Performance Comparison of Deep Learning Models in Predicting Stock Prices

Dr. Divya Pandey[1], Yasir Sarfaraz[2], Vishesh Goel[2], Yogesh Choudhary[2], Kartikeya Bhatnagar[2]

[1]SCSET, Bennett University, Greater Noida, Uttar Pradesh, India
[2]Student, Bennett University, Greater Noida, Uttar Pradesh, India

Abstract-The stock market in India is highly volatile, dynamic, and nonlinear, presenting significant challenges for researchers in time series forecasting. Predicting stock prices accurately has remained a critical topic of interest for both financial analysts and technical researchers. In this study, we deploy four deep learning models—LSTM, GRU, Simple RNN, and CNN1D—to predict the one-day-ahead closing prices of 20 prominent Indian companies grouped into five sectors: Communication Services, Consumer Cyclical, Energy, Healthcare, and Technology. The models were evaluated using five metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Normalized RMSE (NRMSE), and Coefficient of Determination (R^2). Our experimental results provide valuable insights into the comparative performance of these models in predicting stock prices, contributing to a deeper understanding of their applicability in Indian financial markets.

Index Terms—**stock price prediction, deep learning, LSTM, GRU, Simple RNN, CNN1D.**

I. Introduction

THe stock market plays a crucial role in the Indian economy today. Accurately predicting stock prices can help mitigate investment risks and maximize returns, making it a highly sought-after area of research for both financial and technical experts. The Indian stock market, like its global counterparts, is volatile, dynamic, and nonlinear. These characteristics make stock price forecasting a challenging task, influenced by a multitude of factors such as the domestic economy, political developments, corporate performance, and market sentiment.

Stock market prediction typically involves two primary approaches. The first is qualitative (or fundamental) analysis, which examines intrinsic factors such as financial performance, market conditions, management efficiency, and information from sources like economic analysts and social media. The second approach is technical analysis, which relies on historical market data, including opening and closing prices, volume, and trends. While fundamental analysis provides insights for long-term investment, technical analysis is often driven by short-term market movements. Within technical analysis, classical statistical models like ARIMA (Autoregressive Integrated Moving Average) have been widely used for time series forecasting. However, the advent of machine learning has revolutionized stock price prediction. Machine learning models, and particularly deep learning, excel at uncovering complex patterns in large datasets, enabling more accurate predictions.

In this study, we leverage deep learning models—LSTM, GRU, Simple RNN, and CNN1D—to predict the closing prices of 20 Indian companies grouped into five sectors: Communication Services, Consumer Cyclical, Energy, Healthcare, and Technology. Our contributions are as follows:

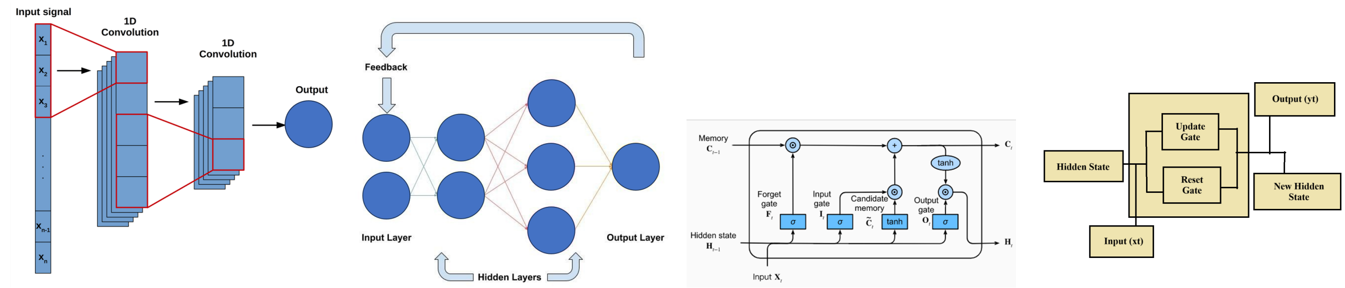
- We conduct experiments to evaluate the performance of four deep learning models.
- We analyze the stock price data of 20 Indian companies across five major sectors.
- We provide key insights into the suitability of deep learning models for stock price forecasting in Indian financial markets.

