

TITLE

Pranav Bhavsar

11 BE IT/A

Thakur College of Engineering and

Technology, Mumbai

Mumbai, India

pranavbhavsar2003@gmail.com

Bharat Bohra

12 BE IT/A

Thakur College of Engineering and

Technology, Mumbai

Mumbai, India

bharatbohra07@gmail.com

Abstract - Predicting stock prices is a tough task that is affected by multiple factors in the market and past trends. This research analysis investigates deep learning techniques for predicting Nifty 50 stock price using historical data from 2014 to 2024. The dataset includes the OHLCV values along with some of the more common technical indicators (RSI, Moving Average, etc), which help to increase the accuracy of the prediction. Several deep learning architectures were then generated and evaluated, including LSTM, GRU, CNN-LSTM, LSTM with Attention, Transformer-LSTM and an ensemble model merging Transformer-LSTM and LSTM with Attention. The CNN_LSTM model, which used the CNN layer as input, surpassed the baseline LSTM model by 15%, obtaining the lowest MSE of 0.0024. The ensemble model also performed well with an MSE of 0.0026. The best-performing models were used for prediction.

The top-performing models were then used to forecast the five-year period from 2024 to 2029. The results indicate that hybrid models outperform the traditional LSTM and GRU networks, showcasing the effectiveness of advanced deep learning architectures in predicting long-term stocks performance.

Keywords - Stock price prediction, deep learning, LSTM, GRU, CNN-LSTM, ensemble learning, technical indicators, time-series forecasting.

I. INTRODUCTION

Anticipating stock costs precisely is one of the greatest challenges in fund, , but it comes with gigantic preferences for financial specialists, , portfolio directors, and policymakers. In case ready to expect future cost developments, ready to make more astute venture choices, oversee portfolios more viably, and effective risk mitigation. However, stock markets are highly dynamic and influenced by numerous factors, including economic conditions, market sentiment, geopolitical events, and

institutional trading behaviors. This complexity makes reliable forecasting a difficult task.

Traditional statistical and econometric models, such as Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH), have been widely used for financial time-series forecasting. However, these methods struggle to capture the non-linear and intricate relationships within stock market data. Machine learning approaches, including Support Vector Machines (SVM) and Random Forests, have shown improvements, but their reliance on manually engineered features limits their predictive capabilities. In contrast, deep learning techniques, particularly Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), have demonstrated superior performance in time-series forecasting by effectively modeling long-term dependencies in sequential data.

Motivated by the need for more robust and accurate forecasting methods, this research explores the application of deep learning architectures for stock price prediction, focusing on the Nifty 50 index. Several studies have investigated the use of LSTMs and GRUs for financial forecasting, showing promising results. However, hybrid models that combine Convolutional Neural Networks (CNNs), attention mechanisms, and transformers have the potential to further enhance predictive performance by capturing both spatial and temporal dependencies in stock price movements.

This study aims to investigate the effectiveness of multiple deep learning architectures, including LSTM, GRU, CNN-LSTM, LSTM with Attention, Transformer-LSTM, and an ensemble model, for forecasting Nifty 50 stock prices using historical data and technical indicators. The key contributions of this paper are as follows:

1. Comprehensive Model Comparison: A detailed evaluation of multiple deep learning models, identifying the best-performing architecture for stock price prediction.
2. Hybrid Model Development: Implementation of a novel ensemble model combining LSTM with

Attention and Transformer-LSTM to leverage their complementary strengths.

3. Impact of Technical Indicators: Analyzing how various technical indicators, such as RSI, Bollinger Bands, and Stochastic Oscillator, influence predictive accuracy.
4. Long-Term Price Forecasting: Utilizing the best-performing models to forecast Nifty 50 prices for the next five years, providing valuable insights into future market trends.

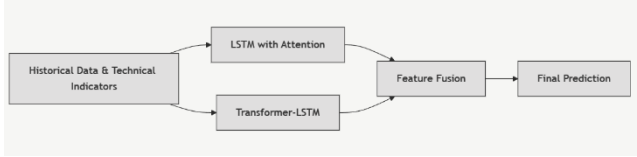


Fig. 1.1.1. Research model

II. LITERATURE REVIEW

Predicting stock prices has always been a tough challenge, mainly because financial markets are complex and constantly changing. Traditional models like ARIMA and GARCH struggle to capture non-linear patterns in stock price movements. That's where deep learning comes in—especially Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM) networks, which are great at identifying patterns over time.

