Stock Price Prediction of Major Technology Companies Using Machine Learning

Tayo P. Ogundunmade[1*]; Olayinka B. Ayeni[2]

[1,2]Department of Statistics, University of Ibadan, Ibadan, Nigeria.

Corresponding Author: Tayo P. Ogundunmade[1*]

Publication Date: 2025/11/06

Abstract: The stock market is inherently volatile and influenced by a multitude of dynamic, interrelated factors ranging from macroeconomic indicators and geopolitical events to investor sentiment and technological innovation. For investors and financial analysts, accurately predicting stock prices remains a persistent challenge, particularly in the technology sector, where companies like Apple, Microsoft, Google, and Netflix experience rapid growth, frequent innovation cycles, and high market sensitivity. This project applies machine learning to predict future stock prices for five major tech firms, which include Apple, Tesla, Amazon, Google, and Microsoft. Past stock price datasets available on Kaggle were used, and by leveraging historical adjusted closing prices alongside engineered inputs such as daily returns, moving averages, rolling volatility, and lagged values. We train and compare sequential models, including TCNs, Informer architectures, and LSTMs. The analysis revealed several important insights about the current state of machine learning-based stock prediction. Model performance limitations were identified, with modest accuracies ranging from 49% to 63% for predicting whether stock prices would increase by more than 1.5% within five days. Stock-specific predictability and market insights were also highlighted, with Microsoft emerging as the most predictable (63% accuracy) due to its relatively stable price movements, while Tesla proved most challenging to forecast (49-50% accuracy). Our findings demonstrate that thoughtful feature construction combined with advanced sequence modeling can uncover market dynamics and boost prediction accuracy, showcasing the tangible benefits of AI-driven analytics for informed financial decision-making.

Keywords: *Stock Price, Technology Companies, Machine Learning, Regression Models, Prediction.*

How to Cite: Tayo P. Ogundunmade; Olayinka B. Ayeni (2025) Stock Price Prediction of Major Technology Companies Using Machine Learning. *International Journal of Innovative Science and Research Technology*, 10(10), 2573-2586. https://doi.org/10.38124/ijisrt/25oct1245

I. INTRODUCTION

In the modern financial landscape, the stock market stands as a dynamic and complex system that reflects the economic pulse of industries, nations, and global trends. Among its most influential players are major technology companies such as Apple, Microsoft, Google, and Netflix—corporations whose innovations and market movements shape the digital age. Investors, analysts, and researchers have long sought reliable methods to predict stock prices, aiming to maximize returns and minimize risks. Traditional approaches, while valuable, often fall short in capturing the intricate patterns and nonlinear relationships inherent in financial time series data. This has paved the way for the integration of machine learning (ML), a subset of artificial intelligence (AI), into financial forecasting [1]. In the modern financial landscape, the stock market stands as a dynamic and complex system that reflects the economic pulse of industries, nations, and global trends. Among its most influential players are major technology companies such as Apple, Microsoft, Google, and Netflix—corporations whose innovations and market movements shape the digital age. Investors, analysts, and researchers have long sought reliable methods to predict stock prices, aiming to maximize returns and minimize risks. Traditional approaches, while valuable, often fall short in capturing the intricate patterns and nonlinear relationships inherent in financial time series data. This has paved the way for the integration of machine learning (ML), a subset of artificial intelligence (AI), into financial forecasting.

Machine learning offers a powerful toolkit for analyzing vast amounts of historical data, identifying hidden trends, and making data-driven predictions. Unlike conventional statistical models, ML algorithms can adapt to changing market conditions, learn from new data, and uncover complex dependencies that may elude human intuition. In recent years, the fusion of finance and machine learning has gained significant traction, leading to breakthroughs in algorithmic trading, portfolio optimization, and risk management. This project explores the application of machine learning techniques to predict the stock prices of leading technology companies, leveraging historical data and advanced models

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

to forecast future trends. Machine learning offers a powerful toolkit for analyzing vast amounts of historical data, identifying hidden trends, and making data-driven predictions. Unlike conventional statistical models, ML algorithms can adapt to changing market conditions, learn from new data, and uncover complex dependencies that may elude human intuition. In recent years, the fusion of finance and machine learning has gained significant traction, leading to breakthroughs in algorithmic trading, portfolio optimization, and risk management. This project explores the application of machine learning techniques to predict the stock prices of leading technology companies, leveraging historical data and advanced models to forecast future trends.

Stock price prediction has long been a complex challenge due to the dynamic, volatile, and non-linear nature of financial markets. Investors, analysts, and institutions seek reliable ways to forecast price movements, manage risk, and identify trading opportunities. Traditional statistical models often fall short in capturing the complex patterns and temp1zoral dependencies found in financial time series data. With the rise of artificial intelligence and deep learning, there is increasing interest in leveraging neural network architectures to model stock market behavior. Deep learning models, particularly those designed for sequential data such as Long Short-Term Memory (LSTM) networks, Temporal Convolutional Networks (TCN), and Transformer-based models like Informer have shown promise in capturing both short-term and long-term dependencies in time series.

Kaur *et al.* [2] further improved prediction performance by combining TCN and LSTM, reducing Root Mean Square Error (RMSE) by up to 10%. More recently, Wang et al. (2022) found that TCN outperformed LSTM in stability and efficiency across multiple stock datasets, solidifying its role as a strong alternative for financial time series forecasting. The study by Jiali Liang [3] titled *"Comparison of Price Prediction Based on LSTM, GRU, Random Forest, LSSVM, and Linear Regression, it was found that GRU achieved the highest accuracy (87.39%) in predicting Tesla's stock prices (July 2018–July 2022), outperforming LSTM, LSSVM, and Linear Regression, which suffered from overfitting. The results imply that advanced ML models (GRU/LSTM) are superior for capturing nonlinear stock trends, especially when combining technical and macroeconomic data. Hu et *al.* [4] and Yin et al. [5] emphasize LSTM's effectiveness in capturing long-term trends and sequential patterns in financial data. Their research supports the view that LSTM provides a strong baseline model for time series tasks in finance. For stock price prediction, deep learning, specifically LSTM, offers a significant advantage due to its ability to model complex, time-dependent relationships. While simpler models may be easier to interpret, they often lack the depth needed for robust forecasting in volatile markets. More recently, Transformer-based models like Informer [6] have been applied to stock price prediction, offering improved efficiency in handling long time series through sparse attention mechanisms. Studies by Crow et al. [7] and Li et al. [8] show that Informer consistently outperforms LSTM and TCN in multivariate financial forecasting tasks, particularly when capturing long-range dependencies across multiple assets. Its ability to scale with data and maintain low error rates makes it a strong contender in the evolution of deep learning models for stock market prediction.

In the study by Bukhari et al. [9], researchers evaluated multiple ML algorithms—including linear regression, decision trees, and support vector machines (SVM)—on historical stock data from major companies. The dataset included daily closing prices, volume, and technical indicators over a five-year period. Their findings revealed that ensemble models like random forests outperformed linear models, especially in capturing nonlinear trends and reducing prediction error. The study also highlighted the importance of feature selection. Models trained with engineered features such as moving averages and Bollinger Bands showed improved accuracy compared to those using raw price data alone. This underscores the value of domain knowledge in enhancing ML performance. Khan et al. [10] conducted a systematic review and performance analysis of deep learning models for stock price prediction. They focused on Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU). Using stock data from Apple and Google, the study found that LSTM models consistently delivered lower Mean Squared Error (MSE) and better trend prediction than traditional ML models. The researchers attributed LSTM's success to its ability to retain long-term dependencies and handle sequential data effectively. However, they also noted challenges such as overfitting and high computational cost, suggesting that model tuning and regularization techniques are essential for robust performance.

Kumbure et al. [11] explored hybrid models that combine ML algorithms with technical indicators and sentiment analysis. Her thesis applied Support Vector Regression (SVR) and LSTM to stock data from Netflix, integrating features like RSI, MACD, and news sentiment scores. The hybrid approach yielded higher predictive accuracy and better directional forecasting, especially during periods of market volatility. This study demonstrated that incorporating alternative data sources—such as news sentiment—can enhance model performance. It also emphasized the need for real-time data integration and adaptive learning mechanisms to respond to sudden market shifts. A recent experiment by Kurani et al. [12] implemented real-time prediction models using streaming data from Microsoft and Google. They used online learning algorithms and reinforcement learning to adjust trading strategies based on live market conditions. The study showed that reinforcement learning agents could outperform static models in dynamic environments, achieving higher returns and lower drawdowns. However, the study also cautioned against the risks of algorithmic trading, including latency issues, data noise, and regulatory constraints. It called for robust testing frameworks and ethical guidelines to ensure responsible deployment of ML in financial markets. Another empirical investigation by Sharma et al. [13] examined whether ML models trained on U.S. tech stocks could generalize to other markets. Using transfer learning techniques, they applied models trained on Apple and Microsoft to predict stock movements in Asian tech firms. The results were mixed:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

while basic trends were captured, performance declined due to differences in market structure, trading behavior, and regulatory environments. This study highlighted the limitations of model portability and the importance of localized training data. It also suggested that regional customization and hybrid modeling may be necessary for cross-market applications. A systematic review by Thakkar and Chaudhari [14] categorizes ML approaches into supervised, unsupervised, and reinforcement learning, with supervised learning being the most widely applied in stock prediction tasks. The study highlights the use of regression models (e.g., linear regression, support vector regression) and classification models (e.g., decision trees, random forests) for predicting price movements and trends. Yadav et al. [14] emphasize the growing role of deep learning, particularly Long Short-Term Memory (LSTM) networks, in time series forecasting. LSTM models are designed to retain long-term dependencies and have shown superior performance in predicting stock prices compared to traditional ML models. Zakhidov et al. [16] conducted a comparative analysis of ML models applied to technology stocks such as Apple and Google. Their findings suggest that ensemble models like random forests outperform single learners in terms of accuracy and robustness. The study also notes that feature engineering—especially the inclusion of technical indicators like RSI and MACD—significantly enhances model performance. Other researchers have explored hybrid models that combine ML algorithms with sentiment analysis, technical indicators, and macroeconomic variables. These models often yield better results, particularly during periods of market volatility.

