Multi-class stock market forecasting with deep learning models: an explainable artificial intelligence

Chhaya Patel[1], Ashwin Raiyani[2]
[1]Department of Computer Engineering, School of Engineering, RK University, Rajkot, India
[2]Institute of Management, Nirma University, Ahmedabad, India

Article Info

Article history:

Received Mar 12, 2025
Revised Aug 2, 2025
Accepted Sep 7, 2025

Keywords:

Convolutional neural networks
Deep learning
Explainable artificial intelligence
Long short-term memory
Stock prediction
Technical indicators

ABSTRACT

In this research, we investigated the influence of different deep learning techniques on time series stock market data, especially for all Nifty50 companies in the Indian stock market. Our proposed method of stock market prediction focused on multi-class classification with explainable artificial intelligence (XAI). Our proposed model incorporates convolutional neural network (CNN) for operational feature extraction and long short-term memory (LSTM) to capture time-based dependencies. Predicted value is classified with multiclass classes-very bullish, bullish, neutral, bearish, very bearish signals for all Nifty50 stocks. The model integrates essential technical indicators to find patterns from basic price data. XAI techniques are also used to find feature contributions to model prediction. It improves the clarity of the model's administrative procedure by figuring out how technical indicators influence stock estimates. The outcomes highlight the model's ability to generate actionable trading signals, reinforced by performance progress metrics, contributing to more well-informed and planned venture decisions. The proposed model reveals greater performance, reaching an average accuracy of 96%, beating LightGBM at 89%, random forest at 85%, and support vector machine at 60%.

Corresponding Author:

Ashwin Raiyani
Institute of Management, Nirma University
Ahmedabad, Gujarat, India
Email: ashwin.rkcet@gmail.com

1. INTRODUCTION

Predicting the short-term future value of particular stock using different systematic approaches is critical for short term trading, which comprises rigorous historical data analysis and pattern finding inside data to produce profits or minimize loss. Brokers can buy and sell shares on the stock market to make money. Predicting the accurate position to buy, sell, or hold might help brokers generate money from the market or minimize loss. Exact price movements lead to big profits; so many researchers are interested in this area. Traditional methodologies, including as linear regression, exponential averaging, autoregressive integrated moving average (ARIMA), and generalized autoregressive conditional heteroskedasticity (GARCH), have been applied for the purpose of predicting stock prices for a considerable amount of time. However, these approaches are limited to linear patterns and assume that the data use a normal distribution. In recent years, the domain of financial forecasting has been influenced by advanced machine learning and deep learning models. These models were developed to overcome the limitations of traditional linear techniques, leading to significant improvements in forecasting processes. Both types of models are capable of handling the non-linear nature of financial data [1], [2]. So here in our research we tried to use different deep learning and machine learning methods for stock market prediction.

Basic information of stock is day open price, day close price, day high price, day low price, and volume. Different types of technical indicators are used to find underlying patterns of stock movement. Based on their roles, these indicators may be divided into numerous categories: volume indicators, momentum indicators, volatility indicators, and trend-following indicators. The most prominent technical indicators include moving average convergence divergence (MACD)-trend-following indicator, relative strength index (RSI)-momentum indicator, average true range (ATR)-volatility indicator, volume-weighted average price (VWAP)-cumulative indicator and rate of change (ROC)-momentum oscillator [3], [4].

Long short-term memory (LSTM) offers a measurable approach to stock market forecasting as it is widely used for time series forecasting like stock data. LSTM model, along with other neural network models, persist a noticeable focus of research and development in time series forecasting. Improvements in LSTM with optimization parameters have the potential to raise stock market prediction correctness. This is vital in stock market forecast, as it allows stockholders and traders to measure the consistency of the model's forecasts, helping them make up-to-date decisions [5], [6]. Convolutional neural networks (CNNs) work by extending over convolutional filters to input data, perceiving native configurations and associations. Each layer of convolutional abstracts expressive structures such as price points, movements, or relationships between technical indicators by scanning filters across the time series data. The pooling layers then reduce the dimensions, protecting the most important features while inhibiting overfitting due to the curse of dimensionality [7]. This research is motivated by the improvement of stock movement prediction by using both price and volume based technical indicators that replicate market momentum and direction. In this study, we recommend a hybrid model that combines CNNs for feature extraction and LSTM for consecutive learning.

Although deep learning models like CNNs and LSTMs deal with high predictive accuracy, their black-box tendency poses an important encounter in financial applications. Financial analysts need not only detailed estimates but also obvious clarifications for model conclusions. Explainable artificial intelligence (XAI) aims to associate this gap by providing visions into how models reach their predictions. In this study, we applied XAI methods: local interpretable model-agnostic explanations (LIME) and Shapley additive explanations (SHAP). By incorporating XAI techniques, the study improves the model's transparency and fosters trust among traders and stakeholders [8]–[10].

Stock market predicting has become a prominent research area in the financial domain and is now vital part of artificial intelligence. Researchers are using numerous statistical measurements, machine learning algorithms, and various deep learning algorithms to find an underlying pattern from stock data. The related works based on the given topics are discussed here with Table 1.

