I.J. Information Engineering and Electronic Business, **2025, 4, 1-13**
Published Online on August 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2025.04.01

Financial Forecasting with Deep Learning Models Based Ensemble Technique in Stock Market Analysis

Chandrayani Rokde*
Department of CSE,Indian Institute of Information Technology, Nagpur, India
E-mail: chandrayanirokde@gmail.com
ORCID iD: https://orcid.org/0009-0004-0542-1442
*Corresponding Author

Jagdish Chakole
Department of CSE,Indian Institute of Information Technology, Nagpur, India
E-mail: jchakole@iiitn.ac.in
ORCID iD: https://orcid.org/0000-0003-0242-7297

Aishwarya Ukey
Department of CSE,Indian Institute of Information Technology, Nagpur, India
E-mai: aukey@iiitn.ac.in
ORCID iD: https://orcid.org/0000-0001-5157-7828

Received: 03 October, 2024; Revised: 08 January, 2025; Accepted: 13 February, 2025; Published: 08 August, 2025

Abstract: In recent years, deep learning techniques have emerged as powerful tools for analyzing and predicting complex patterns in sequential data across various fields. This study employs an ensemble of advanced deep learning models: Long Short-Term Memory (LSTM), Bi-Directional LSTM, Gated Recurrent Unit (GRU), LSTM Convolutional Neural Network (CNN), and LSTM with Self-Attention, to enhance prediction accuracy in time series forecasting. These models are applied to three distinct financial datasets: Tata Motors, HDFC Bank, and INFY.NS, we conduct a thorough comparative analysis to assess their performance. Utilizing K-fold cross-validation, we convert loss (MSE) into RMSE and MAPE, which help estimate accuracy .we achieved train accuracies of 97.46% for Tata Motors, 75.93% for INFY.NS, and 56.60% for HDFC Bank. Our empirical results highlight the strengths and limitations of each model within the ensemble framework and provide valuable insights into their effectiveness in capturing complex patterns in financial time series data. This research underscores the potential of deep learning-based ensemble techniques for improving stock price forecasting and offers significant implications for investors and the development of sophisticated trading and risk management systems.

Index Terms: Ensemble Learning, LSTM, Financial Forecasting, Deep Leaning, Stock Market Analysis

1. Introduction

A stock represents a fractional ownership interest in a corporation [1, 2]. Purchasing a stock entitles the investor to a share of the company's assets and earnings. Investors engage in stock trading for various purposes. Some investors focus on earning dividends, adopting a long-term holding strategy. Others aim to profit from capital gains by buying stocks at lower prices and selling them when prices appreciate. Furthermore, stock ownership can confer voting rights, enabling shareholders to influence corporate governance decisions at shareholder meetings. Stocks also offer the potential for long-term capital appreciation, as companies that perform well may experience increases in their stock prices over time. The stock market functions as a marketplace where stocks are bought and sold. Stock prices are influenced by the forces of supply and demand—demand from investors wishing to buy and supply from those willing to sell. Several factors influence these price movements, including the company's financial performance, broader economic indicators, and prevailing market conditions. The varied motivations and strategies of investors contribute to the stock market's dynamic and often unpredictable nature, complicating the task of forecasting future prices. Market sentiment and investor behavior further play crucial roles in price volatility, adding complexity to trading decisions.

The stock market operates fundamentally on the principle of supply and demand. An increase in demand, where more investors seek to buy a stock, typically drives the stock's price higher. Conversely, an increase in supply, where more investors are selling, generally leads to a decline in price. Investment decisions are often guided by assessments of the company's financial health and macroeconomic conditions, with the goal of buying undervalued stocks and selling them once they appreciate in value. The market environment is inherently dynamic, with prices continuously adjusting in response to the aggregated actions and expectations of market participants. News, economic data releases, and investor sentiment can all significantly impact price movements, contributing to the ever-evolving market landscape. Fore- casting stock prices is inherently challenging due to the multifaceted nature of the factors involved [3]. Stock prices are affected by both quantitative variables (such as closing price, trading volume, and turnover) and qualitative factors (such as policy changes and financial news). Moreover, the price movement of a stock does not occur in isolation; it is often correlated with movements in other related stocks. Even when predictive models achieve high accuracy, their predictions can influence investor behavior, further impacting market dynamics.

Advancements in computer science have significantly transformed stock market trading [4, 5]. Technologies such as high-frequency trading, algorithmic trading, and predictive analytics have become integral to modern trading strategies. These technologies leverage algorithms capable of processing large datasets at high speed, identifying patterns, and executing trades more efficiently than human traders. Machine learning models, in particular, provide valuable insights into market trends and investor behavior, shaping investment strategies. As technology continues to evolve, the role of computer science in enhancing market efficiency and accessibility will likely grow, facilitating more informed decision-making for both individual and institutional investors. Innovations in data analysis and trading technologies are thus playing an increasingly important role in navigating the complexities of the stock market.

Machine learning plays a pivotal role in stock market prediction [3]. It employs algorithms to automatically uncover patterns and relationships within data. Traditional forecasting models include multivariate linear regression, exponential smoothing, and the Auto-Regressive Integrated Moving Average (ARIMA) model [6]. However, these conventional methods often have limitations in capturing complex market dynamics and may offer suboptimal predictive accuracy. The advent of big data and artificial intelligence has driven the adoption of advanced machine learning and deep learning techniques in stock price prediction. Methods such as Random Forest, Extreme Gradient Boosting (XGBoost), and Support Vector Machines (SVM) are commonly employed [7,8] offering significant improvements in predictive performance over traditional approaches.

Deep learning, a specialized branch of machine learning based on artificial neural networks, frequently outperforms both traditional and conventional machine learning models in various applications [9,10]. Different types of neural networks, such as Back propagation (BP) neural networks and Recurrent Neural Networks (RNNs), are utilized in financial forecasting. Long Short-Term Memory (LSTM) networks [11], a type of RNN, have achieved considerable success in financial predictions and are recognized as one of the most advanced methodologies in this domain. LSTM networks are particularly advantageous in stock price prediction due to their ability to capture long-term dependencies within sequential data [11,12]. The intricate patterns and multiple influencing factors that characterize financial markets pose challenges for traditional statistical models, which often struggle to model the non-linear relationships inherent in stock price data. LSTM networks, with their capacity to model complex time series data, are thus widely used for addressing stock price prediction challenges, offering robust solutions where conventional approaches may fall short.