The rest of the paper is organized as follows: Section II reviews related work, Section III introduces the deep learning models, Section IV outlines the data and evaluation metrics, Section V presents the experimental results, and Section VI concludes the study.

II. Related Work

Stock price prediction in the Indian financial market has gained significant attention, with deep learning methods increasingly being applied to address the complexities of stock movements. Traditional methods, such as ARIMA and linear regression, often fail to capture the non-linearities in the market, leading to the growing adoption of advanced machine learning models, particularly deep learning techniques.

Mehtab and Sen (2020) used CNN and LSTM-based models to predict the NIFTY 50 index, finding that deep learning models outperformed traditional methods with a walk-forward validation approach. Similarly, Shachmurove and Witkowska (2000) demonstrated that Multilayer Perceptron (MLP) networks performed better than traditional forecasting models in predicting stock market returns, highlighting the power of artificial neural networks (ANNs) in capturing complex market dynamics. Vargas et al. (2020) combined CNN and RNNs for predicting stock price movements by analyzing financial news and stock data, concluding that CNNs are better for semantic extraction, while RNNs are more effective at modeling temporal dynamics. Choudhary et al. (2024) further extended this by incorporating news headlines into stock prediction models, using a stacked LSTM model trained on a combined dataset of stock data and news. This approach



(a) CNN 1D (b) Simple RNN (c) LSTM (d) GRU

Fig. 1: Architectures of the Models Used

enhanced prediction accuracy by integrating sentiment analysis with technical indicators. Baek and Kim (2018) proposed ModAugNet, a framework that uses LSTM autoencoders and stacked LSTMs to reduce overfitting, significantly improving stock prediction performance over traditional LSTMs.

Rajaa and Sahoo (2019) introduced a model combining CNN feature extraction and Neural Arithmetic Logic Units (NALUs), showing its potential in improving stock price forecasting. Al-Thelaya et al. (2019) demonstrated the effectiveness of LSTM autoencoders for stock market forecasting, which can be adapted for the Indian market. Similarly, Hiemstra (1996) found that non-linear backpropagation networks outperformed linear models in predicting quarterly stock returns, supporting the growing shift towards deep learning-based methods.

These studies underscore the effectiveness of deep learning models, especially hybrid models incorporating both financial data and external sources like news, in improving stock prediction accuracy in the Indian stock market.

III. Models Used

Here we present 4 deep learning models which are used to conduct the experiment results in Section V.

A. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) have revolutionized the field of image processing and classification, but their architecture can also be adapted for sequential data tasks, such as time-series forecasting. 1D CNNs (CNN1D) are specifically designed for this purpose, where the input data is typically a sequence or a time-series, rather than a 2D image. In CNN1D, convolutional filters are applied across the one-dimensional sequence of data, allowing the network to automatically learn spatial hierarchies of features from the sequence, which is particularly beneficial for time-series forecasting where data points are related to one another in a temporal order.

The key advantage of CNN1D lies in its ability to capture local dependencies across different time steps by sliding the convolutional filters over the sequence. This mechanism helps the model detect patterns and features in the data, such as trends, periodicities, and anomalies, without requiring complex recurrent connections like those in Recurrent Neural Networks (RNNs). While RNNs process the data sequentially, CNN1D can process the entire sequence in parallel, making it computationally more efficient. Moreover, CNN1D can automatically learn feature representations from raw data, often outperforming traditional statistical models in extracting meaningful patterns and improving predictive performance.

CNN1D's effectiveness in time-series forecasting tasks has been demonstrated in various applications, including stock price prediction, weather forecasting, and energy demand forecasting, where capturing temporal patterns is crucial for accurate predictions.

B. Recurrent Neural Network (RNN)

Simple Recurrent Networks (RNNs) are a type of neural network designed to handle sequential data by maintaining a hidden state that updates over time as new inputs are processed. This hidden state serves as a memory, allowing the network to capture temporal dependencies and store information about previous time steps. In an RNN, the output at each time step depends not only on the current input but also on the hidden state from the previous step, making it well-suited for modeling time-series data where past events influence future outcomes.