The existing works noted in Table 1. Primarily focus on binary classification or next-day closing price, which is not satisfactory for capturing complex dynamics and particulars of financial markets. Traditional machine learning algorithms like support vector machine (SVM), k-nearest neighbors (kNN), decision tree (DT), random forest (RF), and AdaBoost are not well generalized with multi-class classification due to a lack of capability to handle nonlinear datasets [2], [11]. Current research in deep learning, such as LSTM, recurrent neural networks (RNNs), have exposed noteworthy prospects in time-series prediction by excellently catching chronological dependencies within the data [5].

Table 1. The correspondence of our proposed model with literature

Reference	Datasets	Duration	Methods	Evaluation metrics	Incorporation with XAI
[1]	S&P 500(33 companies)	2017-2021	MLP	MAE, R^2	No
[6]	Chinese companies	2019-2023	LINE, LSTM	MSE, RMSE, MAP, R^2	No
[7]	Nifty50 index	2014-2018	RNN, CNN, LSTM	MSE, RMSE, MAP, R^2	No
[8]	DAX30, FTSE100 S&P500, Nikkie225	1990-2022	DNN	Accuracy, F1-score, precision, recall	Yes
[9]	S&P500 health care price index	2017-2019	LSTM	MSE, RMSE	Yes
[11]	NASDAQ, NYSE, FTSE, NIKKEI	2010-2020	RF, SVM, kNN, ANN,	Accuracy, F1-score, precision, recall	No
[12]	Nifty50, Sensex, S&P500	2015-2021	PSO-LSTM	MSE, RMSE, MAP	No
[13]	Chinese stock market	2018-2019	PCA-LSTM	Accuracy, F1-score	No
[14]	5 Indian companies	2007-2017	MLP, RNN, LSTM, CNN	MAPE	No
[15]	Pingan Bank	2016-2018	PCA-LSTM	RMSE, MAPE	No
[16]	Financial news data	2012-2016	LR, SVR, ANN	RMSE, MAE	No
[17]	NASDAQ-100 index	2020-2021	PSO-ANN	MSE, RMSE, R^2	No
[18]	Nifty50 12-companies	2015-2021	kNN, SVM, DT, LSTM	MSE, RMSE, MAP, R^2	No
Proposed model	Nifty50 all companies	2020-2024	CNN-LSTM	Accuracy, F1-score, precision, recall	Yes

Our proposed model focuses on multi-class classification with classes: very bullish, bullish, neutral, bearish, and very bearish signals for all Nifty50 stocks. As a dataset, we tried to focus on all 50 companies from the Nifty50. To capture an underlying pattern from a given dataset, we integrate different technical indicators such as RSI, MACD, ATR, ROC, VWAP, and many more [6]. Our model identifies the best combination of technical indicators and generates 5 different classes for prediction. A CNN-LSTM fusion model developed for this classification [7]. To enhance prediction capability and understanding of models' behavior, we incorporate the XAI technique SHAP and LIME, which highlight feature influence and interpretability. These methods help us to understand the importance of input features by providing both global and local interpretability. This level of interpretability is crucial for building trust among stakeholders and supporting more informed and confident decision-making. [8], [19]. Most existing models focus on prediction accuracy but lack model transparency, which is very important to understand and trust model. Our research addresses this gap by incorporating CNN-LSTM model with XAI tools, providing both accuracy and interpretability.

2. METHOD

In this research paper, we used different technical indicators to uncover hidden patterns inside basic price data. The addition of technical indicators led to a curse of dimensionality, which may negatively influence model performance. To avoid this issue, CNN was applied to solve the curse of dimensionality. After feature selection, the LSTM model was applied classify stock market trends. We incorporated XAI-SHAP and LIME, to highlight feature impact and improve interpretability.

2.1. Nifty50 companies

The Nifty50 is a major Indian stock market index that represents the performance of the 50 most prominent and actively traded companies on the national stock exchange (NSE). It is a standard index extensively used by shareholders and researchers to measure the performance of the Indian equity market. It includes corporations from different sectors like finance, consumer goods, technology, pharmaceuticals, energy, as shown in Figure 1. Company selection is done based on market capitalization, liquidity, trading frequency, and index rebalancing [13].

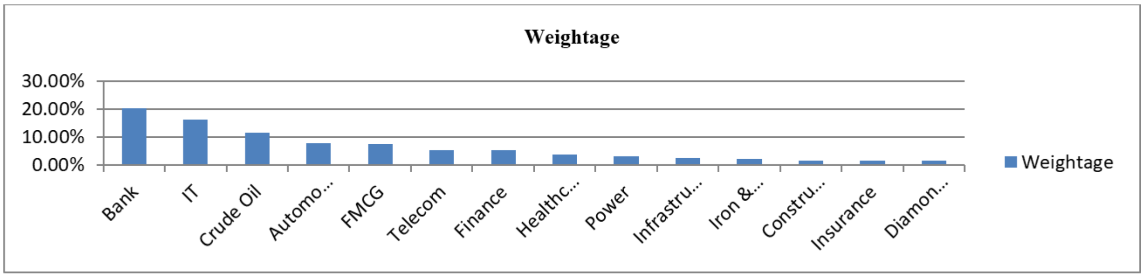

Figure 1. Sector-wise weightage in Nifty50

2.2. Technical indicators

Market technical indicators are quantitative tools that used to identify trends associated with specific stocks. Traders in the market utilize these trends to forecast price fluctuations of stocks. Indicators rely on fundamental stock prices, including the volume of stocks, lowest price, opening price, closing price, and highest price. They offer critical insights and reveal patterns within the data. By employing these indicators, brokers understand the overall market strength and the company’s performance. Our research has focused on calculating the most influential technical indicators, which include the simple moving average (SMA), RSI, MACD, ATR, ROC, and VWAP [20]. The SMA is considered by adding the closing prices of a stock over a listed number of days and then dividing that total by the number of days as shown in (1). For instance, to determine the 20-day SMA of a stock's closing prices, one would sum the closing prices from the last 20 days and divide the sum by 20 [5].

