JSR Publications

Available Online

J. Sci. Res. **17** (1), 195-209 (2025)

JOURNAL OF SCIENTIFIC RESEARCH

www.banglajol.info/index.php/JSR

Examining Daily Closing Price Prediction of the NSE Index using an Optimized Artificial Neural Network: A Study of Stock Market

U. Bashir*, K. Singh, V. Mansotra

Department of Computer Science and IT, University of Jammu, Jammu-180006, J & K, India

Received 4 July 2024, accepted in final revised form 15 October 2024

Abstract

Prediction of the stock market is considered a challenging task because of non-linear and speculative nature of data. Stock prediction means an attempt to forecast the future of a stock or any other financial instrument listed on any stock exchange. The success of stock prediction returns a significant profit for investors and daily traders. Deep learning models have proven to be a reliable option for developing successful prediction systems. In recent years, the use of hyperparameter optimization techniques for the creation of precise models has grown significantly. In this study, an attempt has been made to propose an optimized Artificial Neural Network for predicting the daily closing price of the NIFTY-50, an Index of the National Stock Exchange. For this purpose, the novel dataset has been generated using openly available financial data, and two technical indicators have been used to predict the intraday closing price of the index. To evaluate the performance of the proposed model, a standard percentage-based method known as MAPE, R^2, and RMSE has been applied. After evaluation of the model, lower values of MAPE and RMSE have been achieved which depicts that the model is efficiently predicting the stock closing price.

Keywords: Artificial Neural Network; Backpropagation; Hyperparameters; NIFTY 50; Moving average; Relative Strength index.

doi: https://dx.doi.org/10.3329/jsr.v17i1.74640 J. Sci. Res. **17** (1), 195-209 (2025)

1. Introduction

The stock market is considered non-linear, dynamic, and unpredictable. It is a place where securities of publicly held companies are traded. A stock market can be a primary market or a secondary market. The primary market is where the shares of the company are traded for the first time through the process called Initial Public Offering (IPO) and the secondary market is where the already-owned securities are traded [1]. The stock market is a platform for money exchange and plays a crucial role in a country's economy. Predicting the stock market is a very arduous task because it is volatile and its volatility depends on so many factors including but not limited to the company's strength, dividends, new initiatives, political conditions, global economy, etc. The main aim of investing in the stock market is

* *Corresponding author:* umarbashir932@gmail.com

to gain more and more profit with a low risk of loss. Investors gain profit not only by buying a share at a low price and selling it at a high price but can get profit by selling at a high price and buying back at a low price within the same day, called *short selling* [2]. To minimize the loss and maximize the profit, investors and traders need a system that predicts the stock market by analyzing the trend of the market over the last few years which can prove to be very fruitful for the investors and to make decisions about whether to buy, sell or hold the stock [3]. There are two main approaches to predicting an organization's stock price. The first one is the prediction using technical analysis. In this approach, stock prices like opening price and closing price, highest and lowest price, adjusted close price, and stock volume from the last few years are used to predict the stock's future value. The second approach is qualitative analysis, which is performed based on external factors of the company such as company profile, market situation, political factors, financial news articles, and social media analysis.

The data of the stock market is highly volatile time series data, time series refers to the data collected over a period of time to determine the state of an activity [4]. Different linear models have been used in predicting the stock market like Auto-Regressive (AR), Auto-Regressive Moving Average (ARMA), and Auto-Regressive Integrated Moving Average (ARIMA) [5], but these models perform only for particular time series which means these models perform only for specific company's data and may not perform well for other companies. The nature of the stock market is uncertain and unforeseeable due to which predicting the stock market is risky and it is the main reason for backbreaking in the prediction of the stock market. A wise investment is possible only if the stock market data is thoroughly examined before investment. It is very challenging to analyze the vast amounts of volatile and dynamic financial data. With the rise of online trading, investors are turning from traditional fundamental analysis methods to automated intelligent decision systems.