RNNs face significant challenges when dealing with long sequences. During backpropagation, the vanishing gradient problem occurs when gradients diminish exponentially, making it hard for the network to learn long-range dependencies. Conversely, the exploding gradient problem can cause the gradients to grow uncontrollably, destabilizing the training process. These issues hinder the model's ability to capture long-term dependencies effectively. As a result, while simple RNNs can excel at short-term sequence learning, more advanced architectures like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Units) are often preferred for tasks requiring longer memory and more complex temporal relationships.

C. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks were specifically designed to address the limitations of traditional Recurrent Neural Networks (RNNs), particularly the challenges of learning long-term dependencies and the vanishing/exploding gradient problems. In a standard RNN, as the network processes longer sequences, it struggles to retain information from earlier time steps due to diminishing or escalating gradients during backpropagation. This issue makes it difficult for RNNs to capture relationships over extended sequences.

LSTMs overcome these limitations by introducing memory cells, which can store information for longer periods. These cells are regulated by three key components: input gate, forget gate, and output gate. The input gate controls the flow of new information into the memory cell, the forget gate decides which information to discard, and the output gate determines which data should be output from the memory cell to the next time step. This gating mechanism allows LSTMs to learn and retain long-term dependencies, making them highly effective for tasks such as time-series forecasting, speech recognition, and natural language processing, where the ability to remember information over long periods is crucial.

D. Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is a simplified and computationally efficient variant of the traditional Recurrent Neural Network (RNN), developed to overcome challenges like vanishing gradients and difficulties in learning long-term dependencies. Unlike Long Short-Term Memory (LSTM) networks, which use three gates (input, forget, and output gates) and a separate memory cell, GRUs have a more streamlined architecture with only two gates: the update gate and the reset gate. The update gate controls how much of the previous hidden state should be carried forward and how much should be updated with new input. The reset gate determines how much of the past information should be forgotten, allowing the network to focus on more relevant data.

This simpler structure eliminates the need for a memory cell, instead relying on the hidden state to store information. As a result, GRUs are faster to train, require fewer parameters, and are less computationally intensive compared to LSTMs. Despite their simplicity, GRUs can effectively capture complex dependencies in sequential data, making them suitable for a range of tasks such as time-series forecasting, speech recognition, and natural language processing. Their strong performance and efficiency make GRUs a popular choice in many applications.

IV. Data and Evaluation

A. Data

To evaluate the performance of deep learning models in predicting stock prices, we utilize datasets consisting of historical stock prices for 20 prominent Indian companies, covering five sectors: Communication Services, Consumer Cyclical, Energy, Healthcare, and Technology. These companies have been carefully selected based on their market influence and prominence in their respective sectors, as detailed in Table 1. Each sector contains four companies, to capture a wide range of market behaviors and characteristics across different industries.

1) Companies and Sectors

- **Communication Services:** The Communication Services sector includes companies such as Bharti Airtel (BHARTIARTL.NS), Idea (IDEA.NS), Reliance (RELIANCE.NS), and Tata Communications (TATACOMM.NS). These companies are major players in the telecommunications industry, offering mobile and broadband services across India.
- **Consumer Cyclical:** The Consumer Cyclical sector is represented by Eicher Motors (EICHERMOT.NS), Mahindra & Mahindra (M&M.NS), Maruti Suzuki (MARUTI.NS), and Tata Motors (TATAMOTORS.NS). These companies are key manufacturers of automobiles and two-wheelers, providing essential goods that reflect consumer spending patterns.
- **Energy:** The Energy sector includes BPCL (BPCL.NS), IOC (IOC.NS), ONGC (ONGC.NS), and Reliance (RELIANCE.NS). These companies are vital players in the oil, gas, and energy production industries, impacting both domestic and international energy markets.
- **Healthcare:** The Healthcare sector features Cipla (CIPLA.NS), Divi's Labs (DIVISLAB.NS), Dr. Reddy's Laboratories (DRREDDY.NS), and Sun Pharma (SUNPHARMA.NS). These companies are critical in the pharmaceutical industry, focusing on the development, manufacturing, and distribution of drugs and healthcare products.
- **Technology:** The Technology sector includes HCL Technologies (HCLTECH.NS), Infosys (INFY.NS), Tata Consultancy Services (TCS.NS), and Wipro (WIPRO.NS). These companies are among the largest IT services providers globally, delivering software, consulting, and technology solutions to a wide array of industries.