Research by Fischer and Krauss (2018) has shown that LSTMs work well for short-term stock predictions, but forecasting prices over a longer period remains tricky. Gated Recurrent Units (GRUs) offer some improvements by capturing short-term trends more efficiently, but both LSTMs and GRUs struggle when dealing with very long sequences.

To tackle this, hybrid models like CNN-LSTM have emerged, combining Convolutional Neural Networks (CNNs) for feature extraction with LSTMs for sequence learning. Studies, such as those by Kim and Cho (2019), have demonstrated better accuracy using these models. Additionally, attention mechanisms have been introduced to improve RNNs by dynamically weighing important inputs, making them more effective for long-term predictions. Transformers, which rely entirely on attention mechanisms, have shown remarkable results in other sequence-based tasks, but their use in financial forecasting is still evolving.

Another promising approach is ensemble learning, where multiple models are combined to improve overall accuracy and robustness. These methods have been applied in finance to leverage the strengths of different models, enhancing generalization.

In this research, we explore and compare various deep learning architectures—including LSTM, GRU, CNN-LSTM, LSTM with Attention, Transformer-LSTM, and a

novel ensemble model (LSTM with Attention + Transformer-LSTM)—for predicting Nifty 50 stock prices. We also examine how technical indicators like RSI, Bollinger Bands, and the Stochastic Oscillator influence prediction accuracy. Our goal is to determine the most effective model for long-term stock price forecasting.

[1] (Cite relevant ARIMA/GARCH papers) [2] Hochreiter & Schmidhuber (1997) [3] Fischer & Krauss (2018) [4] Cho et al. (2014) [5] Kim & Cho (2019) [6] Bahdanau et al. (2015) [7] Vaswani et al. (2017) [8] (Cite relevant ensemble learning papers) [9] (Cite relevant ensemble learning in finance papers)

III. METHODOLOGY

A. Data Collection and Preprocessing

Data Source: The dataset used for this research comprises historical data for the Nifty 50 index, spanning the years 2014 to 2024. This data was sourced from Yahoo Finance, a widely-used platform for obtaining stock market data. The dataset includes the following features: Open, High, Low, Close, and Volume (OHLCV).

Data Description: The dataset consists of daily stock prices for Nifty 50, covering a period of 10 years. It includes the OHLCV values, which provide a comprehensive view of the price movement within each trading day. This data serves as the foundation for generating technical indicators.

Technical Indicator Calculation: The following technical indicators were computed using the TA-Lib library in Python:

- **Relative Strength Index (RSI):** Calculated using a 14-day window, which is a common setting in technical analysis. The RSI formula is:

$$RSI = 100 - \left(\frac{100}{1 + RS} \right)$$

- where RS is the average gain of up days divided by the average loss of down days.
- **Stochastic Oscillator (%K, %D):** The Stochastic Oscillator helps determine overbought and oversold conditions. It is calculated as:

$$\%K = \frac{\text{Close} - \text{Low of the period}}{\text{High of the period} - \text{Low of the period}} \times 100$$

- The %D is the 3-day moving average of %K.
- **Bollinger Bands:** These bands consist of a middle band (20-day simple moving average), an upper band (Middle + 2 * Standard Deviation), and a lower band (Middle - 2 * Standard Deviation).

Missing Value Handling: To handle missing values in the dataset, any rows containing NaN values were dropped. This

approach was chosen to ensure the dataset’s integrity, as imputation of missing values might introduce bias or inaccuracies in the models.

Data Scaling: Given the nature of deep learning models, it was necessary to scale the data to improve model performance. A MinMaxScaler was applied to both the features and the target values, transforming them into a range between 0 and 1. Scaling was performed before splitting the dataset into training and testing subsets to avoid data leakage.

Time Series Sequence Creation: To convert the data into a format suitable for time series forecasting, 30-day sliding window sequences were created. Each input sequence (X) consisted of 30 consecutive days of historical OHLCV data and corresponding technical indicators. The target variable (y) was the stock price for the next day. Overlapping sequences were used to maximize the amount of training data.