To analyze this uncertain and volatile data, the application of deep learning (DL) models in the field of finance [6] plays an important role. Deep Neural Networks (DNNs) acquired this name from the architecture of the neural network. It is also called Artificial Neural Network (ANN), which has good approximation power with the capability to learn from experiences. ANNs examine the relationship between the input and output, even if a dataset is complex, and also can handle and analyze massive amounts of data more quickly and effectively than conventional approaches, which may result in better investment decisions. The current study presents a deep analysis of daily closing price predictions of the leading Indian stock market index NIFTY 50 using a state-of-the-art DL approach. This study implements a novel approach to perform stock price prediction by exploring the capability of a customized neural network and extensive hyperparameter optimization in stock index prices. A few noteworthy contributions of the current study are, the first one is a novel dataset has been created using standard technical indicators. Second, an optimized neural network has been developed with an automatic rigorous tuning of hyperparameters like learning rate, first neuron, hidden layers, batch size, activation function, epoch, dropout, and shape. Third, two normalization techniques were applied to the data set to assess the impact on the performance of the model. Finally, a comparative analysis of the proposed model has been done with the already existing modern approaches applied for the prediction of NIFTY 50 and demonstrates that the proposed model results in better performance than the existing study.

The remainder section of this article is structured as follows: A brief idea about ANN and backpropagation, a related work section establishes the concept of stock market data and approaches that have been already applied in this field. The methodology section introduces the dataset preparation and prediction approach, performance analysis discusses the various evaluation metrics that have been used to measure the performance of the proposed model, results achieved in this study are discussed in the experimental results section. Furthermore, the proposed study is compared with the already existing study, and implications of this study are also added. In the final section, several concluding observations and future work are provided.

2. Background

An ANN is simply a mesh of numerical equations where one or more variables act as input, process on input by applying the sequence of equations, and produce one or more results. It is one of the emerging and intelligent techniques that has the capability of adjusting the dynamic nature of nonlinear data and is a bioinspired technique that performs in the same manner as biological neurons [20]. The key advantage of using ANN is that it can identify the underlying patterns from the data, whereas most of the traditional approaches fail [21]. The connection between the neurons in ANN enhances the computational power of the network. In ANN, generally, there are three layers- The input layer, the Hidden layer, and the Output layer. Except input layer, all the neurons in the hidden layer as well as the output layer use an *activation function* to decide whether a neuron should be activated or not [22]. In a neural network, each neuron consists of weighted inputs, activation functions, and output. The structure of the artificial neuron as shown in Fig. 1 is considered the most fundamental unit of a DNN. The artificial neuron has n number of inputs (x_i) and a weight link (w_i) is associated with each input connected to the neuron. Among the three layers of ANN, the hidden layer is the most important because it performs the refining process i.e., it forwards only significant data and patterns from the input layer to the next layer making the network more efficient and faster [23].

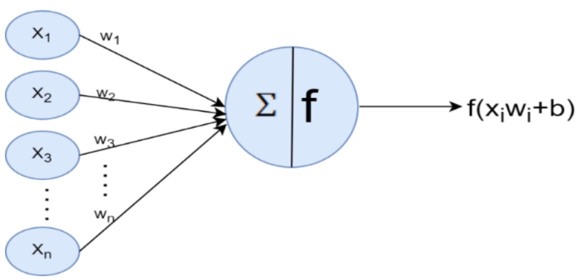

Fig. 1. Architecture of Artificial Neuron.

The neuron performs the summation of inputs multiplied by the corresponding weights as shown below in Eq. (1).

$$Y = \sum x_i w_i + b \qquad (1)$$

Where Y is the weighted sum of all the inputs and b represents the threshold value, the net sum is given to the activation function to generate the output.

The simplest type of ANN is Feed Forward Neural Network, the more appropriate terminology for this is *"Multilayered Network of Perceptrons"* and is commonly known as Multilayer Perceptron (MLP). Fig. 2 shown below illustrates the structure of MLP. The accuracy of the prediction model can be improved by propagating the errors in the backward direction known as backpropagation.

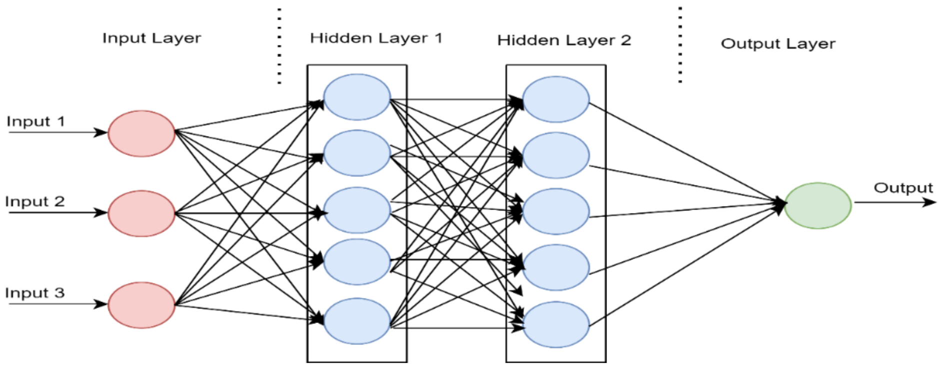

Fig. 2. Architecture of Multilayer Network of Perceptron.