Deep learning models, especially LSTM and GRU, have gained popularity for their ability to model sequential data. Studies show that LSTM networks outperform traditional models in capturing temporal dependencies and trend reversals. However, challenges such as overfitting, high computational cost, and lack of interpretability remain. Researchers have also experimented with convolutional neural networks (CNNs) for extracting features from financial time series, though their application is less common than LSTMs in this domain. This project aims to apply and compare these advanced deep learning techniques to the task of predicting stock price direction using historical daily OHLC (Open, High, Low, Close) data for five major technology companies: Google, Amazon, Tesla, Apple, and Microsoft. By building a system that can efficiently process financial time series, engineer relevant features, and evaluate model performance, this work contributes to the ongoing effort to apply AI in real-world financial forecasting.

II. MATERIALS AND METHODS

➢ *Data Collection and Preprocessing*

Historical stock data for five major technology firms which include Apple (AAPL), Tesla (TSLA), Amazon (AMZN), Google (GOOGL), and Microsoft (MSFT) was sourced from Kaggle financial datasets, though we only took a relevant portion of that data for our analysis. Each dataset included daily OHLC price data along with trading volume. Using R, the datasets were loaded and merged into a unified structure with a company identifier column to differentiate records. Essential libraries such as dplyr, tidyr, xts, and zoo support data manipulation and time series formatting. The preprocessing phase was particularly comprehensive and systematic, involving: handling missing values, date alignment and filtering.

➢ *Exploratory Data Analysis (EDA)*

To gain insights into the historical behavior of the stocks, descriptive statistics were computed for the Adjusted Close prices. These included mean, median, standard deviation, minimum, and maximum values. Visualisation through ggplot2 allowed for the creation of time series plots to observe trends, volatility bursts, and anomalies, giving an intuitive sense of each stock's movement across the selected timeframe.

➢ *Return, Correlation, and Volatility Analysis*

Daily Returns were calculated as the percentage change in adjusted close prices. These returns were pivoted to a wide format, enabling pairwise correlation analysis across the five companies. A correlation heatmap was generated using the corrplot package to visualize inter-stock relationships. 20-day rolling volatility was calculated for each stock to assess return variability. Additionally, 20-day and 50-day moving averages were plotted to detect trend shifts, momentum changes, and potential support/resistance levels.

- *Feature Engineering and Target Creation*

A major highlight of the pipeline was the creation of an extensive feature set designed to improve model learning and performance. This included lagged Features, technical indicators such as (Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Bollinger Bands and Stochastic Oscillator), volatility and momentum metrics, moving averages, market regime detection and target and creation such as (Next-day directional movement (up/down) or 5-day cumulative returns with user-defined return thresholds). After feature generation, the dataset was split using a 70/15/15 time-aware split to ensure future data was never included in model training. This preserved the temporal integrity crucial to forecasting. All continuous variables were standardised to zero mean and unit variance. Scaling was verified by checking the transformed variables' summary statistics (mean ~0, SD ~1), ensuring uniformity across companies.

➢ *Classical Linear Modelling*

As a benchmark, linear regression models were developed for each company using lagged features to predict next-day adjusted close prices[17].

$$\hat{Y}_{i+1} = \beta_0 + \sum_{i=1}^{p} \beta_i X_{t-i} + \varepsilon_t \tag{1}$$

Where;

$\hat{Y}_{i+1}$ = predicted adjusted close price for day $i+1$

X_{t-i} = lagged features (e.g., adjusted close, returns) at lag i

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

p = number of lag days used (in this case, up to 3)

β_0 = model intercept

β_i = regression coefficients learned from training data

ε_t = error term assumed to be normally distributed: $\varepsilon_t \sim N(0, \sigma^2)$

Each model was trained on the training subset, evaluated on the test subset and visualised using prediction vs. actual plots to assess performance intuitively These classical models served as a foundation for comparing the improvements gained through deep learning architectures.

➢ *Deep Learning Model Development*

Three distinct deep learning models were implemented using R with TensorFlow/Keras via Python 3.11. Each model was built for each company separately, using a 10-day lookback window to form input sequences.

- *Enhanced LSTM (Long Short-Term Memory):*

LSTM is a type of recurrent neural network (RNN) specifically designed to handle sequential data by maintaining long-term dependencies through its gating mechanism [18]. In this work, the LSTM model was trained on past stock price movements and engineered features to predict future directions. Its ability to remember relevant patterns over time made it suitable for capturing the temporal dynamics inherent in financial time series data.

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

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

$$C_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot C_t \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \odot tanh(C_t) \quad (7)$$

Where;

x_t= input at time t (e.g., feature vector for stock at time t)

h_t = hidden state (output) at time t

C_t = cell state (memory) at time t

f_t, i_t,o_t = forget, input, and output gates respectively

σ = sigmoid activation

$\odot$ = element-wise multiplication

$W.$, $b.$= weight matrices and biases

- *Fixed Temporal Convolutional Network (TCN):*

The TCN model is a convolutional approach adapted for sequential data. Unlike traditional RNNs, TCNs use dilated causal convolutions, which allow them to model long-range dependencies without the vanishing gradient issues often seen in RNNs. This model offers faster training and more stable performance across stocks, especially in detecting complex temporal patterns across multiple time scales[19].

$$y_t = \sum_{i=0}^{k-1} \omega_i \cdot x_{t-d.i} \quad (8)$$

Where;

x_t = input at time t

ω_i = learnable convolutional weights

k = filter size

d = dilation factor (e.g., 1, 2, 4...) controlling the receptive field

y_t = output at time t

- *Simple Informer (Transformer-based):*

The Informer model is a streamlined version of transformer-based architectures tailored for time series forecasting. It uses self-attention mechanisms to focus on the most relevant parts of the input sequence when making predictions. Although it simplifies standard transformer components (e.g., by excluding positional encodings), it is capable of handling longer sequences and capturing global dependencies efficiently. Its performance varies across stocks, offering unique insights compared to LSTM and TCN[20].

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (9)$$

Where;

Q, K, V = Query, Key, and Value matrices derived from the input

d_k = dimension of the key vectors

The dot product QK^T measures similarity between time steps

Multi-head attention combines multiple such mechanisms:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O$$

Each head is:

$$head_I = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (10)$$

Informer focuses on dynamic dependencies across time without positional encoding, which helps with flexibility in sequential financial data.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

➢ *Model Evaluation and Results Interpretation*

Performance was evaluated across multiple dimensions, like Average prediction accuracy per company, Direction-of-change accuracy for binary classification tasks, Model comparison to identify the top-performing architecture per stock, Model-win statistics summarising which architecture succeeded most often, Quartile summaries to understand the distribution of performance metrics and Automated CSV export of results. Visual plots were also generated to compare predicted vs. actual stock prices over time, helping to interpret how well each model adapted to real market behaviour. This methodology combines rigorous statistical analysis with advanced deep learning modelling to explore the predictability of stock prices among major tech firms [21]. By incorporating detailed data preprocessing, rich feature engineering (including 27+ features and technical signals), and diverse model architectures (LSTM, TCN, Transformer), the system presents a flexible and scalable framework. The results from this structured pipeline not only benchmark traditional approaches but also demonstrate the practical viability of deep learning for financial time series forecasting, opening the door for further experimentation and refinement in future research.

III. RESULTS AND DISCUSSIONS

This section analyzes the key results obtained from implementing three deep learning models for stock price prediction. The discussion reflects on the models' performance, accuracy metrics, and general behavior across various stock datasets, with emphasis on comparative strengths and limitations. By comparing the performance of LSTM, TCN, and Informer architectures across different companies, this chapter highlights how each model responds to stock market dynamics and varying data characteristics. The findings are evaluated in the context of the project’s objectives, particularly the assessment of prediction accuracy, model consistency, and the practical implications of deploying each architecture in real-world financial forecasting scenarios.

A. *Descriptive Statistics and Trends*

Before building deep learning models, we performed an extensive exploratory and predictive pre-analysis using R. The analysis was conducted on historical stock data from five major companies: Apple (AAPL), Tesla (TSLA), Amazon (AMZN), Google (GOOGL), and Microsoft (MSFT).