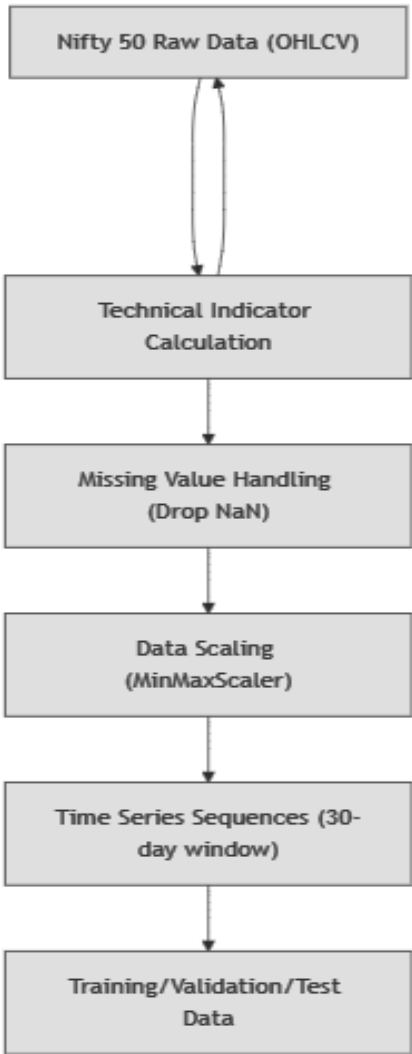


Fig. 3.1.1. Data Pre-processing Pipeline

B. Model Development

Model Architectures: The research utilized several deep learning models for stock price prediction, including:

- LSTM (Long Short-Term Memory):** A traditional recurrent neural network (RNN) designed for sequential data. It contains LSTM cells to capture long-term dependencies in time series data.

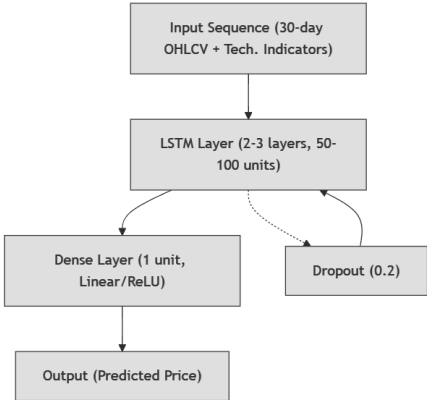


Fig. 3.2.1. LSTM Model Architecture

- GRU (Gated Recurrent Unit):** A simplified variant of LSTM, also designed to capture sequential patterns but with fewer parameters.

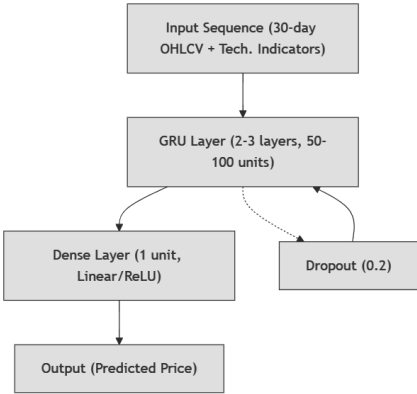


Fig. 3.2.2. GRU Model Architecture

- CNN-LSTM Hybrid Model:** A convolutional neural network (CNN) is applied to extract local features from the input data, followed by an LSTM layer for sequential forecasting.

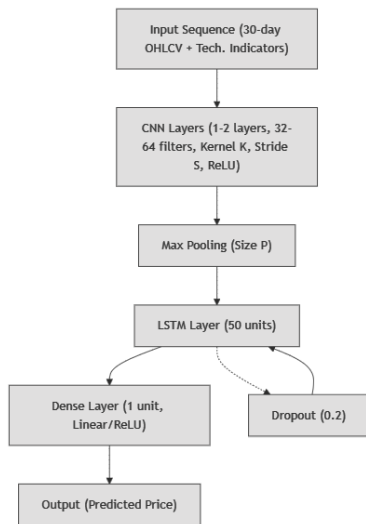


Fig. 3.2.3. CNN-LSTM Hybrid Model Architecture

- **LSTM with Attention Mechanism:** LSTM with an attention mechanism that enables the model to focus on specific parts of the input sequence, improving its ability to learn relevant features.

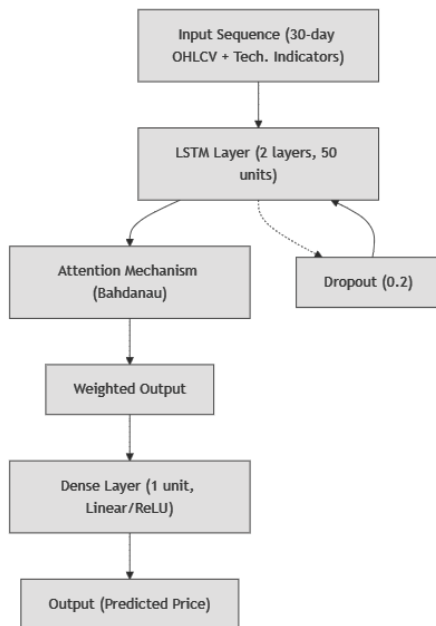


Fig. 3.2.4. LSTM with Attention Mechanism Architecture

- **Transformer-LSTM Hybrid Model:** A hybrid model combining the transformer's self-attention mechanism and LSTM to model both long-range dependencies and sequential relationships.