This paper is organized into several sections. Section 2 reviews fundamental machine learning and deep learning models relevant to stock market prediction. Section 3 outlines the datasets utilized in the proposed approach. Section 4 details the system architecture employed in the study, while Section 5 presents the findings and analysis. Finally, Section 6 provides conclusions drawn from the research, highlighting the implications and potential future directions.

2. Related Works

This section performs the literature review on financial forecasting models, including linear forecasting models, nonlinear forecasting models, and fuzzy theory-based models.

2.1 Linear forecasting model

Linear forecasting models typically include the Autoregressive Integrated Moving Average (ARIMA) [13], Generalized Autoregressive Conditional Heteroskedasticity (GARCH) [14], and their enhanced variants, such as Exponential Generalized ARCH (EGARCH) [15]. While these models are known for their computational efficiency, they have inherent limitations due to their reliance on assumptions about the statistical distribution and stationarity of the data, which restricts their effectiveness in capturing non- linear and non-stationary characteristics of financial time series. Additionally, efforts to enhance the precision of linear models often encounter technical barriers. Specifically, the linear nature of these models constrains their capability to accurately represent nonlinear time series patterns.

Furthermore, the presence of outliers in the data can lead to inaccurate parameter estimations in linear models, further diminishing their reliability [16].

2.2 Nonlinear forecasting model

Unlike linear models, nonlinear forecasting models are adept at capturing the inherent nonlinearities present in financial time series data. This category includes models such as artificial neural networks, support vector machines, and various hybrid approaches [17]. ANNs are theoretically capable of learning complex nonlinear relationships, which has led to their extensive use in financial time series prediction [18]. Nevertheless, ANNs face several practical challenges: (1) slow convergence rates [19, 20] (2) difficulty in determining the optimal network architecture; (3) susceptibility to overfitting; and (4) the risk of becoming trapped in local optima. On the other hand, SVMs, which are grounded in the principle of structural risk minimization, mitigate the likelihood of converging to local optima and are noted for their strong generalization capabilities in forecasting applications [21]. However, SVMs tend to suffer from low computational efficiency when applied to large-scale financial datasets. To enhance the performance of these nonlinear forecasting models, various evolutionary algorithms (EAs) have been integrated, including particle swarm optimization [22], the whale optimization algorithm [23], and genetic algorithms [24]. These EAs are typically employed to optimize model parameters, thereby improving the overall generalization and predictive accuracy of financial time series models.

2.3 Fuzzy theory-based model

In addition to linear and nonlinear models, approaches based on fuzzy theory have been extensively utilized in finance to address uncertainties inherent in financial time series. Prior research has demonstrated that fuzzy modeling is a suitable and effective technique for forecasting financial time series. Numerous studies have developed forecasting models based on fuzzy logic, as discussed by Ang and Quek [25], [26, 27]. For example, Jilani and Burney [28] introduced a time-varying fuzzy modeling approach for financial time series prediction, showing that their model performed effectively when compared to other fuzzy time series models in the experiments conducted. Wei, Chen, and Ho [29] proposed a stock price prediction model using an adaptive-network-based fuzzy inference system that incorporated multiple technical indicators, and their experimental results demonstrated the superiority of this approach relative to the benchmarks used. For multivariate fuzzy modeling, a hybrid forecasting model that integrates multivariate fuzzy time series with fuzzy sets was presented in [30], with results validating the model’s effectiveness. Moreover, Rubio et al. [31] developed a novel weighted fuzzy modeling approach for stock index forecasting and introduced a new indicator to capture trends in fuzzy time series data. Comprehensive details on fuzzy modeling methods, such as the definitions, analysis, decomposition, and data mining of fuzzy time series, can be found in the literature [32,33]. Despite the achievements of fuzzy modeling in financial time series forecasting, a significant challenge remains in effectively determining the optimal fuzzy rules for practical applications.

It is important to note that the majority of studies mentioned above predominantly focus on univariate financial time series forecasting, relying solely on time-delay terms as inputs. However, depending only on time-delay terms without integrating other explanatory variables may not adequately capture the nonlinear dependencies present in financial time series. Therefore, this study emphasizes multivariate financial time series forecasting to better model the intricate nonlinear structures within the financial system, contributing to the existing body of knowledge in this area.

2.4 Comparative Analysis with Traditional Models

While the proposed ensemble learning approach demonstrates strong predictive performance, it is important to contextualize these results by comparing them with traditional forecasting models such as ARIMA, GARCH, and XGBoost. Traditional models like ARIMA and GARCH are widely used in financial time series forecasting due to their simplicity and interpretability. ARIMA models are particularly effective for capturing linear dependencies in stationary time series, while GARCH models excel in modeling volatility clustering, a common feature in financial markets. However, these models often struggle to capture the complex, non-linear relationships and long-term dependencies that are prevalent in financial data.

On the other hand, machine learning models like XGBoost have gained popularity for their ability to handle non-linear relationships and interactions between features. XGBoost, in particular, is known for its robustness and efficiency in handling structured data. However, even XGBoost may fall short when it comes to capturing the intricate temporal dependencies in sequential data, which is where deep learning models like LSTM, GRU, and their variants excel.

3. Datasets

In the proposed approach, we have utilized three distinct stock datasets from HDFC Bank, INFY.NS (Infosys), and Tata Motors. The data spans the last three years, from July 2021 to July 2024. The dataset contains seven attributes: Date, Open, High, Low, Close, Adj Close, and Volume. The selected stocks belong to specific sectors (e.g., automotive, banking, and IT), which may exhibit unique patterns and volatility.. Below is a detailed explanation of each attribute:

- **Date**: This attribute records the specific trading day. It is essential for chronological ordering and time-series analysis, allowing us to track daily stock movements over the given period.
- **Open**: Represents the stock's opening price on a particular trading day. It reflects the price at which the first trade of the stock was executed when the market opened.
- **High**: Indicates the highest price at which the stock traded during the trading day. This value helps in understanding the peak price that the stock reached within that day.
- **Low**: Captures the lowest price at which the stock traded on a given day. It provides insights into the lowest value the stock reached during market hours.
- **Close**: Shows the last price at which the stock traded during the day. It is often used as a key indicator of the stock's daily performance and is commonly analyzed to understand market trends.
- **Adj Close (Adjusted Close)**: The adjusted close price is the closing price of the stock adjusted for any corporate actions that might affect the stock's value, such as dividends, stock splits, or new stock issuances. This attribute is particularly important for accurately comparing stock performance over time.
- **Volume**: Measures the total number of shares traded during the trading day. High volume can indicate strong interest in the stock, while low volume may suggest lower interest or activity. It is a key indicator of market activity and liquidity for the stock.