2) Data Characteristics

The historical stock price data used in this study spans several years, providing a comprehensive view of each company's performance over time. The data for each stock contains six key features:

- Date,
- Open Price,
- High Price,
- Low Price,
- Close Price,
- Volume.

Among these, the **Close Price** is chosen as the target variable for prediction, as it is commonly used in financial analysis and represents the final price of the stock at the end of each trading day.

3) Preprocessing and Data Preparation

Before applying deep learning models, several data preprocessing steps were carried out to ensure the dataset's accuracy and consistency. These included:

- **Handling Missing Values:** Any missing data points were carefully handled, either through interpolation or removal, depending on the extent of the missing values.
- **Normalization:** The stock prices can vary significantly across companies and sectors, the data was normalized to ensure that all values lie within a similar range, which helps improve the performance and convergence speed of machine learning models.

TABLE 1
List of twenty stocks divided into five groups. The symbol of stocks and their market cap in INR are shown. T stands for trillion and B denotes billion.

Communication Services		Consumer Cyclical		Energy		Healthcare		Technology	
Stock Symbol	Market Cap	Stock Symbol	Market Cap	Stock Symbol	Market Cap	Stock Symbol	Market Cap	Stock Symbol	Market Cap
BHARTIARTL	0.5T	TML	0.3T	RELIANCE	1.5T	SUNPHARMA	0.2T	INFY	0.6T
RELIANCE	1.5T	M&M	0.4T	ONGC	0.7T	DRREDDY	0.3T	TCS	1.0T
IDEA	0.1B	MARUTI	0.5T	IOC	0.8T	CIPLA	0.4T	WIPRO	0.7T
TATACOMM	0.2B	EICHERMOT	0.6T	BPCL	0.9T	DIVISLAB	0.5T	HCLTECH	0.8T

B. Evaluation

The performance of the models was assessed using five statistical indices:

Mean Absolute Error (MAE): Measures the average magnitude of errors between the predicted and actual values. Root Mean Squared Error (RMSE): Calculates the square root of the average squared differences between the predicted and actual values. Mean Absolute Percentage Error (MAPE): Expresses the prediction error as a percentage of the actual values. Normalized Root Mean Squared Error (NRMSE): Normalizes the RMSE by dividing it by the range of observed values. Coefficient of Determination (R^2): Indicates the proportion of variance in the observed data that is predictable from the model. The formulas for these indices are as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{3}$$

$$\text{NRMSE} = \frac{\text{RMSE}}{\max(y) - \min(y)} \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5}$$

Where

- y_i is the actual value (true value).
- $\hat{y}_i$ is the predicted value.
- $\bar{y}$ is the mean of the actual values.
- n is the number of data points.
- $\max(y)$ and $\min(y)$ are the maximum and minimum values of the actual data.

This evaluation framework ensures a comprehensive comparison of model performance across different companies and sectors.

V. Experimental Results

In this study, we predicted the next day's closing price for each company using historical features, including Open, High, Low, Close, and Volume as inputs. A key challenge in forecasting stock prices is determining the appropriate look-back window size (lags). For simplicity and consistency across models, we used a fixed look-back window of 10 days for all experiments, as this duration often balances short- and long-term dependencies effectively.

We implemented four deep learning models—LSTM, GRU, Simple RNN, and CNN1D—on datasets from 20 companies across five sectors: Communication Services, Consumer Cyclical, Energy, Healthcare, and Technology. Each sector comprised four major companies, with historical stock data spanning a multi-year range.

Model	MAE	RMSE	MAPE	NRMSE	R^2
LSTM	19.34	25.31	1.94	2.12	0.99
GRU	17.70	22.66	1.75	1.89	0.99
SimpleRNN	46.86	56.05	4.25	4.68	0.96
CNN1D	28.63	36.61	2.78	3.06	0.98

TABLE 2
Performance results of 4 Different Models on Reliance stock

Table 2 presents the performance results between models for predicting Reliance communication stocks. GRU outperformed all other models, achieving the lowest MAE (17.70), RMSE (22.66), and MAPE (1.75%), indicating superior accuracy in predicting Reliance's stock prices. LSTM followed closely, showing robust performance with slightly higher errors. CNN1D demonstrated moderate accuracy, with errors substantially higher than GRU and LSTM. SimpleRNN performed the worst, with MAE of 46.86 and RMSE of 56.05, indicating significant difficulties in capturing the stock's complex price movements. The high R^2 values (0.99 for GRU and LSTM) suggest strong fits, while SimpleRNN's lower R^2 (0.96) reflects poor predictive reliability. GRU's strong results highlight its effectiveness for this sector.