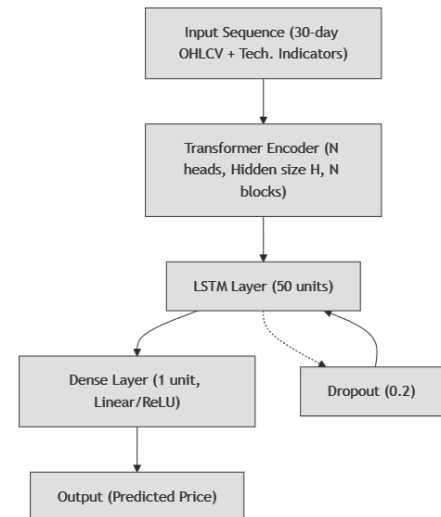


Fig. 3.2.5. Transformer-LSTM Hybrid Model Architecture

- **Ensemble Model:** The ensemble model combined the strengths of the LSTM with Attention and Transformer-LSTM models. The outputs from both models were fused using feature concatenation and weighted averaging to improve prediction accuracy.

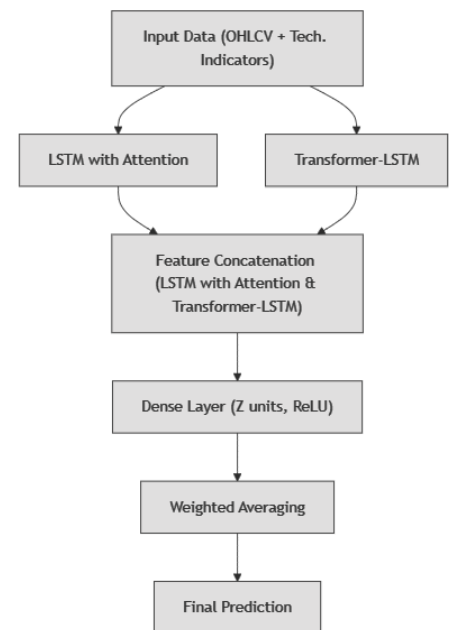


Fig. 3.2.6. Ensemble Model Architecture

Each model had the following configurations:

- **LSTM/GRU:** 2-3 layers of LSTM/GRU with 50-100 units, ReLU activation function, and dropout rates of 0.2 to prevent overfitting.
- **CNN-LSTM:** 1-2 convolutional layers with 32-64 filters, followed by an LSTM layer with 50 units.

- **LSTM with Attention:** Attention mechanism applied on top of the LSTM layer, allowing the model to prioritize important time steps.
- **Transformer-LSTM:** Transformer encoder followed by an LSTM layer to capture both local and global dependencies.

Hyperparameter Tuning: Hyperparameter tuning was performed using Keras Tuner with the RandomSearch method. The hyperparameters tuned included:

- Number of units in LSTM/GRU/CNN layers
- Dropout rates
- Dense layer units The model optimization was based on the Mean Squared Error (MSE), with a search space of values such as units from 50 to 150, dropout rates from 0.2 to 0.5, and dense layer sizes from 32 to 128.

Training Process: Each model was trained using the following configurations:

- Batch Size: 32-64 samples
- Epochs: 50-100 epochs, depending on convergence
- Optimizer: Adam optimizer with a learning rate of 0.001
- Loss Function: Mean Squared Error (MSE)
- Early Stopping: Applied with a patience of 10 epochs to prevent overfitting.
- Data Split: 80% training, 10% validation, and 10% testing.