$$SMA = \frac{Sum\ of\ Closing\ Prices\ for\ n\ periods}{n} \tag{1}$$

The RSI is a technical indicator working to measure the speed and price actions in stocks. This indicator works on the principle that prices usually change inside a specific range under varying market circumstances. It helps in knowing overbought and oversold situations, in addition to potential reversals in trends. The calculation of the RSI can be performed using the RS value as calculated in (2) [3]. RS value is generated based on the average gain and average loss of a specific period as given with (3). With the help of (4) and (5), the average price gain and average price loss over a certain time period are used to compute the average gain and average loss.

$$RSI = 100 - \frac{100}{1+RS} \tag{2}$$

$$RS = \frac{Average\ Gain}{Average\ Loss} \tag{3}$$

$$Average\ Gain = \frac{Sum\ of\ the\ Price\ Gains\ over\ the\ specified\ period}{Number\ of\ periods} \tag{4}$$

$$Average\ Loss = \frac{Sum\ of\ the\ Price\ Losses\ over\ the\ specified\ period}{Number\ of\ periods} \tag{5}$$

Price gain denotes the change between the closing prices of the current period and the previous period. In contrast, price loss shows the difference between the closing prices of the previous period and the current period. The RSI is resulting from the average of recent price gains and losses over a chosen timeframe. The usually utilized period for RSI design is 14 days [3].

The MACD is a widely used technical examination tool used in financial markets to classify possible changes in momentum, trend reversals, and signs for buying or selling. The MACD is derived from two exponential moving averages (EMAs) of stock prices, exactly the 26-day EMA and the 12-day EMA. The MACD line is calculated by subtracting the 26-day EMA from the 12-day EMA. Analysts and traders often employ the MACD together with other technical indicators to improve their trading decisions [15].

The ATR is extensively recognized as a measure of volatility. It assists traders in assessing the volatility of a stock. A higher ATR indicates that the stock is more volatile, while a lower ATR signifies reduced volatility. The calculation of ATR utilizes the current period's high, the current period's low, and the closing price from the previous day as using (6) [3].

$$ATR = \frac{Previous\ ATR(n-1)+TR}{Number\ of\ periods} \tag{6}$$

The Bollinger bands work as a volatility indicator consisting of three lines: the middle band, which is the 20-day SMA specified as (7); the higher band, calculated as the middle band plus two times the standard deviation (SD) specified as (8); and the lower band, determined by subtracting two times the SD from the middle band as specified (9). The 20-day Bollinger band is calculated accordingly [3].

$$Middle\ Band = SMA(20\ days) \tag{7}$$

$$Higher\ Band = Middle\ Band + (2 \times SD\ over\ 20\ days) \tag{8}$$

$$Lower\ Band = Middle\ Band - \ (2 \times SD\ over\ 20\ days) \tag{9}$$

2.3. Dimensionality reduction with convolutional neural network

In stock market prediction, we frequently have a large number of features, including basic prices and different technical indicators. Additionally, we have data from different companies. High-dimensional data creates the curse of dimensionality, triggering overfitting in the model. Different machine learning/deep learning models are used to reduce the curse of dimensionality, like principal component analysis (PCA), autoencoders, and CNNs. CNNs are supposed to cut this high-dimensional data into lower-dimensional, more descriptive data while conserving important patterns compared with PCA and autoencoders. CNN finds non-linear patterns from given data, as compared to PCA, which works only with linear data. CNN is composed of multiple layers, as illustrated in Figure 2. Convolutional layers are used for feature extraction. Pooling layers are used for dimensionality reduction. Non-linearity is introduced using the activation function rectified linear unit (ReLU). We can apply multiple convolutional and pooling layers to recognize patterns from given data. The final output is applied to the flattened layer that converts to a 1D vector [16], [21].

2.4. Temporal modeling with long short term memory

LSTM is a specialized architecture of RNNs that excels in dealing with time series data, making it mainly useful in stock price forecasting. LSTM networks are engineered to efficiently capture long-term dependencies, demonstrating to be particularly helpful once examining time series data, including past stock prices. The design of an LSTM model specifically addresses the problems associated with learning and retaining long-term dependencies within sequential datasets. Associated to conventional RNNs, LSTMs possess a more complicated structure, interpreting them highly effective for applications that require the accepting of long-term dependencies. An outline of the main elements that make up an LSTM network's structure is provided with Figure 3 [9], [21]. The cell state input (Ct_0) works as the long-term memory within the LSTM architecture. It processes the whole structure, permitting the addition or elimination of data through specifically designed mechanisms known as gates. In contrast, the hidden state input (Ht_0) represents the short-term memory, formed by both the current input and the cell state. LSTMs employ three gates to accomplish the direction of data flows: input gate, output gate, and forget gate. The input gate handles which data passed in the cell state, while the forget gate allow to reject information. The output gate normalizes the following hidden state based on the adapted cell state [10], [21].