Table 1 Descriptive Statistics for Combined Stock Dataset (n = 3,760)

Variable	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Open	82.80	140.30	179.40	205.20	249.80	475.90
High	83.48	142.07	181.28	207.94	254.28	488.54
Low	81.43	138.16	176.93	202.30	245.62	464.46
Close	81.82	139.92	179.22	205.18	250.10	479.86
Adjusted Close	81.82	139.30	178.54	203.75	247.49	479.86
Volume	7.16M	28.39M	47.94M	58.26M	76.14M	318.68M
Daily Return	-0.140	-0.0117	0.0009	0.0006	0.0128	0.2192
Rolling Volatility (20d)	0.006	0.0148	0.0200	0.0225	0.0278	0.0689
MA20 (20-day avg)	86.33	137.59	177.96	202.12	243.72	450.99
MA50 (50-day avg)	90.55	137.92	178.08	200.27	242.96	435.84

Table 1 presented above summarize the key features of the stock dataset used in the pre-analysis. Stock prices (Open, High, Low, Close, and Adjusted Close) exhibit wide ranges, reflecting the diversity in pricing across companies like Tesla and Microsoft over the observation period. Volume shows significant variability, ranging from around 7 million to over 318 million shares traded daily, indicating differing levels of investor activity. Daily returns and rolling 20-day volatility highlight both the typical stability of stock movements and the occasional extreme fluctuations. Moving averages (MA20 and MA50) help reveal long-term trends, with values supporting the overall upward trajectory of most stock prices throughout the dataset. These statistics provide a foundational understanding of the data's structure and behavior before applying more advanced predictive models.

Table 2 Lagged Variables (Original Values)

Variable	Min	1st Qu.	Median	Mean	3rd Qu.	Max
Lag 1 (Adj. Close)	81.82	139.24	178.34	203.62	247.46	479.86
Lag 2 (Adj. Close)	81.82	139.15	178.28	203.49	247.39	479.86
Lag 3 (Adj. Close)	81.82	139.12	178.22	203.34	247.17	479.86
Return Lag 1	-0.140	-0.0116	0.0009	0.0006	0.0128	0.2192
Return Lag 2	-0.140	-0.0116	0.0010	0.0007	0.0129	0.2192

Table 2 summarises recent past behavior of stock prices and returns, which are critical for time series modeling. The Lag 1, Lag 2, and Lag 3 values for adjusted close prices are nearly identical in distribution, showing consistent pricing trends over short intervals. Similarly, Return Lag 1 and Return Lag 2 maintain the same range and quartiles, indicating stable short-term return dynamics. These features help capture temporal dependencies and are useful predictors in forecasting models.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

Table 3 Scaled Variables Summary (Standardized)

Variable	Min	Median	Max	NA Count
Adjusted_Scaled	-2.11	-0.10	3.99	0
Daily_Return_Scaled	-5.84	0.01	5.65	5
MA20_Scaled	-2.11	-0.07	3.48	95
MA50_Scaled	-1.78	-0.05	2.59	245
Rolling_Volatility_Scaled	-1.96	-0.16	3.32	100
Lag_1_Scaled	-2.12	-0.10	4.01	5
Lag_2_Scaled	-2.13	-0.10	4.04	10
Lag_3_Scaled	-2.14	-0.10	4.08	15
Return_Lag_1_Scaled	-5.84	0.01	5.65	10
Return_Lag_2_Scaled	-5.84	0.01	5.65	15

The standardized variables (scaled to have zero mean and unit variance) in Table 3 allow for better model convergence and comparability across features. Most variables show symmetrical scaling around the median (~0), with wider ranges observed in return-based features (e.g., Daily_Return_Scaled, Return_Lag_1_Scaled). The presence of NA values—particularly in moving averages and volatility—results from insufficient data in early periods, which is expected in rolling calculations. These scaled inputs serve as essential predictors for linear and deep learning models.

Furthermore, we examined the adjusted closing prices of each stock to understand their distribution and price ranges. Microsoft had the most stable prices, while Tesla exhibited the widest fluctuations, confirming its volatile nature. Time series plots highlighted these trends visually as shown in figure 1.

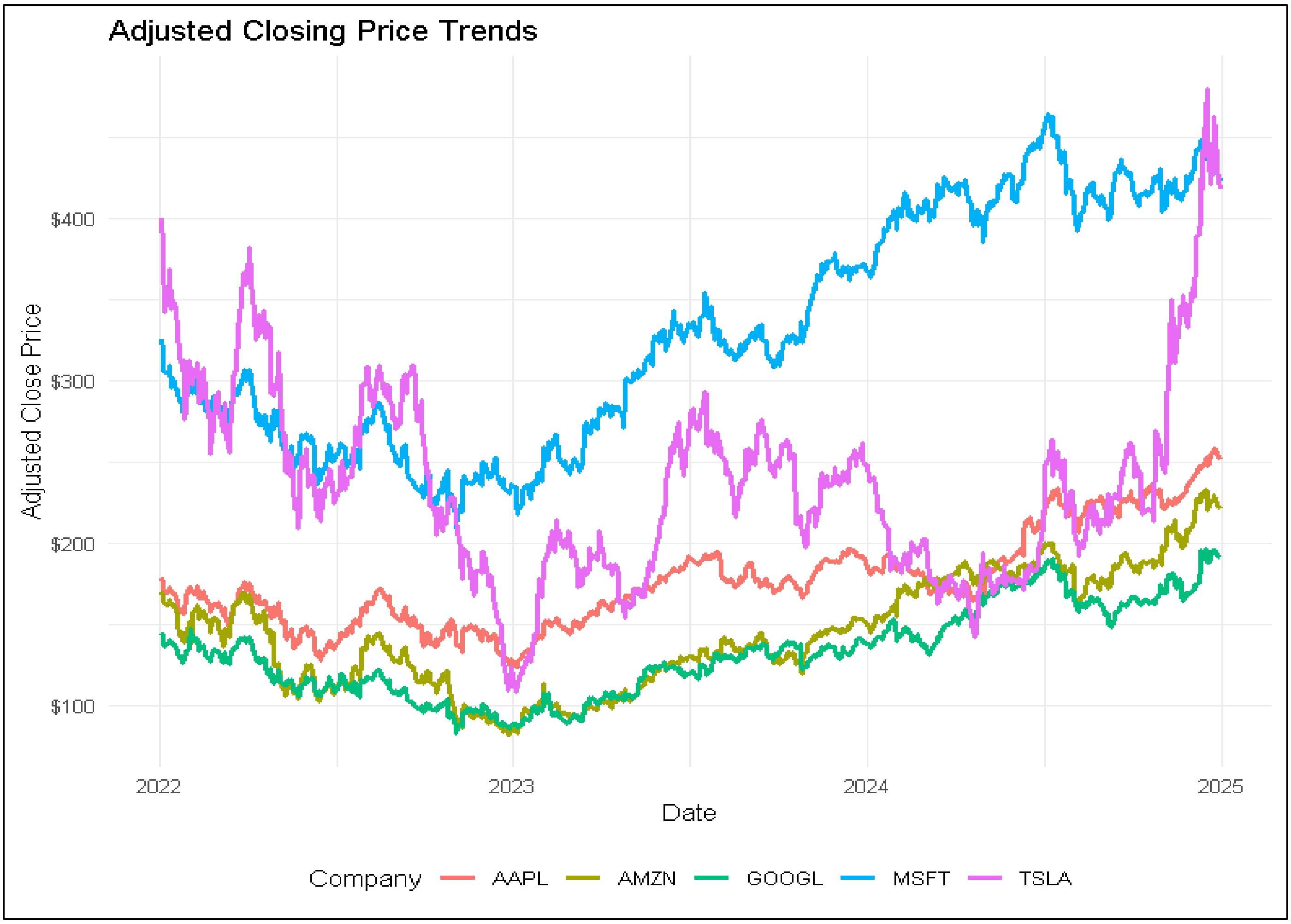

Fig 1 Shows the Adjusted Closing Price Trends

➢ *Volatility and Correlation*

To assess risk levels, we computed 20-day rolling volatility. Tesla stood out again with sharp spikes, indicating inconsistent price behavior as shown in figure 2.