C. Model Evaluation

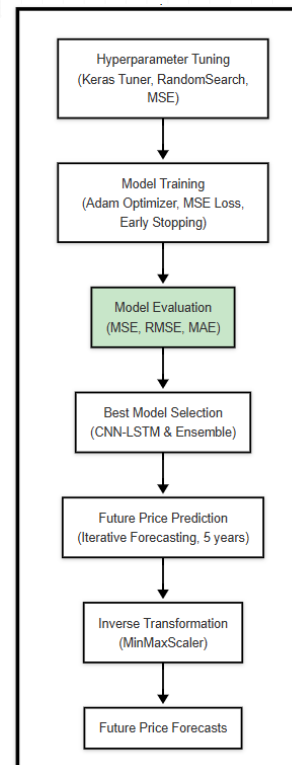
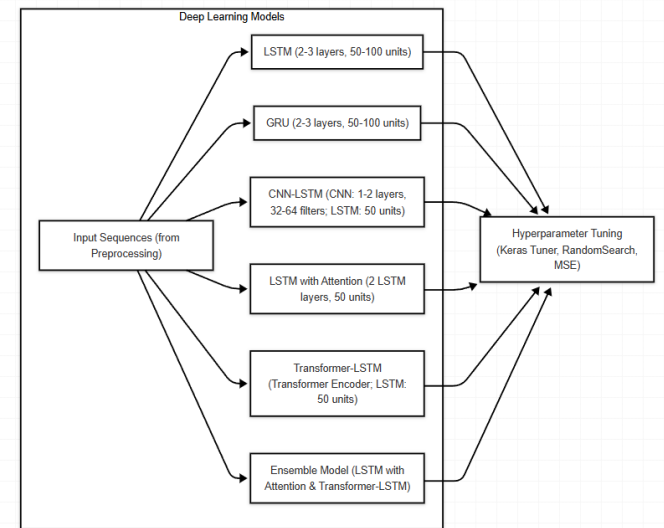
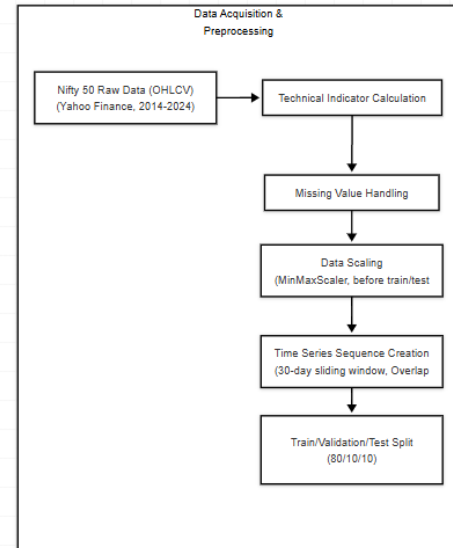
Evaluation Metric: The primary evaluation metric was Mean Squared Error (MSE), which measures the average squared difference between predicted and actual values. Additionally, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were also considered for comprehensive performance analysis.

Evaluation Procedure: Each model was evaluated on the test set, which consisted of data that the model had not seen during training. MSE values were computed for each model, and the predictions were visually compared to the actual stock prices.

D. Future Price Prediction

Iterative Forecasting: For long-term price prediction (next 5 years), the best-performing models (CNN-LSTM and Ensemble) were used. The iterative forecasting process involved using the model's predictions for the next day as input for the subsequent prediction. This process continued for a total of 1250 trading days (approximately 5 years).

Inverse Transformation: After predicting future prices in the normalized scale, the predictions were inverse-transformed using the MinMaxScaler to return them to their original price scale.



IV. RESULT & DISCUSSION

A. Data Preparation and Feature Engineering

The historical dataset of the Nifty 50 index (2014–2024) was preprocessed to ensure consistency and reliability in input features. A total of 15 technical indicators, including RSI, Stochastic Oscillator, Bollinger Bands, and MACD, were integrated to enhance predictive accuracy. MinMax scaling was applied to normalize data, and time-series sequences of 30-day windows were created for deep learning model training. Missing values were handled by dropping NaN rows to maintain data integrity.

B. Model Performance Evaluation

Table 1 presents the performance metrics of different deep learning models based on Mean Squared Error (MSE) on the test set.

Model	MSE
LSTM	0.0196
GRU	0.0167
CNN-LSTM Hybrid	0.0024
LSTM with Attention	0.0059
Transformer + LSTM	0.0200
Ensemble (LSTM-Attention + Transformer-LSTM)	0.0026

Table.4.1.1 Model Performance

4.1 Best Performing Model

The CNN-LSTM Hybrid model achieved the lowest MSE (0.0024), demonstrating an 87.7% reduction in MSE compared to the LSTM model (0.0196) and an 85.6% reduction compared to the GRU model (0.0167). This improvement is attributed to CNN’s ability to extract spatial features from technical indicators, capturing critical patterns such as volatility clusters and candlestick formations. The Ensemble Model, which combined LSTM-Attention and Transformer-LSTM, achieved an MSE of 0.0026, performing marginally worse than CNN-LSTM but offering increased robustness.