2.5. Explainable artificial intelligence

XAI is very important for AI driven models as it increases transparency and interpretability. It allows us to establish confidence and trust in complex black-box AI models. Given that stock prediction models commonly exploit intricate machine learning and deep learning algorithms, it is significant for traders, investors, and financial analysts to understand the decision-making processes behind these models. Different XAI methods, such as SHAP, LIME, and gradient-weighted class activation mapping (Grad-CAM), are used to identify important features in AI models [22], [23].

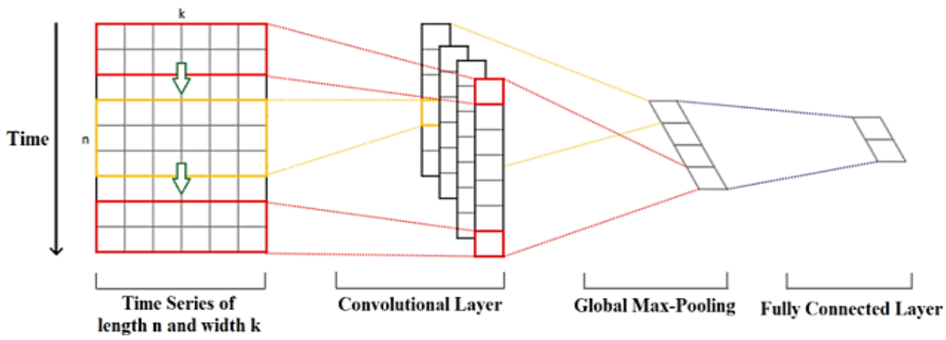

Figure 2. Different layers of CNN

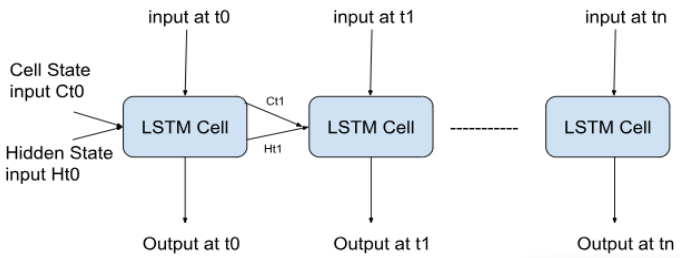

Figure 3. LSTM network's structure

3. PROPOSED MODEL

A summary of the proposed fusion model, incorporating LSTM with CNN, is presented here along with Figure 4. The proposed model takes 5 years of open, high, low, and close (OHLC) raw data from all Nifty50 companies as input. Feature selection techniques are applied to extract key indicators relevant to market movements. After data cleaning, target signals are generated to train model. The selected features are then fed into the LSTM-CNN fusion model for effective market prediction.

3.1. Data collection

The dataset for our research work is the Nifty50 all companies' historical information, including all basic OHLC prices and volume. The period of historical data is 5 years, from January 1, 2020 to December 31, 2024, so for one company, stock data rows are about 1,206; hence, for a total of 50 companies, collected rows are about 57,105. All companies' data are collected using the yfinance API [2], [17].

3.2. Feature engineering

To predict future price actions, analysts operate technical indicators, which are methods for examining previous price and volume data of a company. These indicators, which are derived from mathematical calculations, can assist traders in making knowledgeable decisions concerning the buying/selling of particular stocks. TA-Lib library is used to calculate technical indicators for stocks. Our research proposed the calculation of the most affected technical indicators, which are RSI, MACD, SMAs, ATR, ROC, and VWAP [11].