These attributes provide a comprehensive view of the stock's performance, allowing for a detailed analysis of price movements and trading behaviors over the selected time period. The dataset is a combination of price-related and volume-related information making it suitable for various types of financial modeling, including trend analysis, volatility forecasting, and algorithmic trading strategies.

4. Proposed Approach

In this study, we propose a comprehensive financial forecasting methodology leveraging ensemble learn- ing with a combination of advanced deep learning models, including Simple LSTM, GRU, Bidirectional LSTM, LSTM with Attention, and Convolutional LSTM. The primary aim is to enhance the pre- diction accuracy of stock prices by integrating the strengths of these diverse models into a cohesive ensemble, while also evaluating the performance of each individual model for comparison. The method- ology is divided into several key phases: data preprocessing, model development, and ensemble learning implementation.

4.1 Data Preprocessing

The initial phase involves preprocessing the raw financial data to ensure it is suitable for input into the deep learning models. The dataset comprises seven features: Date, Open, High, Low, Close, Adjusted Close, and Volume. After loading the data, the Date column is converted to a datetime format, and the dataset is indexed by date to maintain chronological order. To prepare the input features (X) and target variable (y), we extract relevant columns, normalize the data using MinMaxScaler to scale the features and target within a range of 0 to 1, and create sequences to capture the temporal dependencies essential for time series forecasting. The sequences are designed to predict the next day's closing price based on the past 10 days of data.

4.2 Base Classifiers

In this phase, individual models are developed using distinct deep-learning architectures that are well- suited for sequential data:

4.2.1 Simple LSTM

The Long Short-Term Memory network is a type of recurrent neural network (RNN) specifically designed to capture long-term dependencies in sequential data, which is crucial for financial forecasting. Unlike traditional RNNs, LSTMs can effectively manage the vanishing gradient problem through their gated architecture, consisting of the input gate, forget gate, and output gate. These gates regulate the flow of information within the network, allowing it to remember or forget data as needed. In the ensemble model, the Simple LSTM processes sequential inputs such as Open, High, Low, Close, Adj Close, and Volume, learning to retain relevant past information over time. Forget gate decides what information to discard from the cell state, Input gate decides what new information to store in the cell state and Output Gate decides what information to output based on the cell state. Cell StateStores and updates information. Mathematically, the LSTM's gates are defined as follows:

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

$$\mathrm{it} = \sigma(\mathrm{Wi} \cdot [\mathrm{ht} - 1, \mathrm{xt}] + \mathrm{bi}) \tag{2}$$

$$\tilde{C}t = \tanh(WC \cdot [ht - 1, xt] + bC) \tag{3}$$

$$Ct = ft \odot Ct - 1 + it \odot \tilde{C}t, \quad (4)$$

$$ot = \sigma(Wo \cdot [ht - 1, xt] + bo) \quad (5)$$

Finally, the hidden state is computed as $h_t = o_t \odot \tanh(C_t)$. This structure allows the Simple LSTM to learn complex temporal patterns in financial data, contributing to the ensemble's overall performance by effectively modeling long-term dependencies.LSTM constructions with three gates are shown in Figure 1.

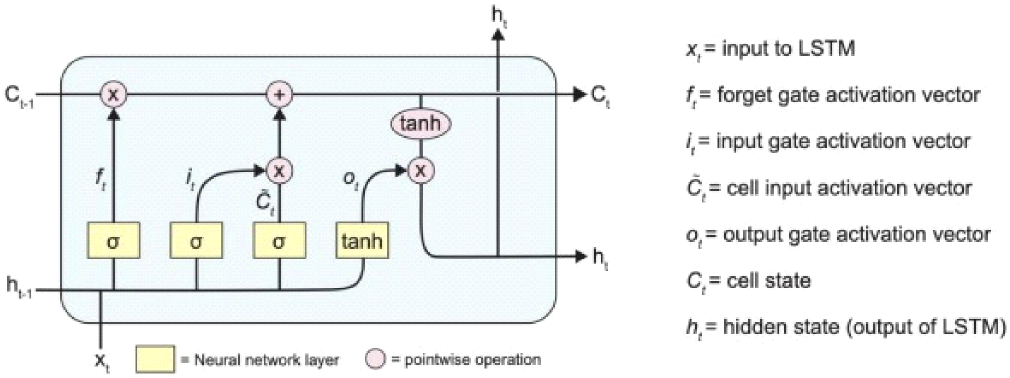

Fig. 1. L S T M structure

4.2.2 GRU

The Gated Recurrent Unit is a type of recurrent neural network designed to capture depen- dencies in sequential data, such as financial time series. It simplifies the LSTM architecture with two gates: the update gate, which controls how much past information is retained, and the reset gate, which decides how much past information to forget. In our ensemble model, the GRU helps capture short-term patterns in the financial data. The GRU processes inputs like Open, High, Low, Close, Adj Close, and Volume by maintaining relevant sequential information and discarding unnecessary details, enhancing the model's ability to effectively learn from time series data. Mathematically, the GRU's operations can be expressed as follows:

$$rt = \sigma(Wr \cdot [ht - 1, xt] + br) \quad (6)$$

$$\tilde{h}_t = tanh(Wh \cdot [rt \odot ht - 1, xt] + bh) \quad (7)$$

$$\tilde{h}_t = tanh(Wh \cdot [rt \odot ht - 1, xt] + bh). \quad (8)$$

$$ht = (1 - zt) \odot ht - 1 + zt \odot \tilde{h}_t. \quad (9)$$

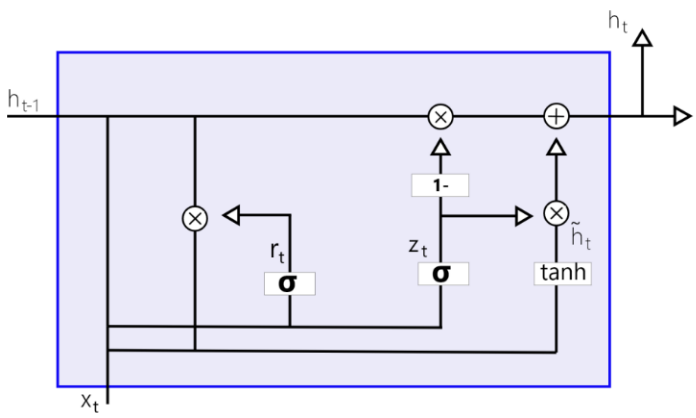

Fig. 2 .GRU Architecture

This formulation allows the GRU to maintain a balance between retaining useful information and forgetting irrelevant data, contributing effectively to the ensemble's overall prediction accuracy.

