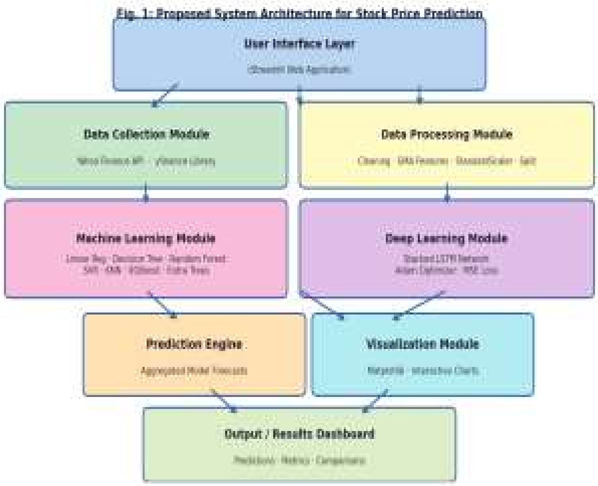

Fig. 1: Proposed System Architecture for Stock Price Prediction

C. System Architecture

The system follows a modular, layered architecture comprising seven functional components (Fig. 1): (1) User Interface Layer (Streamlit); (2) Data Collection Module; (3) Data Processing Module; (4) ML Module (seven classical algorithms); (5) Deep Learning Module (LSTM); (6) Prediction Engine; and (7) Visualization Module(Matplotlib dashboards). This layered design ensures modularity and easy extensibility.

IV. ALGORITHM IMPLEMENTATION

The system implements eight predictive algorithms spanning classical ML regressors and a sequential deep learning model. Table I summarizes all models with category, R^2 score, MAE, and training time.

A. Linear Regression

Linear Regression models the relationship through: $\hat{y} = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n + \varepsilon$. Coefficients are estimated via OLS. Despite its simplicity(<1 sec), the linear assumption limits performance with non-linear financial data. Baseline: $R^2 = 0.85$, MAE = 3.20 USD.

B. Decision Tree Regressor

The Decision Tree partitions the feature space through recursive binary splitting. Depth constraints (max_depth = 10) and pruning improve generalization: $R^2 = 0.87$, MAE = 2.95 USD.

C. Random Forest Regressor

Random Forest trains n = 100 Decision Trees on bootstrapped subsets, averaging outputs to substantially reduce variance: $R^2 = 0.91$, MAE = 2.20 USD.

D. Support Vector Regression (SVR)

SVR uses the RBF kernel to map inputs into a high-dimensional feature space enabling complex non-linear modelling. SVR achieved R^2=0.89, MAE=2.60 USD.

E. K-Nearest Neighbors (KNN)

KNN identifies K = 5 most similar historical observations using Euclidean distance and averages their target values. Non-parametric, no training phase required: $R^2 = 0.88$, MAE = 2.75 USD.

F. XGBoost Regressor

XGBoost sequentially trains shallow trees correcting residual errors with built-in L1/L2 regularization, delivering the highest accuracy among classical ML models: $R^2 = 0.93$, MAE = 1.60 USD.

G. Extra Trees Regressor

Extra Trees selects features and split thresholds randomly at each node, further reducing variance with faster training (~6 sec): $R^2 = 0.92$, MAE = 2.05 USD.

H. Long Short-Term Memory (LSTM)

LSTM is a specialized RNN designed to overcome the vanishing gradient problem via three gating mechanisms: the input gate controls what new information is stored; the forget gate decides what historical information to discard; the output gate determines what is passed to the next time step. A persistent memory cell enables selective retention over arbitrarily long sequences.

The stacked LSTM network comprises two LSTM layers(64 units each) followed by a Dense output layer. Historical closing prices are normalized and reshaped into overlapping 60-day sequences. Training uses Adam Optimizer(lr = 0.001) with MSE loss over 50 epochs (batch = 32). Best overall performance: $R^2 = 0.95$, MAE = 1.30 USD—a 59% reduction vs the baseline.