Model	MAE	RMSE	MAPE	NRMSE	R^2
LSTM	37.97	49.62	2.75	3.44	0.98
GRU	16.50	22.59	1.36	1.57	0.99
SimpleRNN	27.67	38.27	1.95	2.65	0.99
CNN1D	37.95	47.60	2.94	3.30	0.99

TABLE 3
Performance results of 4 Different Models on Infosys stock

Looking at Table 2 and 3, GRU consistently outperforms the other deep learning models—LSTM, GRU, SimpleRNN, and CNN1D—across both sectors (Reliance in Communication Services and Infosys in Technology), achieving the lowest MAE, RMSE, and MAPE values. For Reliance, GRU's MAE is 17.70, and for Infosys, it further improves to 16.50, demonstrating its ability to adapt to different stock price dynamics. SimpleRNN, however, performs poorly in both sectors, with significantly high errors, such as an MAE of 46.86 for Reliance, indicating its inability to capture the complexity of stock price movements. LSTM shows moderate performance, with an MAE of 19.34 for Reliance but higher errors for Infosys at 37.97, suggesting it is less precise compared to GRU. CNN1D displays variable accuracy, with better performance for Reliance (28.63 MAE) but worse results for Infosys (37.95 MAE). The Communication Services sector, characterized by higher volatility, leads to higher error metrics for all models, whereas the Technology sector's smoother trends result in lower errors. All models, especially GRU, demonstrate high R^2 values (0.99), reflecting their strong ability to capture stock price patterns.

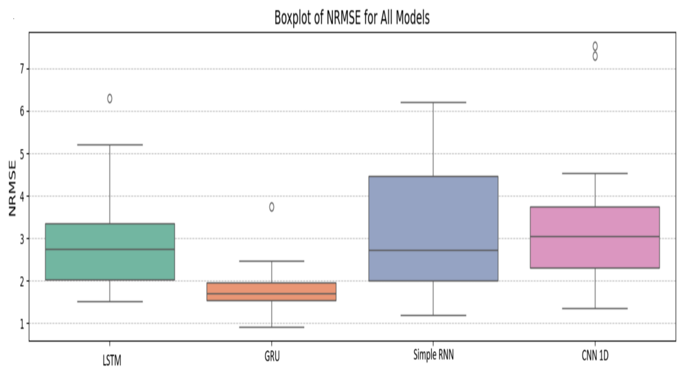


Fig. 2: Boxplot of NRMSE results across all sectors

The metrics MAPE, NRMSE, and R^2 play a pivotal role in evaluating the relative error between actual and predicted values, allowing for meaningful comparisons across models and sectors. These indices provide insights into forecasting performance, helping to identify the most effective models for stock price prediction.

From Figure 2, which presents the boxplot of NRMSE results across all sectors, the GRU model stands out with consistently lower median NRMSE values and narrower interquartile ranges compared to other models. This indicates that GRU achieves reduced forecast errors and maintains stability across diverse datasets. The LSTM model follows closely but exhibits slightly larger forecast errors and variability, while SimpleRNN often has the highest NRMSE and the widest spread, making it the least reliable among the four models.

The R^2 score, a measure of the goodness of fit, provides additional clarity on model performance. Figure 2 shows the R^2 scores of the models in each sector, indicating that the GRU achieves the highest R^2 scores across key sectors, demonstrating its robustness in predicting unseen samples with high accuracy. For instance, GRU outperforms other models in Communication Services ($R^2 = 0.9886$), Energy ($R^2 = 0.9930$), and Healthcare ($R^2 = 0.9945$), highlighting its ability to handle the complexities of these datasets effectively. In the Consumer Cyclical sector, GRU also leads ($R^2 = 0.9955$), but CNN1D emerges as a strong alternative, particularly excelling in cases requiring nuanced pattern recognition.