In backpropagation, after every forward pass, a backward pass is performed to adjust the model parameters such as weights and biases [24]. By using backpropagation, weights are updated to minimize the prediction errors and minimize the cost function by applying the gradient descent approach. Firstly, for weights W, arbitrary values are selected (it is a very common approach to choose the weights randomly in the range of -0.1 to 0.1) then the output $\hat{y}(t)$and error E(t) on a selected set of parameters are calculated along with the derivatives as shown below in Fig. 3. If the increase in weight leads to an increase in error, then adjust the weight downwards and vice-versa. This process continues till the loss is minimized. The main point in backpropagation is that it calculates the derivatives of all the weights in a single pass [25].

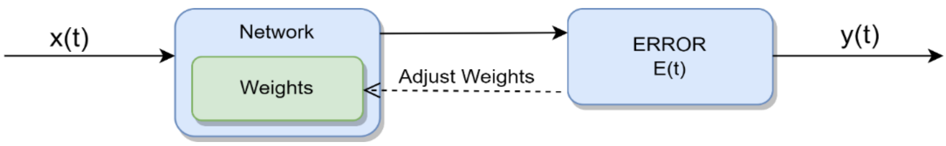

Fig. 3. Backpropagation in Artificial Neural Network.

Optimization of neural networks is a very important step to train the model to perform the specific task. It involves adjusting several parameters to improve the performance of the model, making the model more precise and efficient in performing classifications or predictions [26]. Several reasons why parameter optimization of the neural network is important are: Enhancing model performance, faster convergence, generalization, efficient resource utilization, robustness, and scalability. The selection of specific parameters depends on the type of task and try to find the best values of these parameters [27].

3. Related Work

Due to the versatile nature of ANN, several problems have been solved due to which the technique is getting more attention than other approaches. ANNs are primarily built to extract features from the input data. This deals with the nonlinear relationship of time series data through experience-based learning and produces the optimal results for prediction problems [7]. ANNs are one of the regularly employed machine learning (ML) algorithms utilized to forecast the stock market using time series data [8]. The relevant work that has already been done on the prediction of the market is discussed below:

Kara [9] developed two techniques based on ANN and SVM to predict the direction movement of daily ISE[†] National 100. Technical indicators were used as input and a comprehensive parameter setting was used to improve the prediction performance for both the approaches. The average performance obtained using SVM and ANN is 71 % and 75 % respectively. However, the performance of these two models can be improved by applying a more sensitive hyperparameter setting. Yetis [10] performed the price prediction of the NASDAQ index using ANN using given input parameters OHLV and achieved RMSE 37.12. This study can be further extended by considering some technical indicators with input data for achieving better results. Jain and Kain [11] applied various ML algorithms namely Linear Regression (LR), Random Forest (RF), and Multilayer Perceptron (MLP) for the prediction of the stock market. Ten-year data of two indices Dow Jones Industrial Average and NY Times were used for the experiment. By analyzing the plotted graph MLP showed good results as compared to the other two techniques but the main limitation is that it showed good results only in a certain range. Mostafa [12] applied MLP and regression neural networks to predict the closing price movement of KSE[‡]. The researcher has confirmed that the neural network approach can learn complex relationships between the input features and target features to make better predictions.

A hybrid ANN was developed by Khasei and Bijari [13] which follows two phases. ARIMA model was used in the first phase for the extraction of essential features from the time series data and ANN model was used in the second phase for predicting features extracted in the first phase. The model is applied to three different data sets namely the Canadian lynx data, the US dollar exchange rate data, and the last one is the Wolf's sunspot

---

[†] Istanbul Stock Exchange

[‡] Karachi Stock Exchange

data. The performance of the model has been evaluated using MSE and MAE. Guresen *et al.* [14] analyzed the performance of neural network techniques such as MLP, DAN2, and hybrid neural network which use GARCH for the extraction of new input variables. While applying MLP, DAN2, and GARCH-DAN on NASDAQ data, MAD and MSE achieved 2.51 % and 2478.14, 2.718 % and 1472.27, and 6.48 % and 20901.19 respectively. From this study, it is observed that MLP outperformed DAN2 and GARCH-MLP. In another study, Vijh *et al.* [15] applied ANN and RF for the prediction of the next-day closing price of five different companies. Openly available market data OHLC were used to generate new variables and act as input to the model. For performance evaluation of the model RMSE and MAPE have been used and concluded that ANN outperforms Bhattacharjee and Bhattacharya [16] applied different statistical and ML approaches for the prediction of two tech companies (Tesla and Apple) and performed a comparative study of different prediction approaches in terms of prediction performance. From the comparative analysis, it is concluded that neural network models are found the most appropriate for the prediction of the stock market. The main limitation of this study is that missing values in the dataset are not properly treated which leads to a negative impact on the performance of the prediction technique.