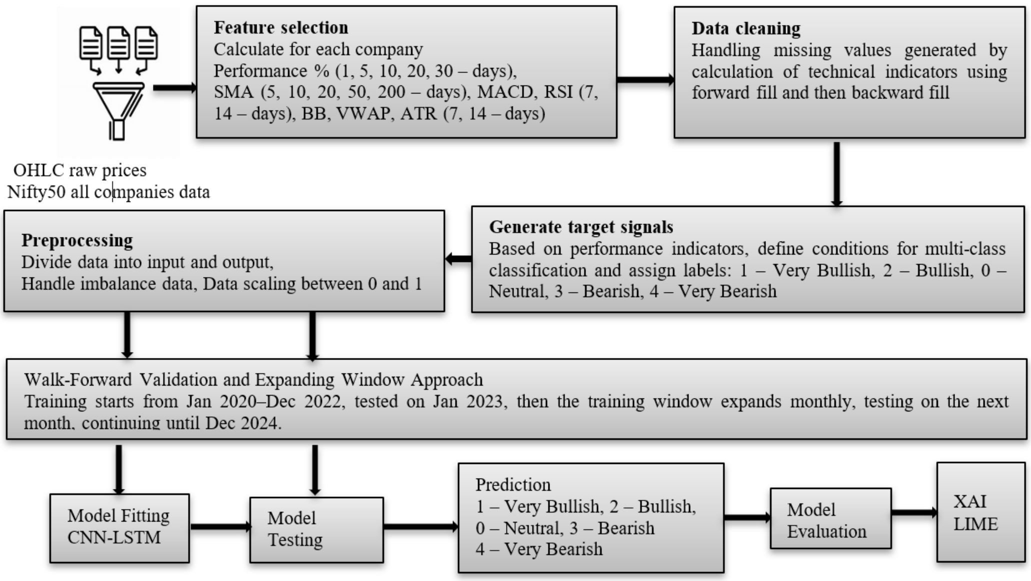

Figure 4. Proposed CNN-LSTM model

3.3. Exploratory data analysis

After the calculation of technical indicators, we now had 27 features or columns to be considered for data exploration. SMA_20 days will generate NaN value for the first 20 days of a particular company, the same way RSI_14 will generate NaN for the first 14 days. Based on their mathematical formulations, the applied technical indicators inherently generate NaN values for the initial periods. To make data ready for CNN, we filled the NaN value by using forward and then backward methods.

3.4. Generate target classes based on performance indicators

As we implemented multi-class classification, we had to generate a total of 5 classes based on performance indicators. We calculated the average of the percentage returns across 1, 5, 10, 20, and 30 days for each row in the dataset, and we saved the outcome in the "Average_Return" column as described (10). Based on Average_Return, we classify each row into one of five classes described in Table 2.

$$Average\_Return = \frac{1D+5D+10D+20D+30D}{5} \tag{10}$$

Table 2. Generate target classes based on average_return indicators

Condition	Target label	Meaning
Average_Return >5	Class 1	Very bullish
2 ≤Average_Return ≤5	Class 2	Bullish
-2 <Average_Return <2	Class 0	Neutral
-5 ≤Average_Return ≤-2	Class 3	Bearish
Average_Return <-5	Class 4	Very bearish

3.5. Data preprocessing

As part of data preprocessing, the data is distributed into features and target variables. Technical indicators are used as features, while the target consists of five different classes representing various stock price activities. Data scaling is vital to standardize the features, guaranteeing they fall inside a precise series, which helps increase the performance and convergence of deep learning prototypes.

3.6. CNN-LSTM model architecture

Our proposed model integrates a combination of the CNN with LSTM architecture to enhance the competency of learning both spatial and temporal features from time series stock data. First, we applied convolutional layer to find an underlying pattern from technical indicators. Then the maxpooling layer is applied to reduce dimensionality by absorbing the max value from each filter's output. After CNN, the LSTM model was applied to capture temporal dependencies over time. To ensure the model does not suffer from overfitting, regularization is used during the training phase. Then flattened layer followed dense layer applied. Finally, the output layer with the SoftMax activation function applied for multiclass classification. To make comparisons between existing models, we selected SVM, RF, and LightGBM as base models. Details of model architecture and training configurations are given with Table 3.

Table 3. CNN-LSTM model architecture parameters

Layer/Task	Parameters
1D convolutional	Time series input derived from technical indicators
MaxPooling1D	64 filters, kernel size=3, activation=ReLU, Pool size=2, used to reduce dimensions
LSTM	50 units, return_sequences=True, captures sequential dependencies
Dropout	Dropout rate=0.2, used to prevent overfitting
Flatten	Flattens 3D output into 1D for the dense layer
Dense	Fully connected layer, activation=ReLU
Output layer	Dense layer with SoftMax activation for multiclass classification
Compilation	Optimizer: Adam; Loss: categorical crossentropy
Training config.	Epochs: 50; Batch size: 32;
Validation strategy	Walk-forward validation (24 Iterations)
Training window	Expanding time window from Jan 2020 to current test month
Testing window	Immediate next month (e.g., Jan 2023, Feb 2023, ..., Jan 2025)
Total iterations	24 walk-forward splits per company (1,200 total iterations (24 splits×50 companies))
Evaluation metrics	Accuracy, precision, recall, F1-score, Macro-average AUC, ROC curve analysis

3.7. Explainable artificial intelligence integration for interpretability

This research paper used XAI techniques for stock market prediction. We applied LIME and SHAP. It makes model transparent and understandable especially complex model like CNN-LSTM model. AI models performances like black-box, they give good performance but it is unclear about how they reached to that decision. XAI open the black box and help traders to understand and trust output of model. LIME focus on local explanation while SHAP gives both local and global inside data. It gives contribution of each technical indicator in prediction. By using them, we bridge the gap between model performance and interpretability, guaranteeing AI-driven financial predictions is both accurate and transparent [24]–[26].

3.8. Model evaluation

As principal metrics for classification, we calculated confusion matrix and from it we derived accuracy, precision, recall, F1-score, ROC curve, and AUC value. Accuracy dealings the proportion of correctly classified instances (both true positive and true negative) over the total number of instances. Precision calculates the proportion of properly predicted positive instances out of all instances predicted as positive. Recall reflects the proportion of actual positive instances that were properly recognized by the model. F1-score is the harmonic mean of precision and recall. Additionally, we present confusion matrices and ROC curves to further evaluate the model's predictive capabilities. For formula refer Table 4.