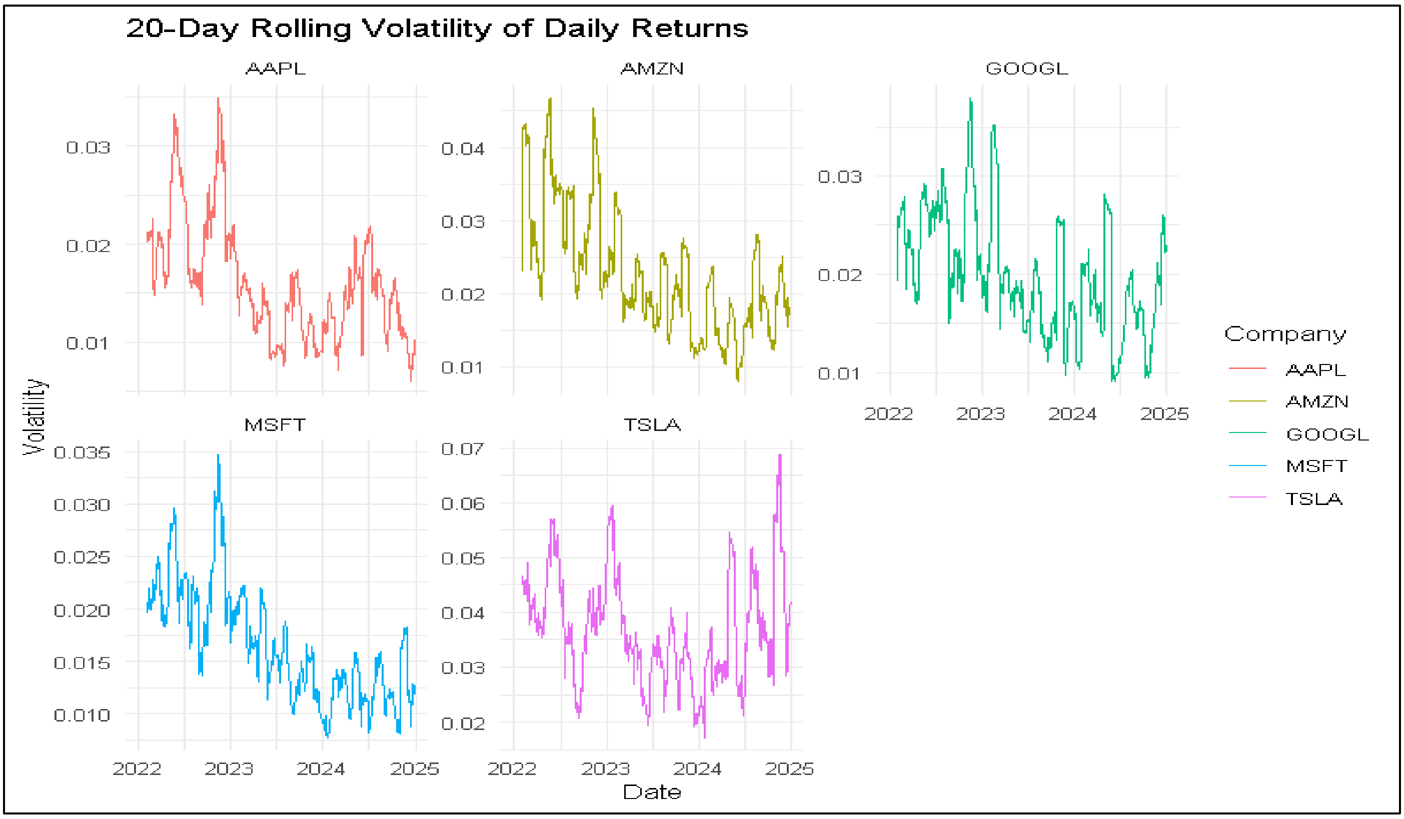

Fig 2 Shows the 20-Day Rolling Volatility of Daily Returns

Additionally, daily return correlations revealed that Microsoft and Google exhibited the strongest positive correlation, while Tesla showed weaker correlation with other stocks, reaffirming its unique market behavior as in figure 3.

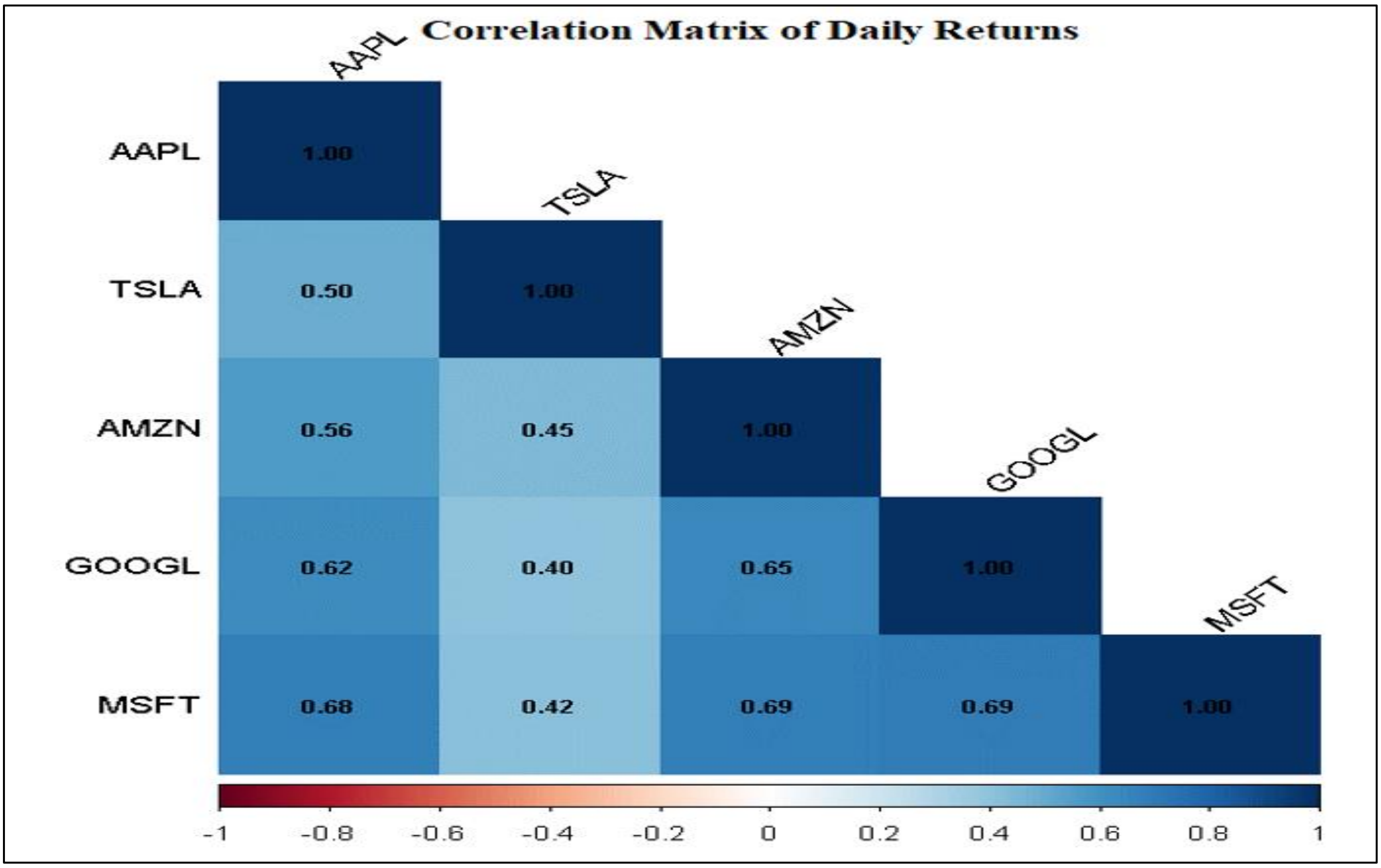

Fig 3 Shows the Correlation Matrix of Daily Returns

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

➢ *Moving Averages*

We applied 20-day and 50-day moving averages to detect underlying price trends and short- vs. long-term momentum. These smoothed curves provided a clearer indication of trend shifts across all companies as seen in figure 4.

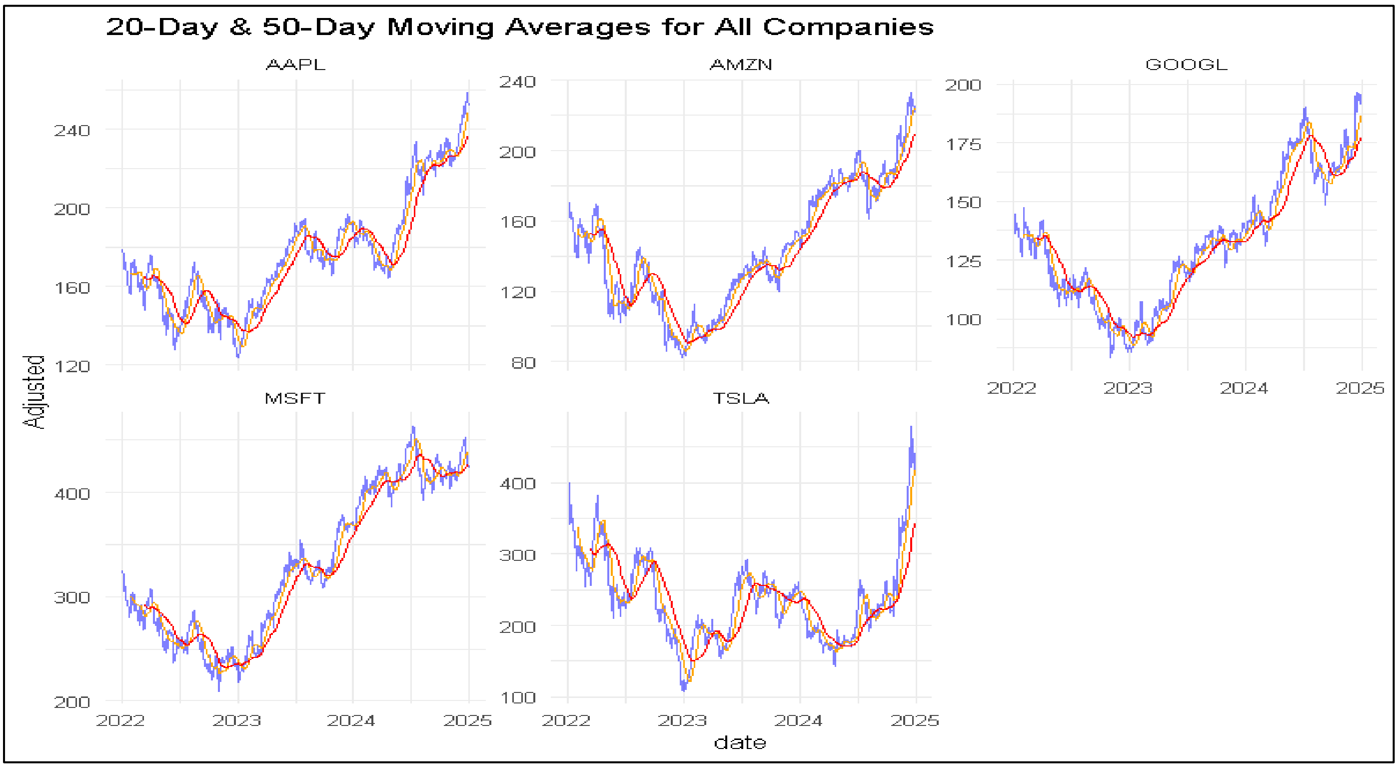

Fig 4 Shows the 20-Day and 50-Day Moving Averages for All Companies

➢ *Feature Engineering and Scaling*

Key predictive features were engineered, including lagged prices and returns, moving averages, and rolling volatility. All features were standardized to ensure consistency across inputs for later modeling as shown in figure 5.