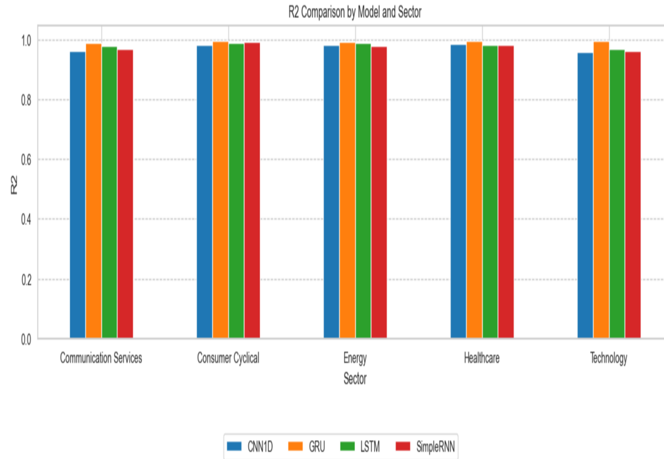


Fig. 3: R2 Score comparison of models by sectors

The Technology sector showcases GRU's efficiency, achieving the lowest MAPE (1.49%) and the highest R^2 (0.9941), while CNN1D remains a competitive option with reasonable forecast accuracy. SimpleRNN, by contrast, exhibits consistently higher errors and lower R^2 values, indicating its limited applicability in this domain.

When comparing models by sector:

- **GRU:**
  - Consistently delivers high accuracy and stability across all sectors.
  - Proven to be the most reliable model.
- **LSTM:**
  - Demonstrates robust performance across most sectors.
  - Ranks second or third in terms of accuracy.
  - Its ability to handle temporal dependencies makes it a reliable choice, especially in moderately variable sectors like Healthcare and Energy.
- **CNN1D:**
  - Performs exceptionally in Consumer Cyclical and Technology sectors.
  - Proves to be a strong competitor in specific cases.
- **SimpleRNN:**
  - Struggles to provide accurate predictions.
  - Exhibits higher forecast errors in sectors with high variability.

In the Technology sector, a comparative analysis of the actual vs. predicted performance of the four models reveals distinct patterns in their forecasting abilities. Among all models, GRU stands out with the best performance metrics, achieving the lowest MAE (18.98), RMSE (25.47), and MAPE (1.49%), along with the highest R^2 score of 0.9941. This indicates that GRU provides highly accurate predictions and effectively captures trends in the Technology sector's stock price movements. In the Technology sector, a comparative analysis of the actual vs. predicted performance of the four models reveals distinct patterns in their forecasting abilities. Among all models, GRU stands out with the best performance metrics, achieving the lowest MAE (18.98), RMSE (25.47), and MAPE (1.4%), along with the highest R^2 score of 0.9941. This indicates that GRU provides highly accurate predictions and effectively captures trends in the Technology sector's stock price movements. Its strong performance can be attributed to its ability to efficiently handle sequential data and learn complex temporal dependencies, making it ideal for forecasting stock prices.

On the other hand, CNN1D demonstrates robust but slightly less accurate performance, with an R^2 score of 0.9583. While its forecast errors are larger than those of GRU, CNN1D remains a strong contender, particularly in terms of its stability across sectors. CNN1D excels in sectors with relatively predictable stock movements, capturing patterns in the data efficiently, but still falls behind GRU in terms of overall accuracy.

Both LSTM and SimpleRNN show relatively higher forecast errors, with LSTM achieving an R^2 of 0.9671 and SimpleRNN lagging slightly behind at 0.9626. Notably, SimpleRNN exhibits the highest RMSE (60.79), indicating that it struggles to capture finer nuances in the data. LSTM, while showing more stability than SimpleRNN, still falls short of GRU's accuracy in the Technology sector.

GRU emerges as the most effective model for the Technology sector, offering precise and consistent predictions. CNN1D and LSTM provide reasonable alternatives, especially in less volatile sectors. SimpleRNN's performance, though less reliable, highlights its potential in simpler, less complex forecasting scenarios. Across all models, GRU's dominance is clear, but CNN1D and LSTM also provide valuable insights, making them suitable for specific use cases where higher accuracy is not the primary concern. This indicates that model choice should be based on both the complexity of the sector's data and the forecast accuracy requirements for optimal results.