V. RESULTS AND DISCUSSION

All eight models were trained on 80% of the chronologically ordered AAPL dataset (2015-2024, ~2,265 trading days) and evaluated on the remaining 20% holdout(~566 days). Table 1 presents the complete comparative results: Fig. 2 provides a visual comparison.

Table1: Comparative Model Performance on AAPL Dataset (2015-2024)

Model	Category	R^2 Score	MAE (USD)	Train Time
Linear Regression	Classical ML	0.85	3.20	< 1 sec
Decision Tree	Classical ML	0.87	2.95	< 1 sec
K-Nearest Neighbors	Classical ML	0.88	2.75	< 1 sec
Sup. Vector Reg.	Classical ML	0.89	2.60	~3 sec
Random Forest	Ensemble ML	0.91	2.20	~8 sec
Extra Trees	Ensemble ML	0.92	2.05	~6 sec
XGBoost	Ensemble/Boosting	0.93	1.60	~5 sec
LSTM	Deep Learning	0.95	1.30	~120 sec

Table I reveals a clear performance hierarchy. Linear Regression ($R^2 = 0.85$, MAE = 3.20 USD) captures broad linear trends but fails to model non-linear dynamics. Decision Tree: $R^2 = 0.87$ (MAE = 2.95). KNN: $R^2 = 0.88$ (MAE = 2.75). SVR: $R^2 = 0.89$ (MAE = 2.60).

Ensemble methods substantially outperform individual models. Random Forest: R^2=0.91 (MAE=2.20).Extra Trees: $R^2 = 0.92$ (MAE = 2.05). XG Boost: $R^2 = 0.93$, MAE = 1.60 USD—best among classical ML.

LSTM achieved the best overall: $R^2 = 0.95$, MAE = 1.30 USD—a 59% reduction vs the baseline. Its gating mechanisms selectively retain and discard historical context, effectively modelling auto regressive dependencies. The 60-day lookback captures medium-term trends invisible to regression-based approaches, consistent with Fischer and Krauss (2018) [5] and Sezer et al.(2026) [6].

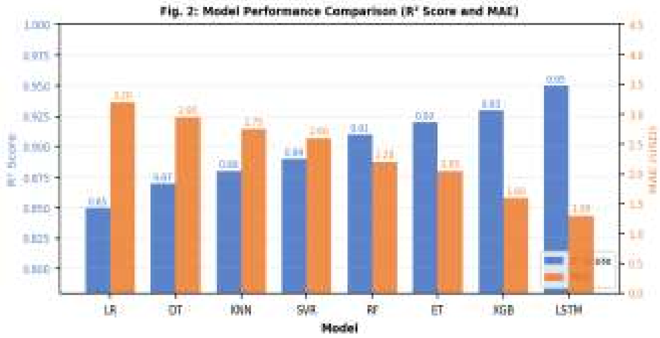

Fig. 2: Model Performance Comparison – R^2 Score and MAE (USD) across all eight models

Fig. 2 visually confirms the monotonic improvement in R^2 and reduction in MAE as model sophistication increases. The Streamlit interface provides: (i) stock symbol input; (ii) historical trend charts; (iii) tabular data view; (iv) prediction output graphs; (v) model comparison charts; and (vi) performance metrics panel for each model.

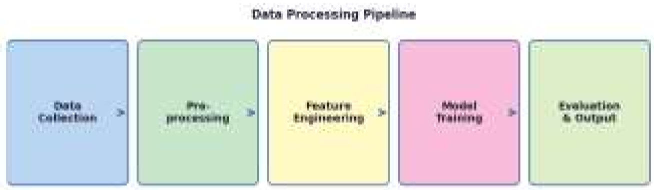

Fig. 3: Data Processing Pipeline – Sequential stages from raw data to model-ready features

VI. CONCLUSION

This paper presented a comprehensive Stock Price Prediction System integrating eight ML and DL models within a streamlined, interactive deployment framework. Historical AAPL stock data was dynamically retrieved via the Yahoo Finance API, subjected to robust preprocessing including SMA-500 and SMA-200 feature engineering, and used to train and evaluate multiple predictive algorithms across an 80/20 chronological train-test split.