Table 4. Model evaluation metrics

Metrics	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FN}$
Recall	$\frac{TP}{TP + FP}$
F1-score	$2 \times \frac{Precision - Recall}{Precision + Recall}$

4. RESULT ANALYSIS

We weighed proposed model and benchmark models performance using accuracy, precision, recall, F1-score, AUC, and ROC curve. We executed walk-forward validation (24 iterations) outcomes to statistically relate the proposed model with base models. Additionally, distinguishing the rising attention in model interpretability, we used LIME methods to examine the model's decision-making foundation as well.

4.1. Performance evaluation of CNN-LSTM model

The performance of the proposed model was evaluated against benchmark models (SVM, RF, and LightGBM) using classification metrics accuracy, precision, recall, F1-score, and macro-average AUC, averaged across the WFV strategies. The RF model and LightGBM achieves outstanding performance due to its collaborative approach, while SVM which have simpler structures, show less impressive predictive abilities. Our anticipated model beats existing models, accomplishing the highest accuracy. A comprehensive comparison of the models is presented in Table 5.

Table 5. Comparison of accuracy, precision, recall, F1-score and AUC

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Macro-average AUC (%)
SVM	65	62	64	66	51
RF	89	95	92	89	91
LightGBM	89	94	91	88	92
Proposed model	97	96	96	97	96

4.2. Statistical significance testing

To make sure the reliability of the performance improvements, we carried out paired t-tests comparing the proposed CNN-LSTM model with base models across the outcomes of walk forward validation. As given in Table 6, all comparisons produced p-values under 0.05. This confirms that the improvements of the proposed model are statistically significant and not due to random chance.

Table 6. Statistical significance testing

Comparison	Metric	t-Statistic	p-Value	Significance ((p <0.05)?)
Proposed model vs. SVM	Accuracy	86.06	0.00000	Yes
	Precision	51.01	0.00000	Yes
	Recall	64.91	0.00000	Yes
	F1-score	82.32	0.00000	Yes
Proposed model vs. RF	Accuracy	20.85	0.00003	Yes
	Precision	4.00	0.01613	Yes
	Recall	21.00	0.00003	Yes
	F1-score	21.92	0.00003	Yes
Proposed model vs. LightGBM	Accuracy	25.30	0.00001	Yes
	Precision	3.14	0.03492	Yes
	Recall	∞	0.00000	Yes
	F1-score	28.46	0.00001	Yes

t=∞ means no variance in the difference values

4.3. CNN-LSTM ROC curve

The ROC curve is illustration of the true positive rate and the false positive. The ROC curve helps to picture how well a classifier is performing across different classification thresholds. Figure 5 shows ROC curves for proposed multi-class model.

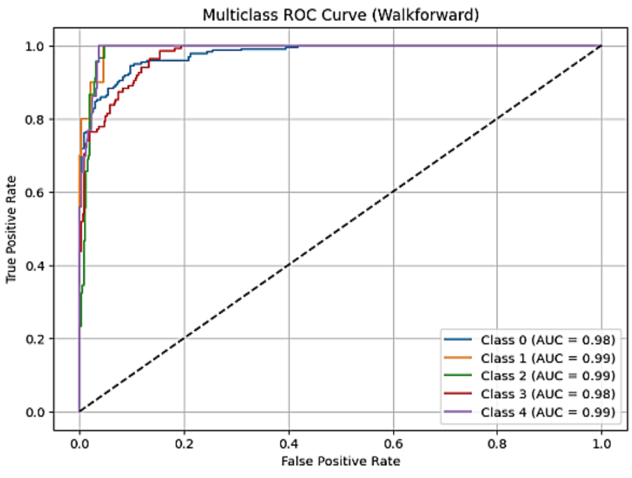

Figure 5. ROC curves for proposed model

4.4. XAI-feature contribution using LIME

We used LIME to understand the predictions of our CNN-LSTM model for stock market analysis using the Nifty50 dataset. LIME offers local interpretability, revealing how key features influence individual predictions. As shown in Figure 6, it suggests traders a strong observation of the model's decision-making process. The outcomes show that 20_Days (%), SMA_5, 30_Days (%), and SMA_10 play the maximum important roles in determining the model's resolutions. This proposes that the model seriously trusts on short-term price volatility when making predictions.