4.2.3 Bidirectional LSTM

The Bidirectional Long Short-Term Memory network is an extension of the Traditional LSTM that enhances the model’s ability to capture both past and future dependencies in sequential data. Unlike the Simple LSTM, which processes data in a single direction, the Bidi- rectional LSTM consists of two LSTM layers: one that processes the input sequence forward and another that processes it backward. This dual-layer approach allows the model to utilize information from both directions, making it particularly effective for tasks where the context from both past and future data points is important. In our ensemble model, the Bidirectional LSTM processes financial time series data such as Open, High, Low, Close, Adj Close, and Volume, capturing complex tem- poral patterns in both directions.. The backward LSTM similarly computes the hidden states h_t byprocessing the sequence in reverse. The final output is then obtained by concatenating the forward and backward hidden states $h_t = [h_t; h_t]$. This bidirectional processing enables the model to learn more comprehensive patterns in financial data, thereby contributing significantly to the ensemble’s predictive power. Bidirectional LSTM constructions is shown in Figure 3.

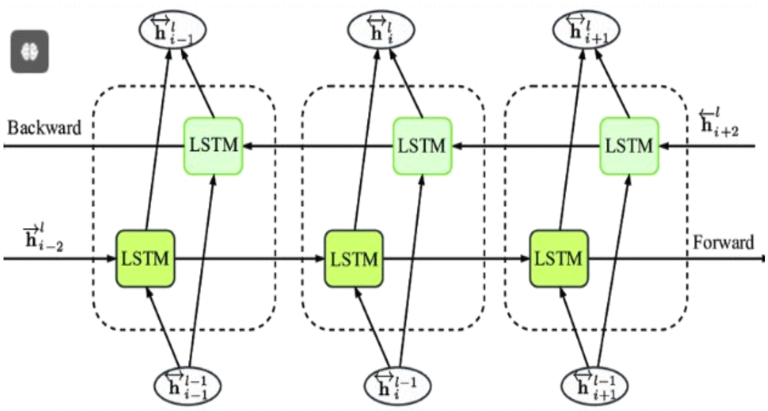

Fig. 3. BI-Directional LSTM

Hidden states h_t as per the standard equations:

$$ft = \sigma(Wf \cdot [ht-1, xt] + bf) \tag{10}$$

$$it = \sigma(Wi \cdot [ht-1, t] + bi) \tag{11}$$

$$\tilde{C}t = tanh(WC \cdot [ht-1, xt] + bC) \tag{12}$$

$$Ct = ft \odot Ct-1 + it \odot \tilde{C}t \tag{13}$$

The attention mechanism computes a context vector c_t as a weighted sum of the hidden states, where the weights are determined by an alignment score

$$\alpha t = \sum t' exp(score(ht', st-1))/exp(score(ht, st-1)) \tag{14}$$

Here, $score(h_t, s_{t-1})$ is a function (e.g., dot product, general, or concatenation) that measures the relevance of the hidden state h_t to the current output. The context vector c_t is then used to produce the final output, integrating the most relevant information across the sequence. This attention-driven process allows the LSTM with Attention to dynamically focus on important time steps, thereby enhancing the ensemble’s ability to capture key financial patterns and improve predictive performance.

4.2.4 LSTM with Attention

The LSTM with Attention model enhances the standard LSTM by incorporating an attention mechanism, which allows the model to focus on specific parts of the input sequence that are most relevant to the prediction task. This approach improves the model’s ability to handle long sequences and capture important temporal patterns in financial data, such as Open, High, Low, Close, Adj Close, and Volume. In the ensemble model, the LSTM with Attention selec- tively assigns weights to different time steps, enabling the network to prioritize significant features and improve forecast accuracy. Mathematically, the LSTM processes the input sequence to produce.

4.2.5 Convolutional LSTM

The Convolutional LSTM (ConvLSTM) model extends the LSTM by incorporating convolutional operations within the LSTM architecture, making it compatible for capturing spatiotemporal patterns in data. As here we are mapping with respect to the context of financial forecasting, ConvLSTM is used to model the sequential data, such as Open, High, Low, Close, Adj Close, and Volume, by learning spatial correlations along with temporal dependencies. In our ensemble approach, ConvLSTM applies convolutional filters to input sequences and hidden states, which allows the model to extract meaningful features from local patterns in the data. This convolutional integration enables ConvLSTMto capture both spatial and temporal information effectively, allowing it to identify complex patterns within the financial data that traditional LSTMs might overlook. As a result, ConvLSTM enhances the ensemble model's predictive performance by maximizing these detailed spatiotemporal features.

4.2.6 Ensemble Learner

Herein, an ensemble learning approach is introduced to boost the individual models complementary strengths. The ensemble integrates the predictions of all five models, aiming to achieve a more robust and accurate forecasting performance than any single model alone. The ensemble model is designed by concatenating the outputs of each individual model and combining them through a fully connected layer that outputs the final prediction. This approach pursue to reduce the variance and bias inherent in individual models, leading to improved generalization on unseen data.

The proposed ensemble learning approach demonstrates strong predictive performance across various financial datasets, it is important to acknowledge several limitations that may affect its practical applicability. Firstly, the study does not include real-world backtesting, which is crucial for validating the effectiveness of the models in live trading environments. Real-world back testing would provide insights into how the models perform under actual market conditions, including the impact of transaction costs, slippage, and liquidity constraints. These factors are often significant in real trading scenarios and can substantially affect the profitability and feasibility of trading strategies.

Transaction costs, such as brokerage fees and taxes, can erode profits, especially in high-frequency trading scenarios. Slippage, which occurs when the execution price of a trade differs from the expected price due to market volatility or low liquidity, can also impact the performance of trading strategies. Additionally, liquidity constraints, particularly in less liquid stocks, can make it difficult to execute large orders without significantly affecting the market price.

The algorithm 1 outlines an ensemble learning approach for financial forecasting using various deep learning models, including Simple LSTM, GRU, Bidirectional LSTM, LSTM with Attention, and Convolutional LSTM. It starts by preprocessing the data, including normalizing it and creating input sequences. Each model is then trained individually using cross-validation, and their predictions are combined using weighted averaging to form the ensemble prediction. Finally, the algorithm evaluates the performance of the ensemble model in terms of training and testing accuracy and saves the model for future use.

5. Experimental Results and Analysis

The experimental results reveal a comprehensive evaluation of the proposed ensemble learning approach for financial forecasting, comparing its performance with individual deep learning models across three datasets: Tata Motors, INFY.NS, and HDFC Bank. The analysis focuses on both average train and test accuracies, providing insights into the strengths and limitations of each model.