Experimental results confirm that LSTM outperforms all classical ML baselines ($R^2 = 0.95$, MAE = 1.30 USD). Ensemble methods XGBoost ($R^2 = 0.93$) and Extra Trees ($R^2 = 0.92$) also demonstrated strong performance. Linear Regression at $R^2 = 0.85$ served as an essential interpretability benchmark. Stock market prediction is inherently probabilistic; external shocks remain beyond the modeling capacity of data-driven systems. The system's outputs are intended as analytical aids, not guaranteed investment recommendations.

VII. FUTURE SCOPE

Several promising directions exist for extending the proposed system. Integration of financial news sentiment analysis using NLP techniques can incorporate qualitative market signals that current price-only models overlook. Hybrid LSTM-Attention models and Transformer-based architectures warrant exploration for capturing non-local temporal dependencies.

Real-time data streaming pipelines would enable continuously updated predictions foe algorithmic trading. Portfolio-level analysis—optimizing allocations using predicted returns and covariance matrices—represents a valuable extension. Additional technical indicators (RSI, MACD, Bollinger Bands), Cloud-based deployment on AWS or GCP, and mobile application development would broaden accessibility and support large-scale concurrent usage.

Acknowledgement.

The authors express sincere gratitude to the Department of Artificial Intelligence and Data Science, Aditya College of Engineering, Madanapalle, for providing the necessary infrastructure and academic support. The authors also acknowledge the open-source communities behind scikit-learn, TensorFlow/Keras, XGBoost, yfinance, and Streamlit.

REFERENCES.

1. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed. Wiley, 2015.
2. K. J. Kim, “Financial time series forecasting using support vector machines,” Neurocomputing, vol. 55, no. 1–2, pp. 307–319, 2003.
3. L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
4. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD, 2016, pp. 785–794.
5. T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.
6. O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, “Financial time series forecasting with deep learning: A systematic literature review,” Applied Soft Computing, vol. 90, p. 106181, 2020.
7. G. P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” Neurocomputing, vol. 50, pp. 159–175, 2003.
8. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
9. M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2016. [Online]. Available: https://www.tensorflow.org
10. J. D. Hunter, “Matplotlib: A 2D graphics environment,” Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.
11. W. McKinney, “Data structures for statistical computing in Python,” in Proc. 9th Python in Science Conf., 2010, pp. 51–56.
12. C. R. Harris et al., “Array programming with NumPy,” Nature, vol. 585, pp. 357–362, 2020.
13. Streamlit Inc., Streamlit Documentation, 2025. [Online]. Available: https://streamlit.io
14. Yahoo Finance, Stock Market Data, 2025. [Online]. Available: https://finance.yahoo.com

About the Authors.

Mr. P. Bhaskar is an Assistant Professor in the Department of Artificial Intelligence and Data Science at Aditya College of Engineering, Madanapalle. His research interests include machine learning, data science, and intelligent systems.

S. Sania Mehek is a UG Student (B.Tech, AI & DS) at Aditya College of Engineering, Madanapalle. Her interests lie in deep learning and financial analytics.

C.P. Nanditha is a UG Student (B.Tech, AI & DS) at Aditya College of Engineering, Madanapalle, with interests in machine learning and data preprocessing.

K. Navya is a UG Student (B.Tech, AI & DS) at Aditya College of Engineering, Madanapalle, with a focus on predictive modelling and visualization.

M. Navya is a UG Student (B.Tech, AI & DS) at Aditya College of Engineering, Madanapalle, interested in ensemble methods and time-series analysis.