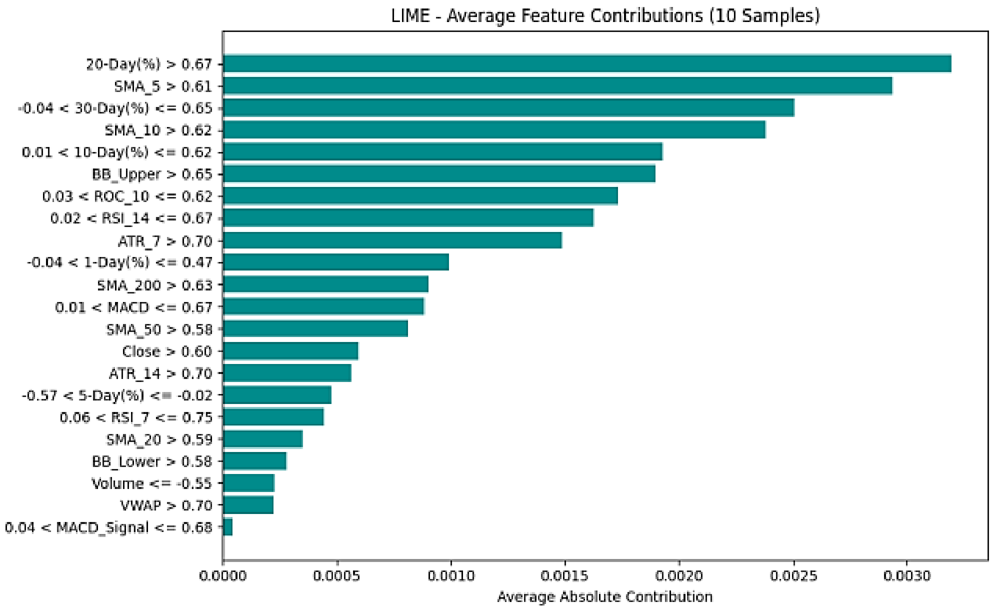

Figure 6. Feature contribution using LIME

4.5. Backtesting of model with unseen data

To evaluate the real-world usefulness of the proposed model, backtesting was performed using unseen data from January 1 to 31, 2025. We compared predictions-ranging from very bullish to very bearish-with actual outcomes. Financial indicators such as return on investment (ROI), measured between 10% and 15%, showed that the model was profitable and robust enough to be used for short-term stock predictions in real-world trading situations.

5. CONCLUSION

In this research work, we implemented CNN-LSTM model for stock market prediction, by using technical indicators such as SMA, BB, ATR, RSI, MACD, and VWAP. The model achieved an accuracy of 96%, demonstrating its effectiveness in catching multilayered patterns and movements in stock price prediction, outperforming traditional methods. To improve transparency, we applied LIME for local interpretability, revealing that short-term volatility indicators like 20-Days%, SMA_5, 10-Days%, SMA_10, RSI14, played the most significant roles in the model's predictions. This work highlights the potential of deep learning and XAI in financial decision-making. By including technical indicators and interpretability methods, our methodology enhances transparency in stock market predictions. Our model will help trader to manage risk and maximize portfolio.

6. FUTURE RECOMMENDATIONS

For future work, we aim to incorporate macroeconomic features and real-time data to further boost analytical performance and applicability in dynamic stock markets. Working with real-time data will allow our model to learn sudden change in market, cultivating decision-making for traders. Furthermore, merging local as well as global news data sentiments perhaps will improve the appropriate accepting of market behavior.

FUNDING INFORMATION

Authors state there is no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Chhaya Patel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Ashwin Raiyani	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓	

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

This study did not involve individuals or any personal identification information that could require any informed consent.

ETHICAL APPROVAL

This paper does not involve people or animals; no investigation has involved human subjects. Therefore, the authors did not seek approval from any institutional review board.

DATA AVAILABILITY

This study utilized historical stock price and volume data for all 50 companies included in the Nifty50 index. The dataset is publicly accessible through the National Stock Exchange of India (NSE) and open financial data platforms such as Yahoo Finance. Researchers can obtain the data directly from the NSE official website at https://www.nseindia.com/ or from Yahoo Finance at https://finance.yahoo.com/. No proprietary or restricted-access data were used in this work.

REFERENCES

[1] G. Sonkavde, D. S. Dharrao, A. M. Bongale, S. T. Deokate, D. Doreswamy, and S. K. Bhat, "Forecasting stock market prices using machine learning and deep learning models: a systematic review, performance analysis and discussion of implications," *International Journal of Financial Studies*, vol. 11, no. 3, 2023, doi: 10.3390/ijfs11030094.

[2] C. Y. Lin and J. A. L. Marques, "Stock market prediction using artificial intelligence: a systematic review of systematic reviews," *Social Sciences and Humanities Open*, vol. 9, 2024, doi: 10.1016/j.ssaho.2024.100864.

[3] M. A. A. Ballesteros and E. A. M. Miranda, "Stock market forecasting using a neural network through fundamental indicators, technical indicators and market sentiment analysis," *Computational Economics*, 2024, doi: 10.1007/s10614-024-10711-4.

[4] V. Zakamulin and J. Giner, "Trend following with momentum versus moving averages: a tale of differences," *Quantitative Finance*, vol. 20, no. 6, pp. 985–1007, 2020, doi: 10.1080/14697688.2020.1716057.

[5] S. Srivastava, M. Pant, and V. Gupta, "Analysis and prediction of Indian stock market: a machine-learning approach," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 4, pp. 1567–1585, 2023, doi: 10.1007/s13198-023-01934-z.

[6] Y. Xu, D. Wang, and J. Hao, "Stock price prediction based on shareholding network topology and LSTM model," *Proceedings of the 2nd International Conference on Public Management, Digital Economy and Internet Technology, ICPDI 2023*, EAI, 2023, doi: 10.4108/eai.1-9-2023.2338695.

[7] Z. Fathali, Z. Kodia, and L. B. Said, "Stock market prediction of Nifty50 index applying machine learning techniques," *Applied Artificial Intelligence*, vol. 36, no. 1, 2022, doi: 10.1080/08839514.2022.2111134.

[8] D. Muhammad, I. Ahmed, K. Naveed, and M. Bendechache, "An explainable deep learning approach for stock market trend prediction," *Heliyon*, vol. 10, no. 21, 2024, doi: 10.1016/j.heliyon.2024.e40095.