Chatterjee and Bhowmick [17] applied a set of time series, econometric, and ML-based approaches for the prediction of the stock price of three different sector companies i.e., Infosys (IT sector), ICICI (Banking sector), and SUN PHARMA (Health sector). Fifteen years of data were used for training and testing of different models and generalize the best model. From this study, it is concluded that, among time series and econometric models ARIMA was proved the best model and from ML approaches MARS (multivariate adaptive regression spline) was considered the good approach, and from DL, the LSTM model was proved to be the best approach. From all these models, LSTM produced excellent results because this model does not have vanishing gradient and exploding gradient problems. Lamba *et al.* [18] applied four ML approaches (Feed Forward Neural Network, generalized regression neural network, radial basis neural network, and exact radial basis neural network) for prediction of the NIFTY 50 closing price. Nine years (2013 to 2021) of historical data was used to carry out this work. The final experimental results reveal that the radial basis neural network showed 98.17 % accuracy and outperformed the other three approaches. In this study, the researcher applied all these approaches on basic features of the data like opening price, high price, low price, and volume were used as predictors, and closing price as the target feature, the performance of this study can be further increased by including technical indicators to predict the closing price. In another study, Sharma *et al.* [19] performed a comparative analysis of various ML techniques like RF, SVR, Ridge Lasso regression, and KNN. All these approaches were applied to the one-year (March 2022- March 2023) dataset NIFTY 50. For model evaluation, R^2 and MAE were used, and concluded that SVR gives better results as compared to the other three approaches.

By considering the above-mentioned review, we developed an automated system to overcome the restrictions in aforementioned studies. The proposed study introduces a novel approach to stock market prediction using the Nifty 50 dataset and incorporating technical indicators at various levels to enhance the dataset's richness. Rigorous hyperparameter tuning of the DL model has been done to further optimize its performance. For model evaluation, the proposed study used a diverse range of regression-based metrics as shown in the comparative analysis section and demonstrating its robustness and effectiveness in accurately predicting NIFTY 50 prices. These results suggest that this approach offers a valuable contribution to the existing literature on stock market prediction, showcasing the potential of DL and advanced feature engineering for improving forecasting accuracy.

4. Proposed Methodology

In this section, every step that has been done in this study is briefly discussed and the proposed data flow of the model is shown below in Fig. 4.

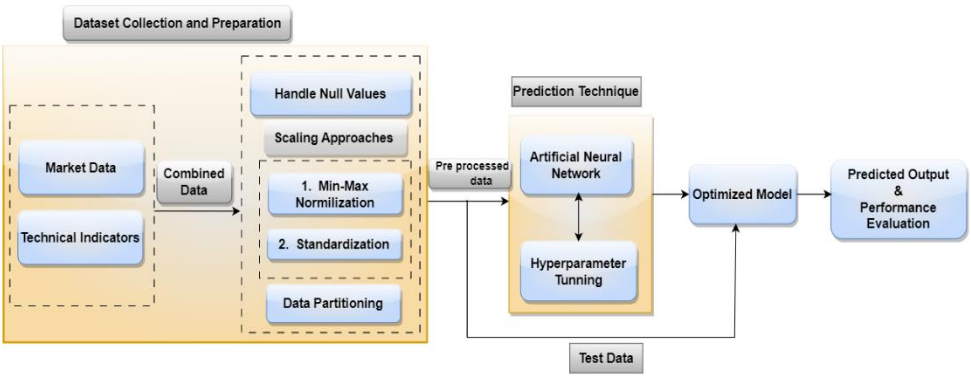

Fig. 4. Proposed methodology for Stock Price Prediction.