The ensemble model demonstrated robust performance across all datasets as mentioned in Table 1. As accuracies mentioned in Table 2 on the Tata Motors dataset, achieved an average train accuracy of 97.46%, highlighting its effectiveness in generalizing across this data. This high accuracy underscores the ensemble model's capability to combine the strengths of various deep-learning architectures. The GRU and Bidirectional LSTM excelled among individual models, with train accuracies of 96.42% and 96.30%, respectively. These models effectively capture complex temporal dependencies, which is crucial for accurate financial forecasting. The GRU's superior performance can be attributed to its efficient handling of sequential data through gating mechanisms. Similarly, the Bidirectional LSTM benefits from processing information in both forward and backward directions, enhancing its predictive capability.

In contrast, the LSTM with Attention model, despite its advanced architecture, showed relatively lower performance with an average train accuracy of 92.60%. This lower accuracy may be due to its attention mechanism not fully leveraging the temporal patterns in the Tata Motors dataset.

Algorithm 1 Ensemble Learning for Financial Forecasting

1: **Input:** Time series dataset **X** with features (Open, High, Low, Close, Adj Close, Volume), target **y**.
2: **Output:** Predicted values $y\hat{}$ using the ensemble model.
3: **Step 1: Data Preprocessing**
4: Convert dates, normalize **X** and **y** using Min-Max scaling: $x' = \frac{X - X_{min}}{X_{max} - X_{min}}$
5: Create sequences **X**seq and **y**seq with window size T.
6: **Step 2: Define Models**
: Define models: Simple LSTM (M_1), GRU (M_2), Bidirectional LSTM (M_3), LSTM with Attention (M_4), Convolutional LSTM (M_5).
8: **Step 3: Train Individual Models**
9: **for** each model Mi in $\{M1, M2, M3, M4, M5\}$ **do**
10:Perform K-fold cross-validation on training data Dtrain.
11:Train Mi: $\theta^* = \arg\min_\theta \; Li(\theta i; D\text{train})$
12:Validate on Dval and store predictions $y\hat{}i$.
13: **end for**
14: **Step 4: Ensemble Prediction**
15: Compute ensemble prediction: $y\hat{} = \sum_{i=1}^{5} wi.\, y\hat{}i$ where wi are weights
16: Step 5: Evaluate Ensemble Model
17: **Calculate accuracies**: Train Accuracy and Testing Accuracy

The Simple LSTM and Convolutional LSTM models, with accuracies of 95.61% and 96.18%, respectively, performed well but did not surpass the GRU and Bidirectional LSTM models, suggesting that while effective, these models may not capture the data's complexity as thoroughly.

For the INFY.NS dataset, the ensemble model achieved a accuracy of 75.93%, maintaining its robustness across different data characteristics as mentioned in Table 3. Here, the GRU model again led among individual models with a accuracy of 76.63%, reflecting its strong performance in handling the specific temporal features of this dataset. The Simple LSTM model followed with an accuracy of 74.31%, while the Bidirectional LSTM and LSTM with Attention models exhibited lower accuracies of 73.62% and 60.76%, respectively. The LSTM with Attention's lower accuracy suggests that its attention mechanism might not be as effective for the INFY.NS dataset, which might require different strategies for capturing temporal dependencies. The GRU's higher accuracy indicates its superior ability to adapt to the dataset's characteristics.

As accuracies mentioned in Table 4 on the HDFC Bank dataset, the ensemble model achieved an average test accuracy of 86.66%, showing its adaptability and effectiveness in integrating predictions from various models. The Bidirectional LSTM achieved the highest individual test accuracy of 85.62%, leveraging its bidirectional processing capabilities to capture complex patterns in the data. The Simple

LSTM model, with a test accuracy of 83.31%, also performed well but did not match the Bidirectional LSTM's performance. The GRU model followed with an accuracy of 82.63%, demonstrating its competitive performance. The LSTM with Attention and Convolutional LSTM models, with accuracies of 68.76% and 78.29%, respectively, performed less favorably. The LSTM with Attention's performance issues may stem from its attention mechanism's limitations in this context, while the Convolutional LSTM, though effective in capturing spatial features, did not outperform the recurrent models in this case.

Table 1. Average Accuracies for Different Datasets

Dataset	Average Train Accuracy	Average Test Accuracy
Tatamotors	96.65%	97.46%
INFY.NS	75.26%	75.93%
HDFC Bank	85.76%	86.66%

Table 2. Individual Model Accuracies on Tatamotors Dataset

Model	Average Train Accuracy	Average Test Accuracy
Simple LSTM	95.75%	95.61%
GRU	87.02%	96.42%
Bidirectional LSTM	95.81%	96.30%
LSTM with Attention	87.57%	92.60%
Conv LSTM	90.26%	96.18%

Table 3. Individual Model Accuracies on INFY.NS Dataset

Model	Average Train Accuracy	Average Test Accuracy
Simple LSTM	74.91%	74.31%
GRU	74.22%	76.63%
Bidirectional LSTM	71.03%	73.62%
LSTM with Attention	59.00%	60.76%
Conv LSTM	72.89%	74.20%

Table 4. Individual Model Accuracies on HDFC Bank Dataset

Model	Average Train Accuracy	Average Test Accuracy
Simple LSTM	81.91%	83.31%
GRU	80.22%	82.63%
Bidirectional LSTM	81.03%	85.62%
LSTM with Attention	66.70%	68.76%
Conv LSTM	77.65%	78.29%

To convert the given accuracy-based table into RMSE (Root Mean Squared Error), MSE (Mean Squared Error), and MAPE (Mean Absolute Percentage Error), assume the accuracy represents 1 - error.