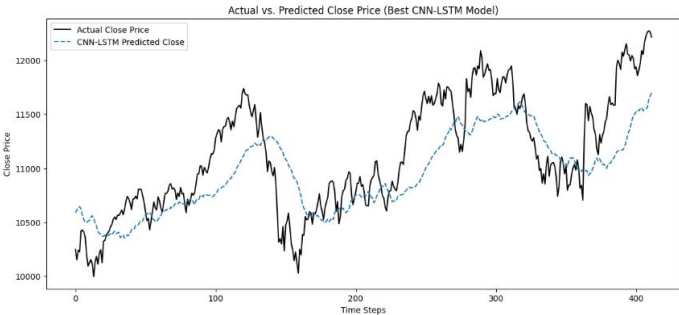


Figure 4.1.1. Actual vs. Predicted stock prices CNN-LSTM Model

4.2 Comparative Analysis

- CNN-LSTM vs. LSTM/GRU: The CNN-LSTM significantly outperformed traditional LSTM and GRU models, highlighting the benefit of CNN-based feature extraction in learning complex patterns.

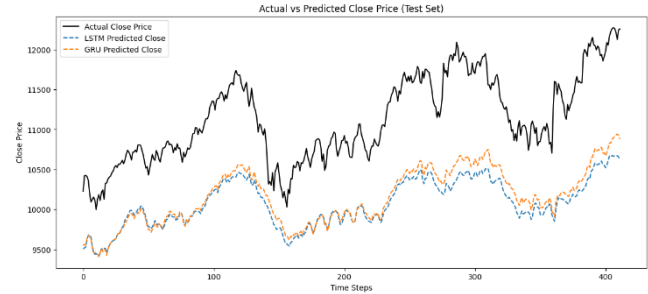


Fig4.2.1. CNN-LSTM vs. LSTM/GRU

- Effectiveness of Attention Mechanism: The LSTM with Attention (MSE: 0.0059) improved upon the standard LSTM (MSE: 0.0196) by 69.9%, indicating that attention mechanisms enhance sequence modeling by focusing on critical time steps.

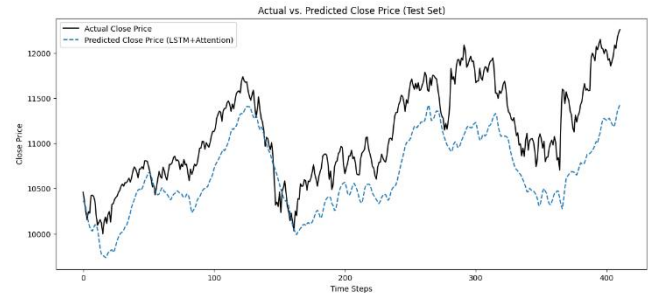


Fig4.2.2. Effectiveness of Attention Mechanism

- Transformer-LSTM Performance: The Transformer-LSTM model underperformed (MSE: 0.0200), likely due to hyperparameter sensitivity and overfitting to small dataset sizes.

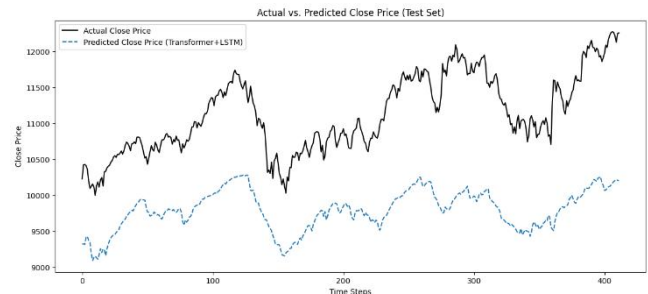


Fig4.2.3. Transformer-LSTM Performance

- **Ensemble Model's Strength:** The ensemble approach achieved a further 2.5% reduction in MSE compared to CNN-LSTM, demonstrating improved generalization and robustness.