4.1. *Dataset collection and preparation*

To start the process of stock prediction, the initial step involves dataset collection. A deep study has been conducted to find the source and format of the data. The selection of the data is affected by various factors like the core of the problem, data availability, data diversity, and relevance of the application. In the current study, NIFTY 50 has been selected for research because it indicates various top sectors such as automobile, bank, IT, real estate, etc. [28]. To generalize our prediction approach, ten years of recent (from January 2013 to December 2023) data of NIFTY 50 has been collected from Yahoo Finance (*https://finance.yahoo.com*) which provides up-to-date information about the stocks which includes stock quotes, press releases, daily market coverage etc. The historical dataset contains openly available stock information like Date, Opening price of the day, Highest price of the day, Lowest price of the day, Closing price, and Volume. Due to some technical issue, the collected data contains some missing values. It is very important to deal with the missing data otherwise the model performance may decrease. There are different approaches to handle the missing values, the approach employed in this study is data imputation. The stock market data is a time series data, so the average of the previous and subsequent day is imputed to replace the missing value and this approach effectively addresses the limitation identified in the previous study [16]. Besides these variables, two technical indicators Moving Average (MA) and Relative Strength Index (RSI) have been added to the market data, both of which have three levels (6 days, 9 days, and 12 days). A Simple Moving Average is created by averaging an asset's price across a predetermined number of periods. There are no weighting parameters taken into account; all prices in the data series are equally weighed [29]. Another technical indicator is the Relative Strength Index (RSI), which is computed based on a certain period, which is 6, 9, and 12 days here in this study. The RSI value lies between the range of 0 and 100. An RSI value close to 100 depicts the stronger buyer's power and close to 0 depicts the stronger seller's power [30]. The formulae and levels of two technical indicators are shown in Table 1. As per literature [31], the addition of technical indicators with the market data makes the model perform better prediction.

Table 1. Technical indicators, formulae, and levels applied in this study.

Technical indicator	Formulae	Levels
Simple n days Moving Average (MA)	$\frac{1}{n}\sum_{i=1}^{n} c_i$	3 (6,9,12)
Relative Strength Index (RSI)	$100 - \left[\left(\frac{100}{1+\left(\frac{AG}{AL}\right)}\right)\right]$	3 (6,9,12)

Note: here *n* represents the period, C_i is the closing price of the i_{th} day, AG represents average gain, AL represents average loss

The features except Closing price act as input features to the neural network, also called predictors. The input features of different ranges are subjected to normalization to unify the data into a common range [32]. Here in this work, two normalization techniques Min-Max and Standardization have been used to see the impact of the normalization technique on the results. Min-max normalization is a technique used to bring all the data into a common range of 0 and 1. Min-Max normalization is performed by using Eq. (2).

$$norm_x = (x - min_x)/(max_x - min_x) \tag{2}$$

Here $norm_x$ represents the normalized value, min_x is the minimum value and max_x is the maximum value in the dataset. Another technique of normalization applied here is standardization, in which data is transformed in such a way that the mean of each feature is zero and the standard deviation is 1. This process is also known as Z-score normalization and is performed by the Eq. (3) as shown below:

$$X' = \frac{X-\mu}{\sigma} \tag{3}$$

Where μ is the feature values mean and σ is the standard deviation of the features. In this case, values are not restricted to any specific range. The preprocessing process has been completed here, the next step is model development and model training. For model training, the preprocessed dataset has been divided into an 80:10:10 ratio i.e., 80 % of the data is used for training the network and the remaining 20 % of the data is further divided into two equal parts, for validation and testing the model. Table 2 given below shows the statistics of all three sets.

Table 2. Statistics of the dataset.

	Training Set	Validation Set	Testing Set
Time Interval	1 /21/2013 - 09/21/2020	09/22/2020 - 05/11/2022	05/12/2022 - 12/29/2023

4.2. *Approach for the prediction*

Analysis of the Indian stock market has been carried out by using ANN with a regressive hyperparameter tuning to achieve optimal results very fast. In the current study, a four-layer ANN has been used with 10 features as input to the input layer of the network and a single neuron in the output layer. The activation function ReLU has been used in hidden layers and, for maintaining the balance between bias and variance the optimal parameters need to be identified for more accurate results [33]. For obtaining optimal parameter setting of the proposed model, hyperparameter tuning has been done using the Talos library and all required relativity has been imported. A dictionary of hyperparameters and their different range of values is built for tuning the model. Instead of passing parameters one by one, the dictionary is supplied to the model and this approach takes less time [34] as compared to other complex and time-consuming approaches like manual search and random search. Table 3 shown below represents the different hyperparameters and their supplied range of values.

Table 3. List of Hyperparameters and their range of values tuned in the current study.