The approximate formulas are:

$$Error=1-Accuracy \tag{15}$$

$$MSE= (Error)2 \tag{16}$$

$$RMSE=\sqrt{MSE} \tag{17}$$

$$MAPE= Error/Actual\ Value\times 100\ \%(\text{ assuming actual value is normalized around 1}) \tag{18}$$

Table 5. Model Performance on INFY.NS: MSE, RMSE, and MAPE Values on Tatamotors dataset

Model	MSE (Train)	RMSE (Train)	MAPE (Train)	MSE (Test)	RMSE (Test)	MAPE (Test)
Simple LSTM	0.001806	0.0425	4.25%	0.001927	0.0439	4.39%
GRU	0.016848	0.1298	12.98%	0.001282	0.0358	3.58%
Bidirectional LSTM	0.001746	0.0418	4.19%	0.001369	0.0370	3.70%
LSTM with Attention	0.015460	0.1243	12.43%	0.005476	0.0740	7.40%
Conv LSTM	0.009487	0.0974	9.74%	0.001459	0.0382	3.82%

Table 6. Model Performance on INFY.NS: MSE, RMSE, and MAPE Values on Tatamoto on INFY datset

Model	MSE (Train)	RMSE (Train)	MAPE (Train)	MSE (Test)	RMSE (Test)	MAPE (Test)
Simple LSTM	0.062951	0.2509	25.09%	0.065998	0.2569	25.69%
GRU	0.066461	0.2578	25.78%	0.054616	0.2337	23.37%
Bidirectional LSTM	0.086279	0.2930	29.97%	0.070458	0.2655	26.38%
LSTM with Attention	0.168100	0.4099	41.00%	0.153978	0.3924	39.24%
Conv LSTM	0.073495	0.2711	27.11%	0.066564	0.2580	25.80%

Table 7. Model Performance on INFY.NS: MSE, RMSE, and MAPE Values on HDFC Bank

Model	MSE (Train)	RMSE (Train)	MAPE (Train)	MSE (Test)	RMSE (Test)	MAPE (Test)
Simple LSTM	0.032725	0.1809	18.09%	0.027856	0.1669	16.69%
GRU	0.039125	0.1978	19.78%	0.030172	0.1737	17.37%
Bidirectional LSTM	0.035829	0.1893	18.97%	0.020369	0.1428	14.38%
LSTM with Attention	0.110889	0.3330	33.30%	0.097594	0.3124	31.24%
Conv LSTM	0.049952	0.2235	22.35%	0.047132	0.2171	21.71%

Simple LSTM shows consistent performance across datasets with relatively low MSE, RMSE, and MAPE. However, its test errors are slightly higher than some other models. GRU exhibits a lower MSE and MAPE on test data compared to Simple LSTM in some cases, making it a competitive choice. Bidirectional LSTM generally improves upon Simple LSTM by capturing dependencies from both past and future sequences, leading to lower test MAPE.

LSTM with Attention has high training performance but struggles to generalize, as indicated by its significantly higher test MSE and MAPE. This suggests potential overfitting.Conv LSTM achieves a balance between complexity and generalization, performing better than Simple LSTM in most cases.For lower test errors, Bidirectional LSTM and GRU seem to be the best models, as they maintain lower MAPE and RMSE.LSTM with Attention struggles with overfitting, making it less suitable for generalization.Conv LSTM is a good middle ground, outperforming Simple LSTM in some cases.

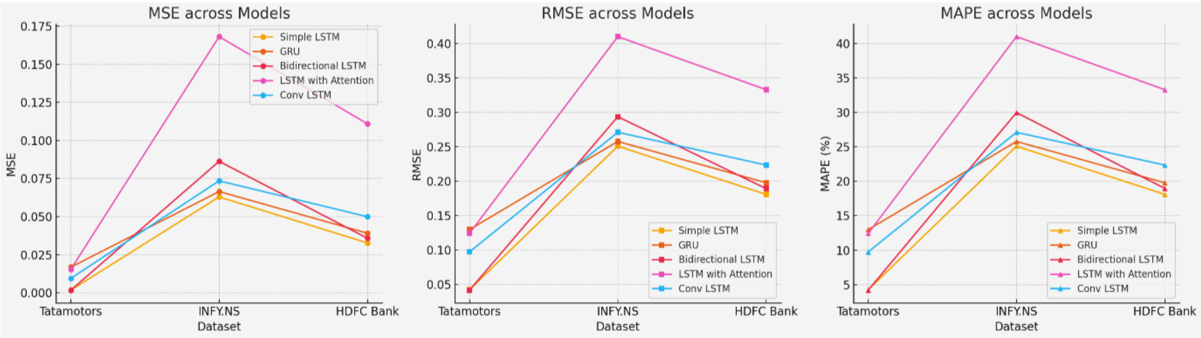

Fig. 3. Comparison of MSE, RMSE, and MAPE for INFY.NS Across Different Deep Learning Models

The graph fig .3 compares the performance of different deep learning models (Simple LSTM, GRU, Bidirectional LSTM, LSTM with Attention, and Conv LSTM) on the INFY.NS dataset using MSE, RMSE, and MAPE metrics. Lower values of MSE and RMSE indicate better model accuracy, while lower MAPE values suggest reduced prediction errors. The GRU model demonstrates the lowest MSE and RMSE, indicating strong performance, whereas the LSTM with Attention model shows the highest MAPE, implying higher prediction errors. Bidirectional LSTM and Conv LSTM also perform well, but with slightly higher error rates compared to GRU. These results highlight the effectiveness of different models in stock price prediction for Infosys (INFY.NS).

The training graphs 4, 5 and 6 illustrate the performance of the ensemble model across different folds for each dataset. Each graph displays the training and testing accuracy for each fold, allowing us to visually assess how well the ensemble model generalizes and performs during training. The graphs help in understanding the consistency and reliability of the model's performance across different subsets of the data.