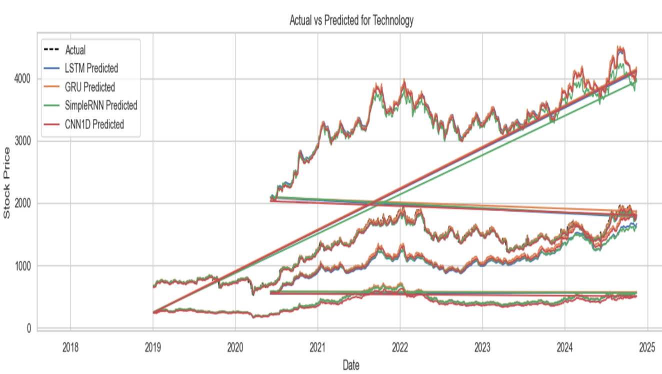


Fig. 4: Actual vs Predicted Plot for Technology Sector

VI. Conclusion

This comprehensive study presents a rigorous evaluation of deep learning models for stock price prediction in the Indian financial market, offering significant insights into the predictive capabilities of different neural network architectures. Our research systematically investigated the performance of LSTM, GRU, Simple RNN, and CNN1D models across five critical economic sectors: Communication Services, Consumer Cyclical, Energy, Healthcare, and Technology.

The experimental results reveal critical findings that substantially contribute to understanding deep learning's potential in stock market forecasting. Gated Recurrent Unit (GRU) emerged as the most consistent and accurate model across sectors, demonstrating superior performance in capturing complex temporal dependencies. With the lowest Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), GRU consistently outperformed other models, particularly in volatile sectors like Communication Services.

Our analysis highlighted the nuanced performance variations across different neural network architectures. While GRU demonstrated remarkable predictive accuracy, LSTM showed robust performance, ranking second in most sectors. CNN1D displayed exceptional capabilities in specific sectors like Consumer Cyclical and Technology, indicating its potential for pattern recognition. Conversely, Simple RNN consistently underperformed, struggling to capture intricate stock price movements.

The high Coefficient of Determination (R^2) values across models, particularly for GRU (approaching 0.99), underscore the effectiveness of deep learning techniques in stock price prediction. These findings validate the potential of advanced machine learning models in addressing the inherently complex and nonlinear nature of financial markets.

This research contributes significantly to the growing body of knowledge in financial forecasting, providing practitioners and researchers with empirical evidence on model selection strategies. Future research could explore hybrid models, incorporate additional features like sentiment analysis, and investigate these architectures' performance across diverse global markets.

References

[1] S. Mehtab and J. Sen, "Designing robust and accurate predictive models for stock price prediction," 2020.

[2] ICCES, "Applications of machine learning in stock market forecasting," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021.

[3] Z. Yan, Z. Huang, and M. Liang, "Stock prediction via linear regression and BP regression network," School of Science, Nanjing University of Science and Technology, 2019.

[4] Y. Shachmurove and D. Witkowska, "Utilizing artificial neural network model to predict stock markets," Department of Economics, City College, City University of New York & University of Pennsylvania, 2000.

[5] Y. Hiemstra, "Linear regression versus backpropagation networks to predict quarterly stock market excess returns," *Computational Economics*, 1996.

[6] Y. Baek and H. Y. Kim, "ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Systems with Applications*, 2018.

[7] M. R. Vargas, B. S. L. P. de Lima, and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," 2020.

[8] M. Choudhary, V. Bajpai, and A. Bansal, "Stock price prediction leveraging news headlines," in *2024 International Conference on Emerging Techniques in Computational Intelligence*, 2024.

[9] K. A. Al-Thelaya, E. M. El-Alfy, and S. Mohammed, "Forecasting of Bahrain stock market with deep learning: Methodology and case study," in *2019 8th International Conference on Modeling Simulation and Applied Optimization*, 2019.

[10] S. Rajaa and J. K. Sahoo, "Convolutional feature extraction and neural arithmetic logic units for stock prediction," in *Advances in Computing and Data Sciences (ICACDS 2019)*, 2019.