Hyperparameter	Range	Hyperparameter	Range
Learning Rate	(0.01-0.05, 3)	Epochs	[200,300,400,500]
First Neuron	[10,20]	Dropout	(0-0.5,3)
Hidden Layers	[1,2,3]	Shapes	[‘brick’, ‘funnel’]
Batch Size	(10-30, 3)	Activation	[‘relu’]

To adjust hyperparameters, the ‘Scan’ function of the Talos library has been utilized for traversing over both the training and validation datasets. This function uses all the permutations and combinations of all the hyperparameters. The ‘Analyze’ object of the Talos carries all the outcomes of the hyperparameter tuning activity. For obtaining the best model, Talos.best_model (metric: low MAPE) has been used and finally, the configuration of the best model is selected for the final prediction on test data. The configuration of the best model is shown in Table 4.

Table 4. Hyperparameters and their selected values for the final optimized model.

Parameter	Value	Parameter	Value
Start	01/05/24	Activation Function	ReLU
End	01/05/24	Batch size	23
First neuron	20	Dropout	0.0
Epochs	500	Shape	Brick
Hidden layers	3	Lr	0.036

4.3. *Performance analysis*

Every ML algorithm has a loss or cost function at its core that estimates prediction error. The main goal is to reduce this error as much as possible. After each iteration, the cost function is calculated and weights are adjusted accordingly to minimize the error. Here in this study, a default cost function mean square error (MSE) has been used to calculate the average mean of squared difference between the actual value and forecasted value. This loss function penalizes large errors more severely than small ones, which helps to increase the accuracy of the model. For evaluating the performance of the model three evaluation metrics have been used in this study as follows:

4.3.1. *Mean absolute percentage error (MAPE)*

MAPE is a percentage-based metric used to measure the prediction accuracy of the model, particularly in finance and economics. It is also abbreviated as MAPD ("D" represents Deviation) and its range is 0 to 100. A value of MAPE close to 0 depicts that the model is performing accurately and MAPE value moving away from 0 depicts the decrease in accuracy of the model. MAPE is calculated using the Eq. (4) as shown below:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Actual - Predicted}{Actual}\right| \times 100 \tag{4}$$

Where *n* is the number of data points, *Actual* and *Predicted* are the actual and predicted prices.

4.3.2. *R Square*

R Square is also called the coefficient of determination and is a correlation-based metric that is used to find the goodness of linear fit in a regression model [35]. It is defined as the square of the correlation between the actual value and the predicted value. The range of R square is between 0 and 1, a value close to 1 depicts that variables are highly correlated and a value close to 0 represents low correlation. The definition of R^2 is presented in Eq. (5).

$$R^2 = 1 - \frac{\sum_1^n \left(D_{act} - D_{pre}\right)^2}{\sum_1^n \left(D_{act} - \bar{D}_{act}\right)^2} \tag{5}$$

D_{pre} *and* D_{act} represents the predicted and actual values respectively, $\bar{D}_{act}$ is the mean of the actual variable and *n* is the amount of data collected.

4.3.3. *Root mean square error (RMSE)*

The third evaluation metric employed in the current study to assess the model's performance is RMSE. This measures the magnitude error between the actual price and the predicted price. RMSE is computed using Eq. (6) shown below and is defined average distance calculated vertically from the predicted price to the actual price on the fit line [28]. The range of RMSE is (0, +∞), the prediction model is more accurate the smaller the RMSE is.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(predicted\ price - actual\ price)^2}{n}} \tag{6}$$

Where n is the number of data points.

4.4. *Experimental results*

To fit the model, different number of epochs have been considered and it is observed that by varying the number of epochs the model performance also varies. Model performance has been assessed in each case and the prediction results are presented in Table 5. The proposed approach exhibited exceptional performance, achieving a MAPE of 0.18 %, an R2 value of 0.996, and RMSE of 24.82. These evaluation metrics collectively suggest that the model is highly accurate in its predictions. While comparing the performance of the optimized ANN model employed on both types of normalized data, it is observed that the proposed model achieved better results with min-max normalized data as compared to z-score normalized data and it is also concluded that normalization approaches also impact the accuracy of the ANN model. Hyperparameter setting also affects the performance of the model. To ensure the model's effective generalization to unseen data, a validation set has been employed to make adjustments to the hyperparameters and prevent overfitting. The whole experimental work has been conducted under the running environment of an Intel i7-8665U 2.11ghz processor, RAM 16 GB, and Windows 10pro. Python and Keras, an open-source package built on TensorFlow are used to implement the ANN.