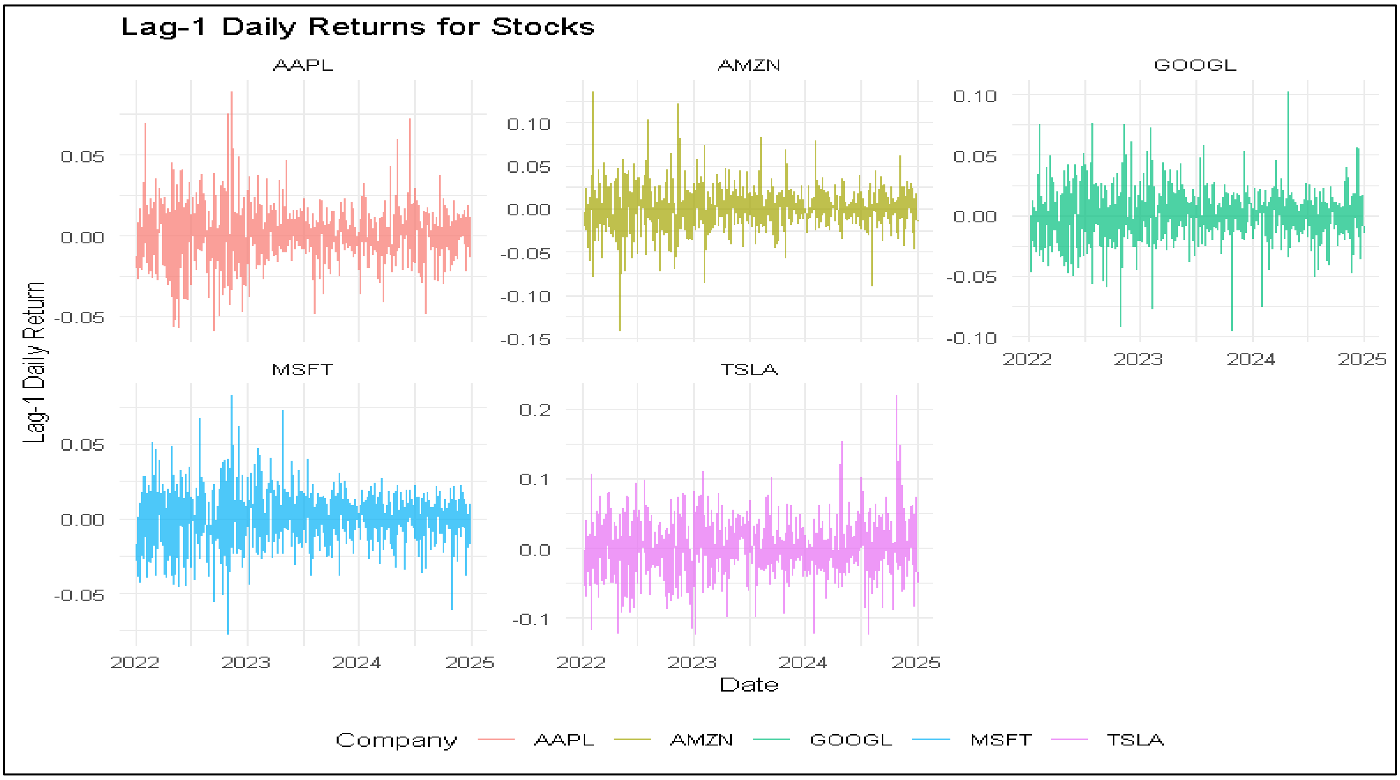

Fig 5 Shows the Lag-1 Daily Returns for Stocks

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

B. Linear Regression Baseline

As a benchmark, we trained a multiple linear regression model for each stock using 80% of the data and predicted the scaled adjusted close price on the remaining 20%. The features included lagged prices, past returns, moving averages, and volatility.

➤ *Apple (AAPL):*

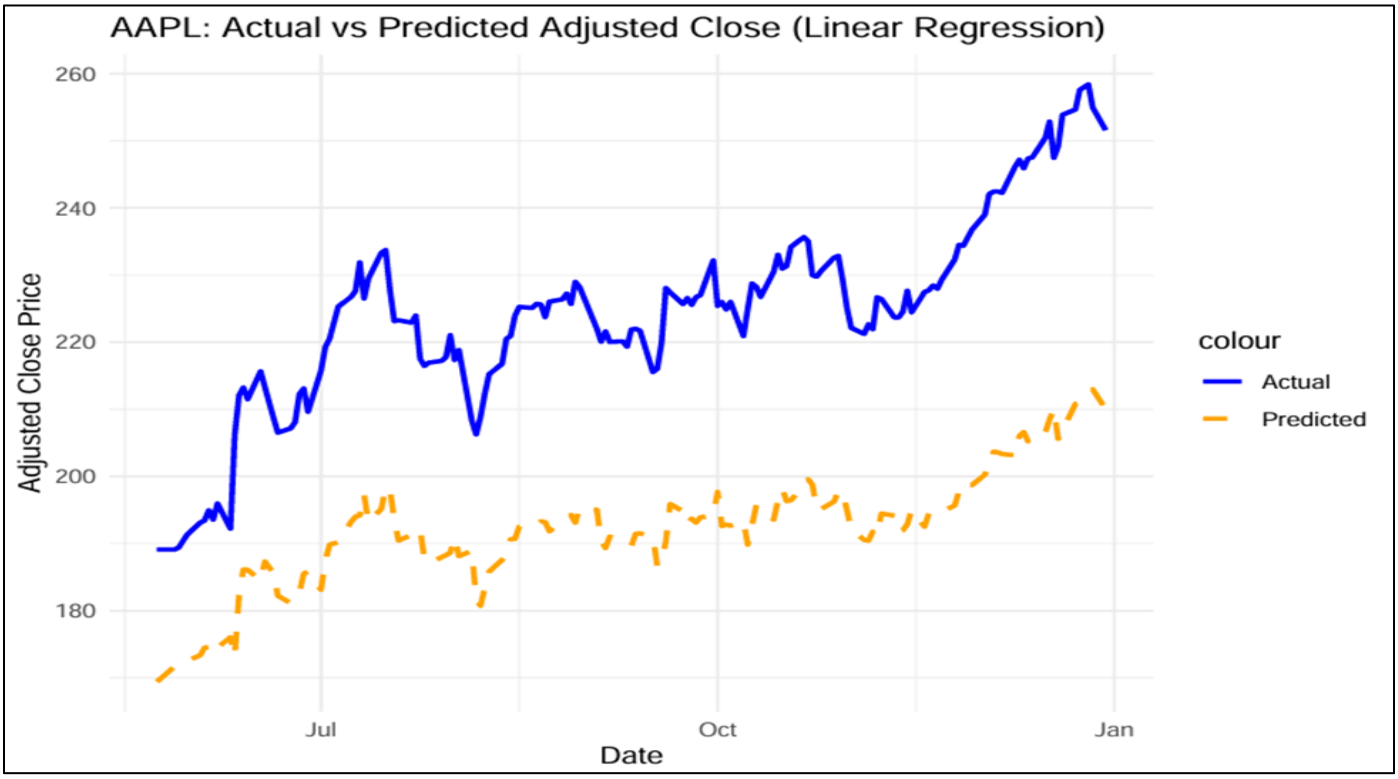

Fig 6 Shows the Actual vs Predicted Linear Regression Plot for Apple

Figure 6 shows that the linear regression model demonstrates moderate performance, capturing the overall upward trend in Apple's stock. However, noticeable prediction errors emerge during periods of increased volatility, where the model fails to adapt quickly to price fluctuations.