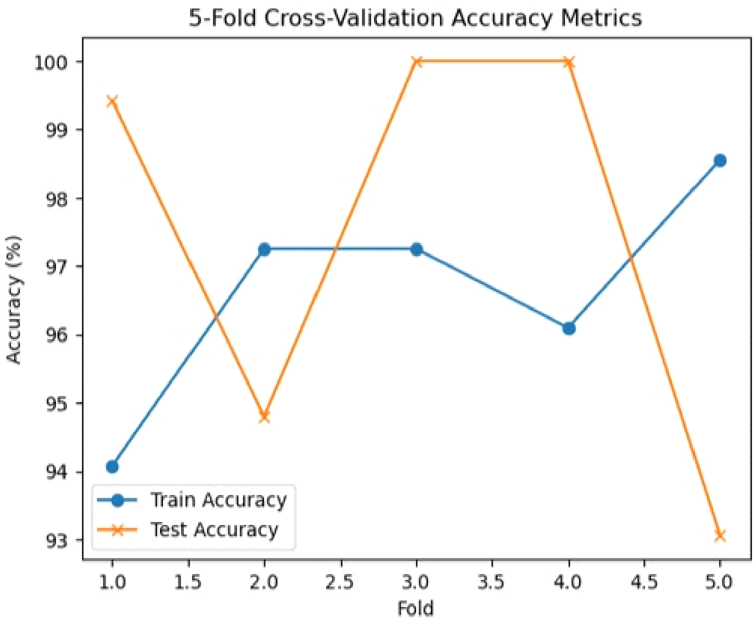

Fig. 4. Training and Testing Accuracy for Each Fold - Tatamotors Dataset

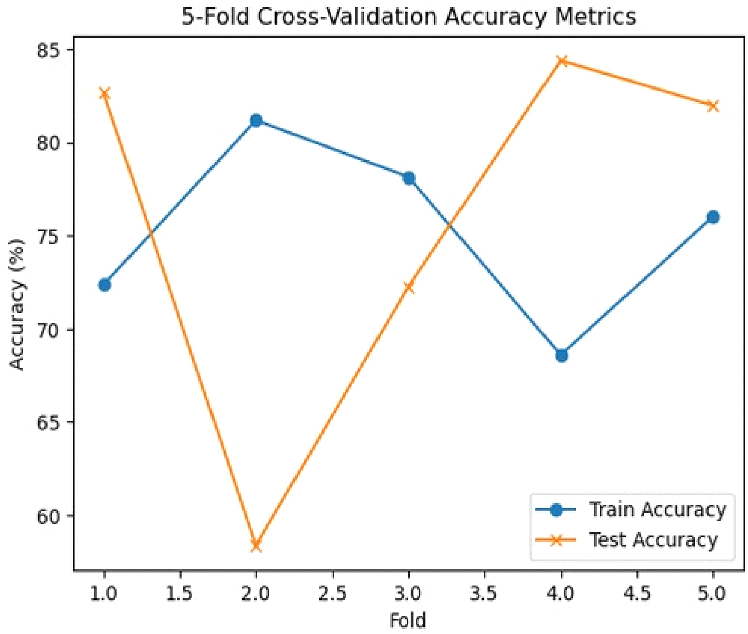

Fig. 5. Training and Testing Accuracy for Each Fold - INFY.NS Dataset

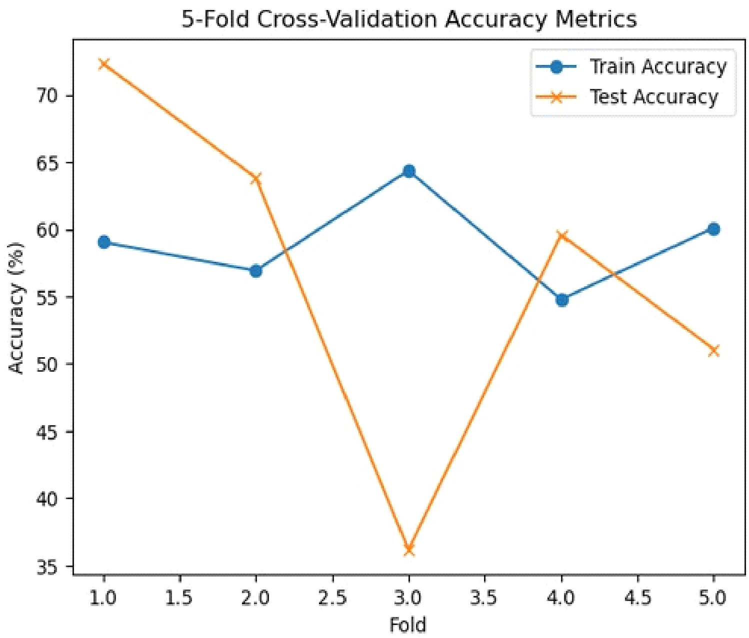

Fig. 6. Training and Testing Accuracy for Each Fold - HDFC Bank Dataset

6. Conclusions and Future Research Directions

This study explores the use of ensemble deep learning models for financial forecasting in stock markets. Advanced models like LSTM, GRU, Bidirectional LSTM, LSTM with Attention, and Convolutional LSTM were employed. The ensemble approach combined predictions from these models to improve accuracy and robustness. Three datasets—Tata Motors, INFY.NS, and HDFC Bank—were used for evaluation. The ensemble model achieved high accuracies: 97.46% (Tata Motors), 75.93% (INFY.NS), and 86.66% (HDFC Bank).GRU and Bidirectional LSTM were the top-performing individual models. LSTM with Attention showed lower performance, possibly due to its inability to fully capture temporal patterns. LSTM with Attention has high training performance but struggles to generalize, as indicated by its significantly higher test MSE and MAPE. This suggests potential overfitting. Conv LSTM achieves a balance between complexity and generalization, performing better than Simple LSTM in most cases.For lower test errors, Bidirectional LSTM and GRU seem to be the best models, as they maintain lower MAPE and RMSE..Conv LSTM is a good middle ground, outperforming Simple LSTM in some cases.

The study highlights the importance of combining models to address diverse financial data complexities. Deep learning models excel in capturing non-linear relationships and long-term dependencies in stock data. Metrics like MSE, RMSE, and MAPE were used to evaluate model performance comprehensively. While our ensemble technique demonstrates improved forecasting accuracy, its performance may be influenced by the degree of market efficiency and the presence of behavioral biases. Future research could explore the relationship between market efficiency and the predictive power of deep learning models, particularly in different market regimes. Factors like transaction costs, slippage, and liquidity constraints were not considered. Future research should incorporate these real-world factors for a more realistic evaluation. Hybrid models combining deep learning with reinforcement learning or fuzzy logic could enhance performance. Integrating external data sources like news sentiment and macroeconomic indicators may improve predictions. The study demonstrates the potential of deep learning in financial forecasting and risk managements. Future work should explore the impact of market sentiment and investor behaviour on predictions. In conclusion, ensemble deep learning models offer a powerful tool for enhancing financial forecasting and decision-making.