[9] P. Kumar, L. Hota, V. A. Tikkiwal, and A. Kumar, "Analysing forecasting of stock prices: an explainable AI Approach," *Procedia Computer Science*, vol. 235, pp. 2009–2016, 2024, doi: 10.1016/j.procs.2024.04.190.

[10] M. T. Hosain, J. R. Jim, M. F. Mridha, and M. M. Kabir, "Explainable AI approaches in deep learning: advancements, applications and challenges," *Computers and Electrical Engineering*, vol. 117, 2024, doi: 10.1016/j.compeleceng.2024.109246.

[11] A. Subasi, F. Amir, K. Bagedo, A. Shams, and A. Sarirete, “Stock market prediction using machine learning,” *Procedia Computer Science*, vol. 194, pp. 173–179, 2021, doi: 10.1016/j.procs.2021.10.071.
[12] G. Kumar, U. P. Singh, and S. Jain, “An adaptive particle swarm optimization-based hybrid long short-term memory model for stock price time series forecasting,” *Soft Computing*, vol. 26, no. 22, pp. 12115–12135, 2022, doi: 10.1007/s00500-022-07451-8.
[13] J. Shen and M. O. Shafiq, “Short-term stock market price trend prediction using a comprehensive deep learning system,” *Journal of Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00333-6.
[14] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “NSE stock market prediction using deep-learning models,” *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018, doi: 10.1016/j.procs.2018.05.050.
[15] Y. Wen, P. Lin, and X. Nie, “Research of stock price prediction based on PCA-LSTM model,” *IOP Conference Series: Materials Science and Engineering*, vol. 790, no. 1, 2020, doi: 10.1088/1757-899X/790/1/012109.
[16] H. Maqsood *et al.*, “A local and global event sentiment based efficient stock exchange forecasting using deep learning,” *International Journal of Information Management*, vol. 50, pp. 432–451, 2020, doi: 10.1016/j.ijinfomgt.2019.07.011.
[17] F. Ghashami, K. Kamyar, and S. A. Riazi, “Prediction of stock market index using a hybrid technique of artificial neural networks and particle swarm optimization,” *Applied Economics and Finance*, vol. 8, no. 3, 2021, doi: 10.11114/aef.v8i3.5195.
[18] M. Bansal, A. Goyal, and A. Choudhary, “Stock market prediction with high accuracy using machine learning techniques,” *Procedia Computer Science*, vol. 215, pp. 247–265, 2022, doi: 10.1016/j.procs.2022.12.028.
[19] J. P. Euko, F. Santos, and P. Novais, “Stock market prediction: integrating explainable AI with Conv2D models for candlestick image analysis,” *Good Practices and New Perspectives in Information Systems and Technologies (WorldCIST 2024)*, Cham, Switzerland: Springer, pp. 13–23, 2024, doi: 10.1007/978-3-031-60218-4_2.
[20] T. Uçkan, “Integrating PCA with deep learning models for stock market forecasting: an analysis of Turkish stocks markets,” *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 8, 2024, doi: 10.1016/j.jksuci.2024.102162.
[21] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*, vol. 2017-January, pp. 1643–1647, 2017, doi: 10.1109/ICACCI.2017.8126078.
[22] K. Letrache and M. Ramdani, “Explainable artificial intelligence: a review and case study on model-agnostic methods,” *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, Casablanca, Morocco, 2023, pp. 1-8, doi: 10.1109/SITA60746.2023.10373722.
[23] J. Černevičienė and A. Kabašinskas, “Explainable artificial intelligence (XAI) in finance: a systematic literature review,” *Artificial Intelligence Review*, vol. 57, no. 8, p. 216, Jul. 2024, doi: 10.1007/s10462-024-10854-8.
[24] H. Bandi, S. Joshi, S. Bhagat, and D. Ambawade, “Integrated technical and sentiment analysis tool for market index movement prediction, comprehensible using XAI,” in *2021 International Conference on Communication information and Computing Technology*, Jun. 2021, vol. 57, no. 8, pp. 1–8, doi: 10.1109/ICCICT50803.2021.9510124.
[25] N. Rane, S. Choudhary, and J. Rane, “Explainable artificial intelligence (XAI) approaches for transparency and accountability in financial decision-making,” *SSRN Electronic Journal*, 2023, doi: 10.2139/ssrn.4640316.
[26] J. Ranjith and S. K. Chandar, “Enhancing stock market trend prediction using explainable artificial intelligence and multi-source data,” *Fusion: Practice and Applications*, vol. 16, no. 2, pp. 178–189, 2024, doi: 10.54216/FPA.160211.

BIOGRAPHIES OF AUTHORS

Prof. Chhaya Patel serves as an Assistant Professor in the Department of Computer Engineering at RK University. She has been pursuing her Ph.D. in Computer Engineering at RK University since 2022. Her research interests include machine learning applications in financial forecasting and intelligent control systems. She can be contacted at email: chhaya.ce@gmail.com.

Dr. Ashwin Raiyani is an Assistant Professor at the Institute of Management, Nirma University. He earned his Ph.D. in Computer Engineering from RK University in 2021. His research interests encompass cloud computing, big data analytics, data science, and machine learning. He has published over 15 peer-reviewed articles in esteemed journals and conferences. He can be contacted at email: ashwin.rkcet@gmail.com.