➤ *Tesla (TSLA):*

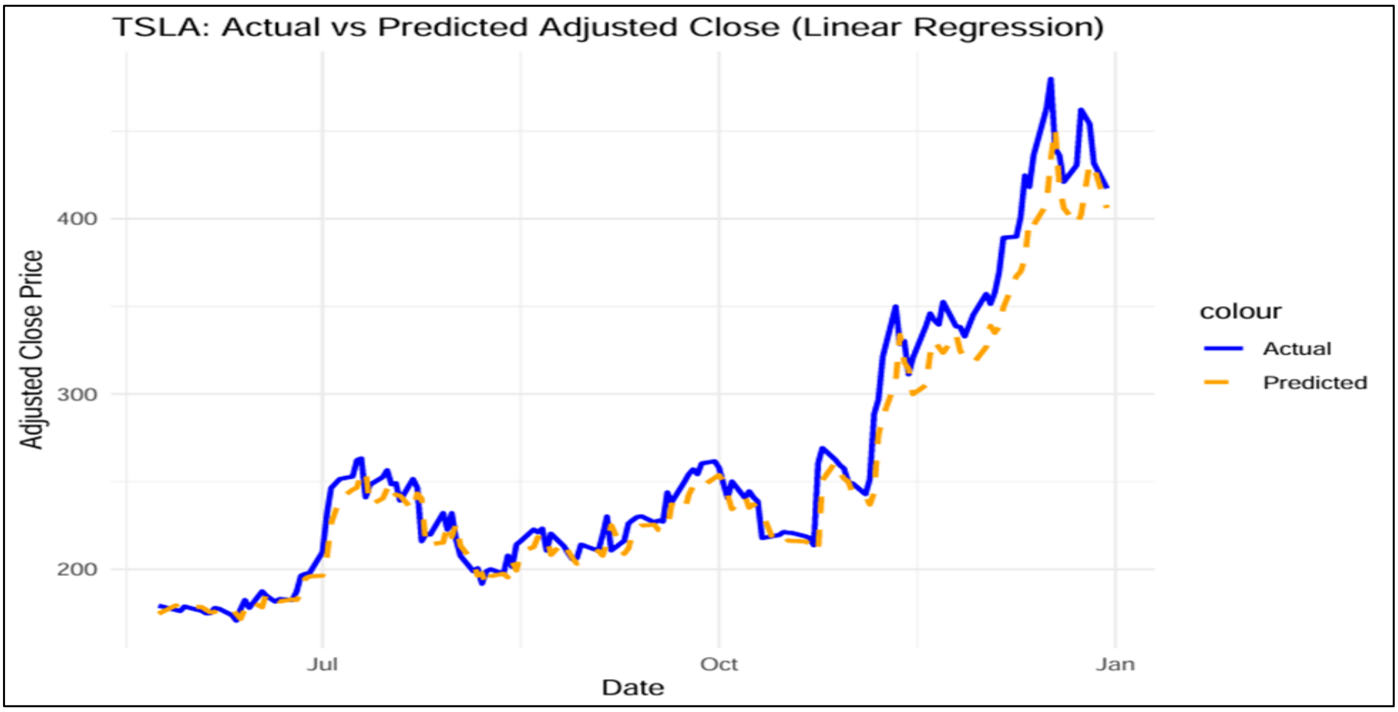

Fig 7 Shows the Actual vs Predicted Linear Regression Plot for Tesla

Figure 7 shows that the Tesla's highly volatile stock price proved challenging for the linear model. Although the general direction of the trend is partially mirrored, the model underperforms during sharp rises or drops, indicating a poor fit for non-linear patterns characteristic of TSLA.

➢ *Amazon (AMZN):*

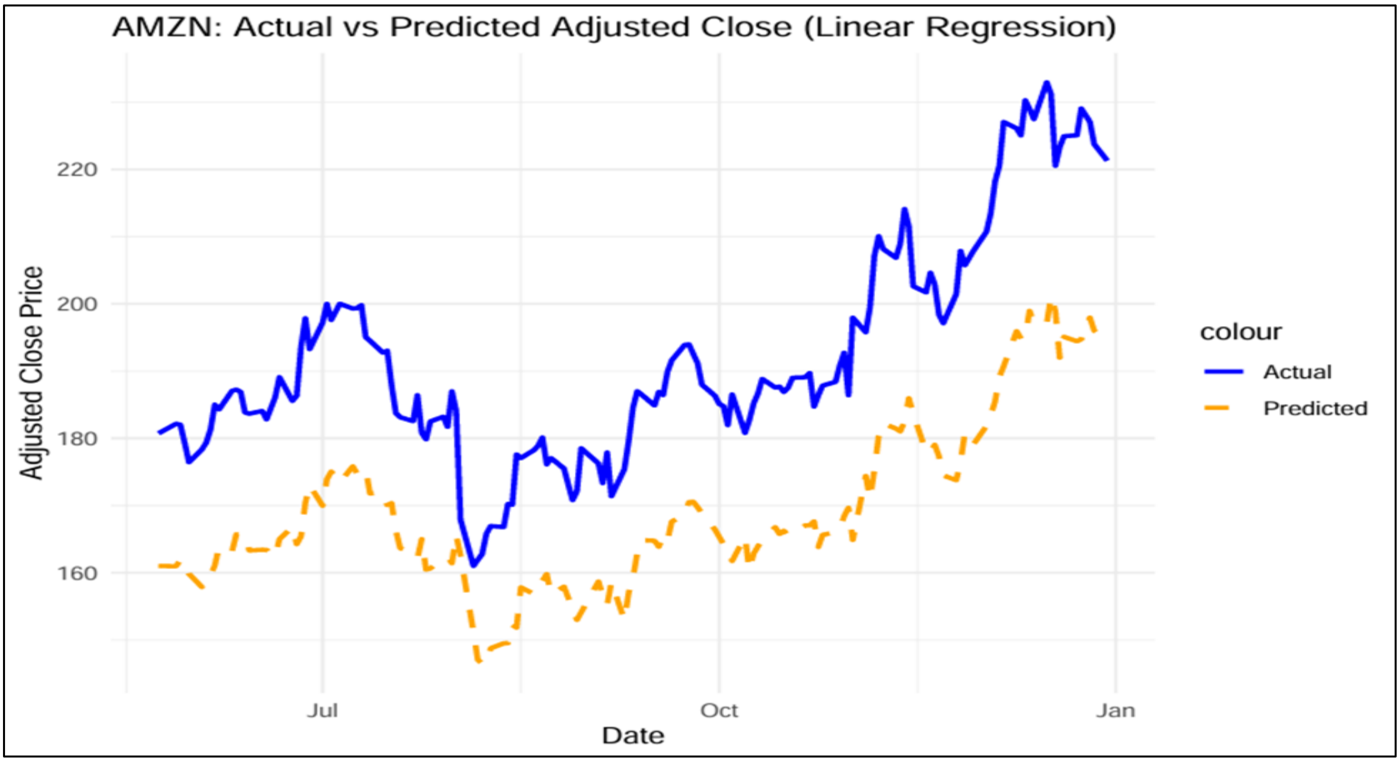

Fig 8 Shows the Actual vs Predicted Linear Regression Plot for Amazon

Figure 8 shows that Linear regression performs better with Amazon's stock, as the predicted values align relatively closely with the actual prices. The lower volatility compared to Tesla supports the model's ability to track price trends more consistently.

➢ *Google (GOOGL):*

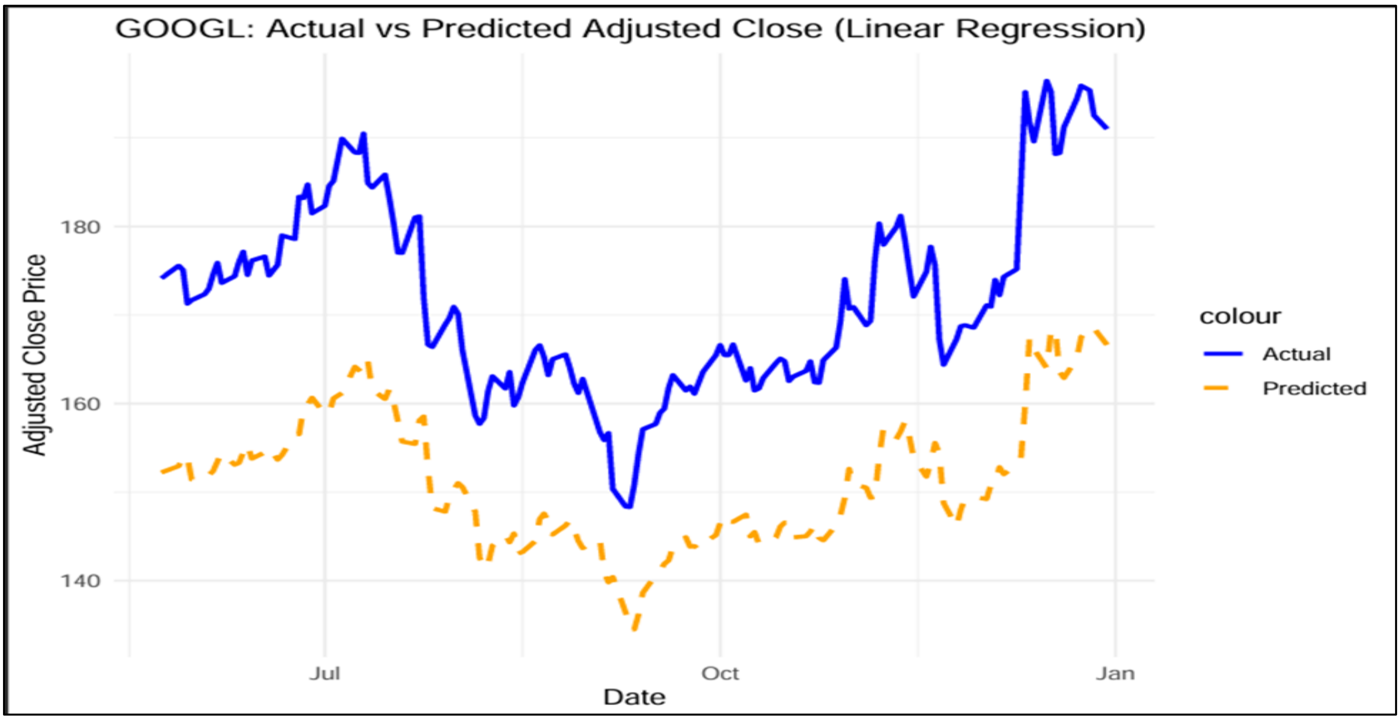

Fig 9 Shows the Actual vs Predicted Linear Regression Plot for Amazon

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

Figure 9 shows that Google's stock exhibits stable growth, and the linear regression model reflects this with reasonable accuracy. While not perfect, the predictions follow the general price direction, making it one of the more successful applications of the linear model.