References

[1] Abu-Mostafa, Y.S., Atiya, A.F.: Introduction to financial forecasting. Applied intelligence 6, 205–213 (1996)
[2] Samonas, M.: Financial Forecasting, Analysis, and Modelling: a Framework for Long-term Forecasting. John Wiley & Sons, (2015)
[3] Wasserbacher, H., Spindler, M.: Machine learning for financial forecasting, planning and analysis: recent developments and pitfalls. Digital Finance 4(1), 63–88 (2022)
[4] Kingdon, J.: Intelligent Systems and Financial Forecasting. Springer, (2012)
[5] 5Enyindah, P., Onwuachu Uzochukwu, C.: A neural network approach to financial forecasting. International Journal of Computer Applications 135(8), 28–32 (2016)
[6] Tripathy, N.: Forecasting gold price with auto regressive integrated moving average model. International Journal of Economics and Financial Issues 7(4), 324–329 (2017)
[7] Vuong, P.H., Dat, T.T., Mai, T.K., Uyen, P.H., et al.: Stock-price forecasting based on xgboost and lstm. Computer Systems Science & Engineering 40(1) (2022)
[8] Kavzoglu, T., Teke, A.: Predictive performances of ensemble machine learning algorithms in land- slide susceptibility mapping using random forest, extreme gradient boosting (xgboost) and natural gradient boosting (ngboost). Arabian Journal for Science and Engineering 47(6), 7367–7385 (2022)
[9] Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied soft computing 90, 106181 (2020)
[10] Khalil, F., Pipa, G.: Is deep-learning and natural language processing transcending the financial forecasting? investigation through lens of news analytic process. Computational Economics 60(1), 147–171 (2022)
[11] Cao, J., Li, Z., Li, J.: Financial time series forecasting model based on ceemdan and lstm. Physica A: Statistical mechanics and its applications 519, 127–139 (2019)
[12] Siami-Namini, S., Tavakoli, N., Namin, A.S.: A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. arXiv preprint arXiv:1911.09512 (2019)
[13] Schaffer, A.L., Dobbins, T.A., Pearson, S.-A.: Interrupted time series analysis using autoregressive integrated moving average (arima) models: a guide for evaluating large-scale health interventions. BMC medical research methodology 21, 1–12 (2021)
[14] Wang, L., Ma, F., Liu, J., Yang, L.: Forecasting stock price volatility: New evidence from the garch-midas model. International Journal of Forecasting 36(2), 684–694 (2020)
[15] Dai, X., Cerqueti, R., Wang, Q., Xiao, L.: Volatility forecasting: a new garch-type model for fuzzy sets-valued time series. Annals of Operations Research, 1–41 (2023)
[16] Shmueli, G., Polak, J.: Practical Time Series Forecasting with R: A Hands-on Guide. Axelrod schnall publishers, (2024)
[17] Kurani, A., Doshi, P., Vakharia, A., Shah, M.: 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 (2023)
[18] Chhajer, P., Shah, M., Kshirsagar, A.: The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. Decision Analytics Journal 2, 100015 (2022)
[19] Zhong, K., Ma, J., Han, M.: Online prediction of noisy time series: Dynamic adaptive sparse kernel recursive least squares from sparse and adaptive tracking perspective. Engineering Applications of Artificial Intelligence 91, 103547 (2020)
[20] Han, M., Zhang, S., Xu, M., Qiu, T., Wang, N.: Multivariate chaotic time series online prediction based on improved kernel recursive least squares algorithm. IEEE transactions on cybernetics 49(4), 1160–1172 (2018)
[21] Cao, L.-J., Tay, F.E.H.: Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on neural networks 14(6), 1506–1518 (2003)
[22] Pradeepkumar, D., Ravi, V.: Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network. Applied Soft Computing 58, 35–52 (2017)
[23] Alameer, Z., Abd Elaziz, M., Ewees, A.A., Ye, H., Jianhua, Z.: Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm. Resources Policy 61, 250–260 (2019)
[24] Chen, Y., Zhang, G.: Exchange rates determination based on genetic algorithms using mendel's principles: Investigation and estimation under uncertainty. Information fusion 14(3), 327–333 (2013)
[25] Ang, K.K., Quek, C.: Stock trading using rspop: A novel rough set-based neuro-fuzzy approach. IEEE transactions on neural networks 17(5), 1301–1315 (2006)
[26] Chang, P.-C., Fan, C.-Y.: A hybrid system integrating a wavelet and tsk fuzzy rules for stock price forecasting. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38(6), 802–815 (2008)

[27] Li, C., Chiang, T.-W.: Complex neurofuzzy arima forecasting—a new approach using complex fuzzy sets. IEEE Transactions on Fuzzy Systems 21(3), 567–584 (2012)
[28] Jilani, T.A., Burney, S.M.A.: A refined fuzzy time series model for stock market forecasting. Physica A: Statistical Mechanics and its Applications 387(12), 2857–2862 (2008)
[29] Wei, L.-Y., Chen, T.-L., Ho, T.-H.: A hybrid model based on adaptive-network-based fuzzy inference system to forecast taiwan stock market. Expert Systems with Applications 38(11), 13625–13631 (2011)
[30] Sun, B., Guo, H., Karimi, H.R., Ge, Y., Xiong, S.: Prediction of stock index futures prices based on fuzzy sets andmultivariate fuzzy time series. Neurocomputing 151, 1528–1536 (2015)
[31] Rubio, A., Bermúdez, J.D., Vercher, E.: Improving stock index forecasts by using a new weighted fuzzy-trend time seriesmethod. Expert Systems with Applications 76, 12–20 (2017)
[32] Nguyen, L., Novák, V.: Forecasting seasonal time series based on fuzzy techniques. Fuzzy Sets and Systems 361, 114129 (2019)
[33] Novák, V., Perfilieva, I., Dvorak, A.: Insight Into Fuzzy Modeling. John Wiley & Sons, (2016)
[34] Yamak, Peter T., Li Yujian, and Pius K. Gadosey. "A comparison between arima, lstm, and gru for time series forecasting." In Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence, pp. 49-55. 2019.
[35] Gao, Ya, Rong Wang, and Enmin Zhou. "Stock prediction based on optimized LSTM and GRU models." Scientific Programming 2021,
[36] Shiller, R. J.. Efficient Markets Theory to Behavioral Finance. Journal of Economic Perspectives. (2003)

Authors' Profiles

Chandrayani Rokde holds a master's degree in Computer Science Engineering & Technology from Yeshwantrao Chavan College of Engineering Nagpur, India.Persuing her Ph.D. from Department of Computer Science & Engineering Indian Institute of Information Technology, Nagpur. Her research interests include Machine Learning, Bioinformatics and Parallel Computing. She is a dedicated researcher with a keen interest in developing innovative solutions.

Jagdish Chakole is working as Assistant Professor in Department of Computer Science & Engineering Indian Institute of Information Technology, Nagpur, India .He had completed his Ph.D. in Computer Science and Engineering from VNIT Nagpur (2022) .His area of interest are Machine Learning and Quantitative Finance.

Aishwary Ukey working as Assistant Professor in Department of Computer Science & Engineering Indian Institute of Information technology ,Nagpur, India . He had completed his Ph.D. in Computer Science and Engineering from MANIT, Bhopal (2019).His area of research are Cognitive Radio Networks, Wireless and Ad hoc Networks, Internet of Things and Computer and Network Security.

How to cite this paper: Chandrayani Rokde, Jagdish Chakole, Aishwarya Ukey, "Financial Forecasting with Deep Learning Models Based Ensemble Technique in Stock Market Analysis", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.17, No.4, pp. 1-13, 2025. DOI:10.5815/ijieeb.2025.04.01