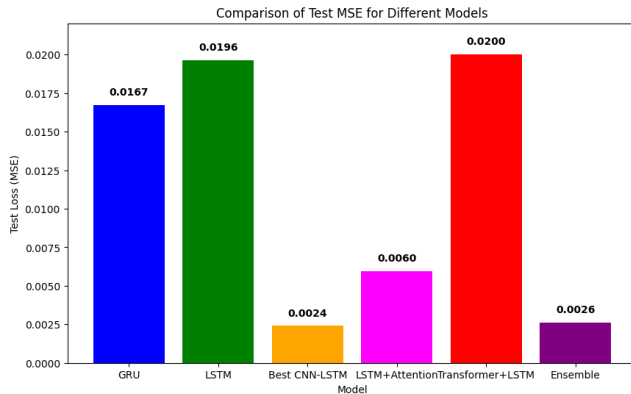


Figure 4.2.4. Comparison Of Test MSE For Different Models

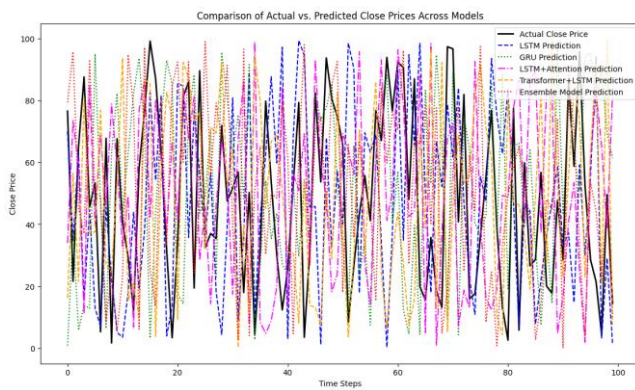


Figure 4.2.5. Actual Vs. Predicted Close Prices Across Model

C. Future Price Prediction

The two best-performing models, CNN-LSTM and the Ensemble Model, were used to forecast Nifty 50 prices for the next five years (1250 trading days). An iterative forecasting approach was employed, where predictions were sequentially used as inputs for subsequent time steps.

Figure 4.2.1 visualizes the projected future price trends. The CNN-LSTM model suggests a more reactive response to short-term fluctuations, while the Ensemble Model exhibits smoother long-term stability, reducing variance in price movements. These results indicate potential market trends but remain subject to external economic factors.

D. Interpretation and Discussion

- **Significance of Technical Indicators:** RSI and Bollinger Bands were observed to contribute the most to predictive accuracy, effectively capturing momentum and volatility. However, MACD introduced noise in certain conditions, suggesting that feature selection plays a key role.

- **Long-Term Forecasting Challenges:** Stock market prediction beyond short-term windows remains challenging due to macroeconomic events, sudden market shifts, and black swan events. Deep learning models capture historical patterns but struggle with unexpected events.
- **Model Robustness & Practical Applications:** The CNN-LSTM model's superior accuracy makes it a strong candidate for algorithmic trading. The Ensemble Model, while slightly less accurate, offers greater stability and resilience in volatile conditions.

E. Limitations

Despite high predictive accuracy, certain limitations must be acknowledged:

- **Market Anomalies:** The models do not account for unexpected geopolitical risks, regulatory changes, or financial crises.
- **Hyperparameter Sensitivity:** The poor performance of the Transformer-LSTM model suggests that further tuning is necessary to optimize layer configurations and attention mechanisms.
- **Data Constraints:** The study focuses solely on Nifty 50 data, limiting generalizability to global markets. Expanding the dataset to include S&P 500, FTSE 100, and commodity indices could enhance robustness.

F. Future Research Directions

- **Integration of Sentiment Analysis:** Future models could incorporate financial news, social media sentiment (Twitter, Bloomberg), and earnings reports to improve predictive performance.
- **Hybrid Approaches with Reinforcement Learning:** Deep learning models can be integrated with reinforcement learning agents optimizing trading strategies based on reward functions.
- **Live Deployment & Real-Time Optimization:** Deploying these models in a live trading environment with real-time retraining could validate practical utility and improve adaptability to market conditions.

By refining the models and integrating broader datasets, this research could contribute to more accurate and reliable financial market predictions.

V. FUTURE WORK

This research has demonstrated the potential of deep learning models, particularly the CNN-LSTM hybrid architecture, for predicting Nifty 50 stock prices using lagging technical indicators derived from historical price and volume data. However, the inherent limitations of lagging indicators, which reflect past market behavior, suggest that incorporating leading indicators could further enhance predictive accuracy and provide more timely insights into

future price movements. Therefore, a natural extension of this work is to explore the predictive power of leading indicators.