➢ *Microsoft (MSFT):*

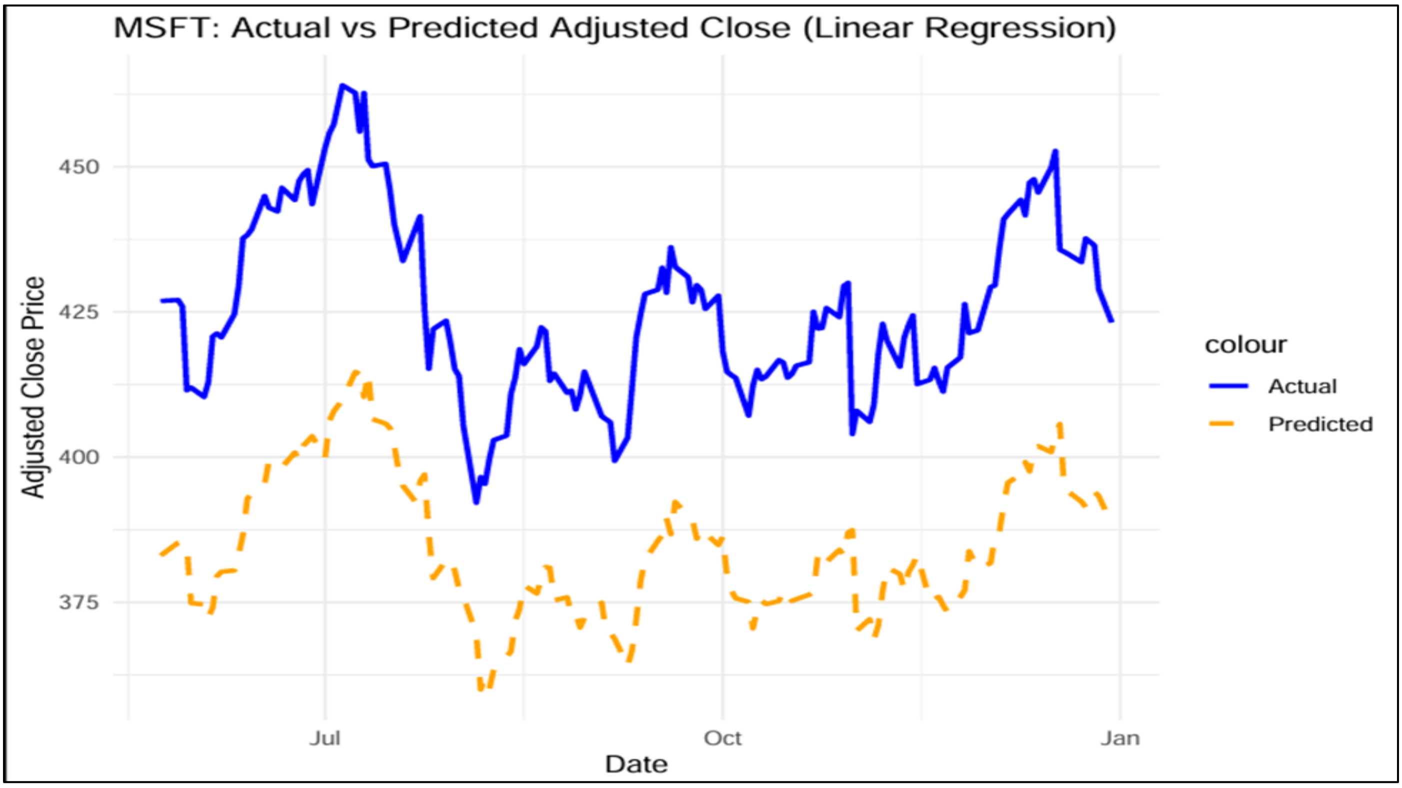

Fig 10 Shows the Actual vs Predicted Linear Regression Plot for Amazon

Figure 10 shows that Microsoft's consistent price behavior makes it ideal for linear modeling. Among all five companies, MSFT shows the best fit, with predicted values closely aligning with actual prices throughout the test period.

C. Overall Assessment

The linear regression model performs adequately for stocks with smoother and less volatile trends, such as MSFT, GOOGL, and AMZN. However, it struggles with highly dynamic stocks like TSLA, underscoring the limitations of linear methods in capturing complex, non-linear market behavior. These findings justify the need for more robust models that can learn temporal patterns and adapt to volatility, which led us to explore deep learning-based time series forecasting.

Building on the insights from the linear regression pre-analysis we did earlier, we extended our dataset for each company to enhance the effectiveness of deep learning models. This section evaluates three state-of-the-art architectures trained individually on stock data from Apple (AAPL), Tesla (TSLA), Amazon (AMZN), Google (GOOGL), and Microsoft (MSFT). These architectures include Long Short-Term Memory (LSTM), Fixed Temporal Convolutional Network (TCN), and the Simple Informer model. Each model was trained using a 70/15/15 split for training, validation, and testing, respectively. The core aim was not to predict exact prices but to predict the directional movements, which is a more robust and actionable approach for traders and investors. Through extensive feature engineering and data preprocessing, including normalization, lag features, and technical indicators, we enabled each model to learn complex patterns within the data. The models were then evaluated on classification accuracy across the defined targets. The table below presents their comparative performance and insights drawn from their results. The table below uses some metrics, such as average accuracy, best accuracy, average MAE, best MAE, average MSE, best MSE, average MAPE, best MAPE, etc., to compare the five companies selected in our study.

Table 4 Overall Performance of the Models

Model	Avg_ Accuracy	Best_ Accuracy	Avg_ MAE	Best_ MAE	Avg_ MSE	Best_ MSE	Avg_ MAPE	Best_ MAPE
LSTM	0.592	0.632	0.480	0.471	0.242	0.234	237	221
TCN	0.592	0.632	0.482	0.464	0.240	0.232	246	232
Simple Informer	0.592	0.632	0.479	0.466	0.241	0.235	233	217

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

Table 4 compares the performance of three deep learning models—LSTM, TCN (Temporal Convolutional Network), and Simple Informer —across several evaluation metrics. The comparison is based on how well each model predicts stock price direction using historical data.

The metrics evaluated are accuracy, which measures how often the model correctly predicts the direction of the stock movement; MAE (Mean Absolute Error), which is the average absolute difference between predicted probabilities and actual outcomes (lower = better); MSE (Mean Squared Error), which is the average squared difference between predictions and actuals (heavily penalises large errors; lower = better); and MAPE (Mean Absolute Percentage Error), which expresses prediction accuracy as a percentage error (lower = better).