Table 5. Comparative performance of optimized ANN on different scaling techniques.

Model	Normalization Technique	Epochs	MAPE (%)	R^2	RMSE
ANN	Min-Max	200	0.28	0.910	36.02
		300	0.23	0.993	29.47
		400	0.20	0.994	27.63
		500	**0.18**	**0.996**	**24.82**
ANN	Z-Score	200	0.29	0.988	39.00
		300	0.23	0.992	32.04
		400	0.16	0.994	26.81
		500	0.20	0.993	29.89

In Fig. 5 the line plot shows the actual price and predicted price on test samples. From the line plot, it is interpreted that stock data is highly volatile and the float of stock is very large. However, it can be seen that the prediction results of the proposed model are very close almost the same as the actual value and the pattern of actual value and predicted value is identical. Experimental results give the idea that neural network architectures with more data sources can predict stock prices more accurately in general.

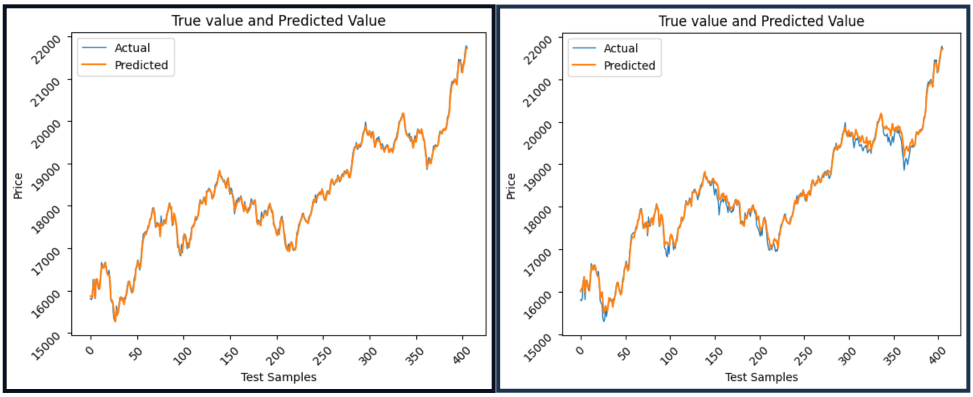

Fig. 5. Actual price vs Predicted price using Optimized ANN (a) using min-max normalized data (b) using z-score normalized data.

4.5. *Comparative analysis*

In this section, a comparative analysis of the proposed study with the already existing state-of-the-art approaches for the prediction of the stock market using the NIFTY 50 dataset was conducted as shown in Table 6. It appears that the proposed study with hyperparameter tuning results showed excellent outcomes on the test set. The work that has been already done in the field of stock market uses different datasets, but in this study, only those papers were mentioned for comparative analysis that have worked on the NIFTY 50 dataset. To evaluate the effectiveness of the model researchers use different evaluation metrics, in this study most commonly used evaluation metrics like MAPE, R^2, and RMSE have been selected for comparison. From the comparison table, it is observed that the proposed approach shows excellent results using all three metrics and makes a significant contribution to the stock analysis research community during this highly volatile nature of the market.

Table 6. Comparative analysis of proposed approaches with the existing study on NIFTY 50.

References	Technique	Dataset	Evaluation metric	
[34]	Optimized ANN	NIFTY 50	MAPE	8.03 %
[36]	CNN RNN LSTM	NIFTY 50	R^2	0.81 0.80 0.53
[28]	Backward elimination with LSTM	NIFTY 50	MAPE RMSE	3.54 % 619.35
Proposed approach	Optimized ANN	NIFTY 50	MAPE RMSE R^2	0.18 % 24.82 0.996

5. Implications

The intraday closing price prediction may have a variety of effects on various stakeholders and elements of the financial ecosystem. The following are some possible ramifications:

(i) Traders may make lucrative trades by using this study to guide their buy or sell signals and boost the trader's assurance by making precise intraday predictions that may lead to better returns and more informed choices.

(ii) This study might be useful in risk assessment and offer insights into possible short-term market fluctuations and benefit from the analysis of real-time financial data.

(iii) Regulatory bodies can use this system to monitor and regulate financial markets.