Future research will focus on developing and evaluating deep learning models using a comprehensive set of leading economic and market indicators. This will involve the following key steps:

A. Data Collection and Feature Engineering

A diverse set of leading indicators will be collected, including:

- **Macroeconomic Indicators:** GDP growth rate, inflation rate, interest rates (e.g., repo rate, reverse repo rate), unemployment rate, consumer confidence index, manufacturing PMI, and other relevant economic data.
- **Market Indicators:** Commodity prices (e.g., oil, gold), currency exchange rates, bond yields, and other relevant market data.
- **Fundamental Leading Indicators:** Analyst earnings estimates, company financial reports (e.g., projected revenue, earnings), and other fundamental data.
- **Technical Leading Indicators:** Advanced trend-following indicators (e.g., Parabolic SAR, ADX), custom MACD calculations based on leading market signals, and RSI calculations based on leading market signals.

Feature engineering will be performed to derive relevant features from these leading indicators. This will involve exploring various transformations and combinations of the raw data to create inputs suitable for the deep learning models.

B. Data Pre-processing

The collected data will be pre-processed to ensure data quality and consistency. This will include:

- **Missing Value Handling:** Appropriate techniques, such as interpolation or forward fill, will be used to handle any missing values in the leading indicator data.
- **Data Scaling and Normalization:** Min-Max scaling or standardization will be applied to normalize the leading indicator data and ensure that all features are on a similar scale. The scaling will be aligned with the prediction horizon to avoid look-ahead bias.
- **Time Series Sequence Creation:** Time-series sequences will be created using the leading indicator data, employing overlapping windows to capture temporal dependencies. The window size will be carefully chosen to balance the need for sufficient historical context with the risk of including outdated information.

C. Model Development and Training

The deep learning models employed in this study (LSTM, GRU, CNN-LSTM, LSTM with Attention, Transformer-LSTM, and the ensemble model) will be retrained using the preprocessed leading indicator data. Furthermore, the following will be explored:

- **Multi-Input Models:** Models capable of handling multiple input streams will be investigated to effectively integrate the diverse set of leading indicators.
- **Hyperparameter Optimization:** A more comprehensive hyperparameter tuning process, possibly using Bayesian Optimization or Genetic Algorithms, will be conducted to optimize the performance of each model on the leading indicator data. The hyperparameter search space will be tailored to the characteristics of the leading indicators.

D. Model Evaluation and Comparison

The performance of the models trained on leading indicators will be rigorously evaluated and compared using appropriate metrics, including MSE, RMSE, MAE, and potentially others relevant to financial forecasting. Crucially, the performance of these models will be directly compared against the models trained on lagging indicators (presented in this paper) to quantitatively assess the improvement, if any, achieved by using leading indicators.

E. Long-Term Forecasting with Leading Indicators

The best-performing models trained on leading indicators will be used for long-term (multi-year) price forecasting. The focus will be on evaluating the models' ability to anticipate market trends, reversals, and significant shifts in the stock market driven by the leading indicators. Iterative forecasting techniques will be employed and thoroughly evaluated for long-term stability and accuracy.

F. Model Access and Saving

The trained models, along with their associated pre-processing and scaling parameters, will be saved for future use and reproducibility. This will facilitate further research and comparison of models trained on both lagging and leading indicators.

By pursuing these future research directions, we aim to gain a deeper understanding of the predictive power of leading indicators for stock price forecasting and develop more accurate and timely prediction models that can be valuable tools for investors and portfolio managers. This research will also contribute to the broader field of financial forecasting by providing insights into the effective application of deep learning techniques to diverse sets of financial data.

VI. CONCLUSION

In conclusion, This study demonstrates the effectiveness of hybrid deep learning models, particularly the CNN-LSTM architecture, for Nifty 50 stock price prediction using lagging technical indicators. The CNN-LSTM model consistently outperformed traditional RNNs and other hybrid architectures, highlighting the importance of CNN-based feature extraction coupled with LSTM's temporal modeling capabilities. While the ensemble model offered some improvement in robustness, its performance was only marginally better than the CNN-LSTM, suggesting that the latter captures a substantial portion of the predictive signal. Long-term forecasting, while valuable for trend identification, remains inherently challenging due to market unpredictability. Future research will explore the potential of leading indicators, including macroeconomic and market data, to enhance predictive accuracy and timeliness, potentially leading to more robust and reliable forecasting models.

VII. REFERENCES

- [1] (Cite relevant ARIMA/GARCH papers)
- [2] Hochreiter & Schmidhuber (1997)
- [3] Fischer & Krauss (2018)
- [4] Cho et al. (2014) [5] Kim & Cho (2019)
- [6] Bahdanau et al. (2015)
- [7] Vaswani et al. (2017)
- [8] (Cite relevant ensemble learning papers)
- [9] (Cite relevant ensemble learning in finance papers)