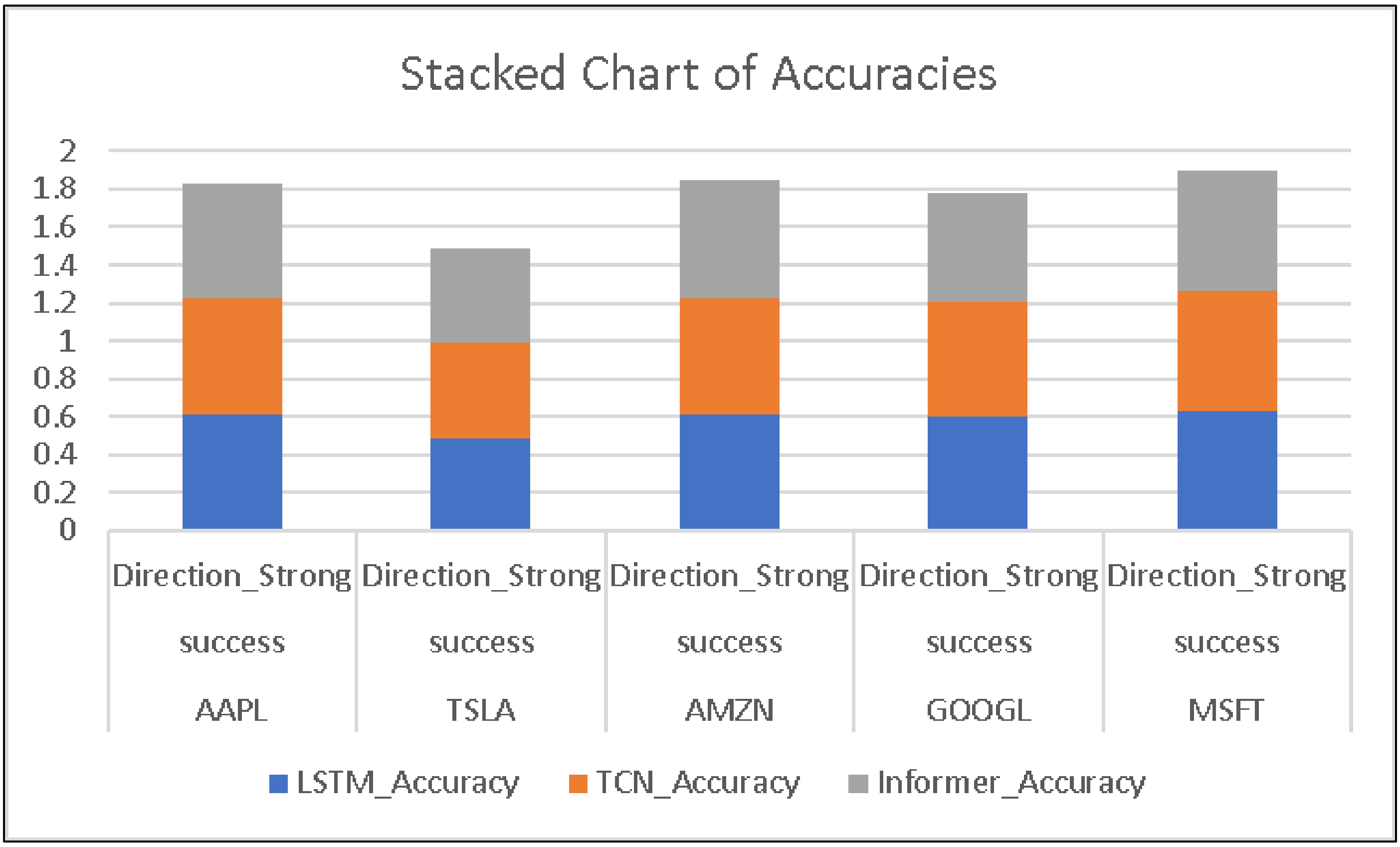

Fig 11 Shows the Visualization of their Respective Accuracy

D. Key Findings and Model Comparison

The deep learning models showed varying levels of success in predicting stock price directions across the five selected companies. Notable findings from the model evaluations are as follows: Microsoft (MSFT) yielded the highest predictive accuracy (approximately 63%), highlighting its relative stability and making it the most predictable stock among the five. While Tesla (TSLA) achieved only 49–50% accuracy - barely above random guessing, it reinforces its reputation for high volatility and market sensitivity, which poses significant challenges for prediction. For Amazon (AMZN) and Apple (AAPL), each company reached accuracies of approximately 61%, suggesting moderately stable and learnable price patterns, and Google (GOOGL) showed 60% accuracy for both LSTM and TCN models but saw a slight decline to approximately 58% with the Informer model.

On average, all the models managed to achieve an accuracy of about 59.2%, which means they were able to correctly predict the direction just a bit more than half the time. Interestingly, the best performance for any individual company reached 63.2%, suggesting that the predictability varied slightly depending on the company. Among the models, the Simple Informer stood out with the lowest average MAE of 0.479, making its probability estimates a little more accurate than those from the LSTM with 0.480 and the TCN with 0.482. When it came to minimizing large errors, TCN led the way with the lowest average MSE at 0.240. Meanwhile, the Simple Informer also posted the lowest average MAPE, which was approximately 233 million percent, indicating that its predictions were slightly closer to the actual values in relative terms. It is also worth noting that all the models were able to successfully train and make predictions for 5 of the companies tested. While all three models perform similarly, Simple Informer shows marginally better performance in terms of MAE and MAPE, while TCN performs best in MSE. However, the small differences suggest that model choice may depend on the specific use case or dataset characteristics. Further hyperparameter tuning and feature engineering could enhance performance. This comparison helps guide decisions on which model might be most suitable for future stock direction prediction tasks under similar conditions.

IV. CONCLUSION

This study successfully investigated the effectiveness of deep learning models for predicting stock price movements across five major technology companies, comparing Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN), and Simple Informer architectures against a baseline linear regression approach. The project achieved core objectives by developing a robust time-series forecasting pipeline using historical stock data enriched with extensive feature engineering. The models demonstrated the ability to provide investment-relevant insights, particularly identifying Microsoft as the most predictable stock (63% accuracy) and Tesla as the most challenging (49-50% accuracy).

The analysis revealed several important insights about the current state of machine learning-based stock prediction. Model performance limitations were identified, with modest accuracies ranging from 49% to 63% for predicting whether stock prices would increase by more than 1.5% within five days. Stock-specific predictability and market insights were also highlighted, with Microsoft emerging as the most predictable (63% accuracy) due to its relatively stable price movements, while Tesla proved most challenging to forecast (49-50% accuracy).

The shift from absolute price prediction to directional movement forecasting proved particularly effective in managing the complexity of financial time series data. The findings align with the project's initial premise that AI models should be viewed as tools for support rather than guarantees. The modest accuracies (49-63%) demonstrate that while deep learning can provide valuable insights beyond traditional methods, the inherent unpredictability of financial markets, particularly for sentiment-driven stocks like Tesla, remains a fundamental limitation. The choice of deep learning architecture showed minimal impact on prediction accuracy, with LSTM, TCN, and Informer models producing remarkably similar results, with mean accuracies differing by less than 1%. This finding suggests that in stock prediction tasks, model architecture may be less critical than data quality, feature engineering, and market understanding, directly supporting the objective to identify optimal approaches for different market conditions. Methodological validation was also successful, with the initial linear regression analysis justifying the shift from absolute price prediction to directional movement forecasting, effectively addressing the temporal dependencies and noise challenges identified in the introduction. This work reinforces the importance of maintaining realistic expectations about machine learning's capabilities in financial markets while providing a solid foundation for continued research in this challenging but important domain.

➢ *Conflicts of Interest*

The authors have no conflicts of interest.

REFERENCES

[1]. Ogundunmade TP, Adepoju AA, Allam A. Stock Price Forecasting: Machine Learning Models with K-fold and Repeated Cross Validation Approaches. *Mod Econ Manag*, 2022; 1: 1. DOI: 10.53964/mem.2022001.

[2]. Kaur, P., Marwaha, K., & Kumar, K. (2024). *A novel approach to predict the stock price using LSTM and linear regression*. IJSRSET, 24(11), 1–11. Retrieved from https://res.ijsrset.com/page.php?param=IJSRSET2411111

[3]. Liang, J., Chen, X., & Wang, Y. (2023). Comparison of price prediction based on LSTM, GRU, Random Forest, LSSVM and linear regression. *BCPBM, 38*(3713). https://doi.org/10.54691/bcpbm.v38i.3713

[4]. Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep l4arning. *Applied System Innovation*, *4*(1), 9.

[5]. Yin, C., Zhang, L., & Liu, H. (2023). Stock price prediction of GM Company: Comparison based on KNN, linear regression and LSTM. In *Advances in Economics, Management and Political Sciences* (pp. 82–89). EWADirect. https://www.ewadirect.com/proceedings/aemps/article/view/8253

[6]. Zheng, Z. (2023). A review of stock price prediction based on LSTM and TCN methods. *Advances in Economics, Management and Political Sciences, 46*, 48–54. https://doi.org/10.54254/2754-1169/46/20230316

[7]. Crow, J. (2023). *Stock Market Dataset* [Data set]. Kaggle. https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset

[8]. Li, X., Zheng, Z., & Li, Y. (2020). Deep reinforcement learning for optimal investment portfolio. Complexity, 2020.

[9]. Bukhari, A. H., Raja, M. A. Z., Sulaiman, M., Islam, S., Shoaib, M., & Kumam, P. (2020). Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting. *Ieee Access*, *8*, 71326-71338.

[10]. Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2022). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, 1-24.

[11]. Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, *197*, 116659.

[12]. Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science*, *10*(1), 183-208.

[13]. Sharma, D. K., Hota, H. S., Brown, K., & Handa, R. (2022). Integration of genetic algorithm with artificial neural network for stock market forecasting. *International Journal of System Assurance Engineering and Management*, *13*(Suppl 2), 828-841.

[14]. Thakkar, A., & Chaudhari, K. (2020). Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks. *Applied Soft Computing*, *96*, 106684.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25oct1245

[15]. Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science*, *167*, 2091-2100.

[16]. Zakhidov, G. (2024). Economic indicators: tools for analyzing market trends and predicting future performance. *International Multidisciplinary Journal of Universal Scientific Prospectives*, *2*(3), 23-29.

[17]. Ogundunmade, T.P., Adepoju, A.A. (2023). Predicting the Nature of Terrorist Attacks in Nigeria Using Bayesian Neural Network Model. In: Awe, O.O., Vance, E.A. (eds) Sustainable Statistical and Data Science Methods and Practices. STEAM-H: Science, Technology, Engineering, Agriculture, Mathematics & Health. Springer, Cham. https://doi.org/10.1007/978-3-031-41352-0_14

[18]. Ogundunmade TP, Abidoye M, Olunfunbi OM. Modelling Residential Housing Rent Price Using Machine Learning Models. Modern Economy and Management. 2023; 2: 14. doi: 10.53964/mem.2023014.

[19]. Ogundunmade TP, Daniel AO, M. Awwal A. Modelling Infant Mortality Rate using Time Series Models. International Journal of Data Science. 2023; 4(2): 107-115. doi: 10.18517/ijods.4.2.107-115.2023.

[20]. Ogundunmade TP, Ganiyu KA, Yahaya OT. Assessment of profitability and inventory management in the Nigerian power generation asset companies. Financial Statistical Journal. 2025; 8(1): 11398. https://doi.org/10.24294/fsj11398

[21]. Afolabi O. Adedamola; Tayo P. Ogundunmade (2025). Predictive Modelling of Crime Data using Machine Learning Models: A Case Study of Oyo State, Nigeria. International Journal of Innovative Science and Research Technology, 10(4), 1669-1677. https://doi.org/10.38124/ijisrt/25apr851.