6. Conclusion and Future Scope

It is important to note that, despite the advancements made in stock prediction models and technologies, predicting stock movements in financial markets is always challenging because of the multitude of causes that affect the stock market, such as fortuitous news, geopolitical news, and industrial conditions. ML and DL techniques have found extensive application in financial markets. This paper investigated the application of ANNs to elevate the accuracy of stock market prediction. In this study, an attempt has been made to predict the future price of the Indian National Stock Exchange index with reliability and greater accuracy using the DNN approach. For speedier and more optimized predictions, a bespoke neural network with a regressive hyperparameter optimization technique has been applied. The results achieved here in this work are quite promising and it is concluded that using DL techniques prediction of the stock market can be made more effective and accurate in the field of the stock market. Although the model showed good results still there may exist deficiencies in the input data. The textual data like social media, financial news, etc. are not considered here in this study. In the future, this work can further be extended by exploring some more DL models and considering other technical indicators as well as textual data from different sources to explore different ensemble methods to investigate the effectiveness of stock market prediction.

References

1. M. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, Procedia Comput. Sci. **132**, 1351 (2018). https://doi.org/10.1016/j.procs.2018.05.050
2. J. E. Engelberg, A. V. Reed, and M. C. Ringgenberg, J. Finance **73**, 755 (2018). https://doi.org/10.1111/jofi.12601
3. A. Murkute, and T. Sarode, Int. J. of Comp. Applicat. **124**, 11 (2015). https://doi.org/10.5120/ijca2015905681
4. M. H. Masum, R. Islam, M. A. Hossen, and A. A. Akhie, J. Sci. Res. **14**, 215 (2022). https://doi.org/10.3329/jsr.v14i1.54973
5. A. Pathak and S. Pathak, Int. J. Eng. Res. **9**, 295 (2020). https://doi.org/10.17577/IJERTV9IS060064
6. C. Yañez, W. Kristjanpoller, and M. C. Minutolo, Neural Comput. Appl. **4**, 15237 (2024).

https://doi.org/10.1007/s00521-024-09898-2
7. M. Inthachot, V. Boonjing, and S. Intakosum, Comput. Intell. Neurosci. **2016**, ID 3045254, (2016). https://doi.org/10.1155/2016/3045254
8. S. Chopra, D. Yadav, and A. N. Chopra, Int. J. Swarm Intell. Evol. Comput. **8**, 1 (2019). https://doi.org/10.4172/2090-4908.1000174
9. Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, Expert Syst. Appl. **38**, 5311 (2011). https://doi.org/10.1016/j.eswa.2010.10.027
10. Y. Yetis, H. Kaplan, and M. Jamshidi, World Autom. Congr. Proc. (2014). https://doi.org/10.1109/WAC.2014.6936118
11. S. Jain and M. Kain, Int. J. Recent Innov. Trends Comput. Commun. **6**, 131 (2018). https://doi.org/10.17762/ijritcc.v6i4.1532
12. M. M. Mostafa, Expert Syst. Appl. **37**, 6302 (2010). https://doi.org/10.1016/j.eswa.2010.02.091
13. M. Khashei and M. Bijari, Expert Syst. Appl. **37**, 479 (2010). https://doi.org/10.1016/j.eswa.2009.05.044
14. E. Guresen, G. Kayakutlu, and T. U. Daim, Expert Syst. Appl. **38**, 10389 (2011). https://doi.org/10.1016/j.eswa.2011.02.068
15. M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, Procedia Comput. Sci. **167**, 599 (2020). https://doi.org/10.1016/j.procs.2020.03.326
16. I. Bhattacharjee and P. Bhattacharja - *4th Int. Conf. Electr. Inf. Commun. Technol. EICT* December (2019) pp. 1-6. https://doi.org/ 10.1109/EICT48899.2019.9068850
17. A. Chatterjee, H. Bhowmick, and J. Sen - *IEEE Mysore Sub Sect. Int. Conf. Mysuru Con* (2021) pp. 289–296. https://doi.org/10.1109/MysuruCon52639.2021.9641610
18. V. Lamba, S. Hooda, V. Solanki, R. Ahuja, S. Ahuja, and A. Kaur - *5th Int. Conf. Inf. Syst. Comput. Networks, ISCON* (2021) pp. 1-5. https://doi.org/10.1109/ISCON52037.2021.9702400
19. D. Sharma, Amisha, P. K. Sarangi and A. K. Sahoo- *3rd Int. Conf. on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India* (2023) pp. 245-250. https://doi.org/ 10.1109/ICACITE57410.2023.10182943
20. P. G. Zhang, Neurocomputing **50**, 159 (2003). https://doi.org/10.1016/S0925-2312(01)00702-0
21. S. K. Pal, B. Datta, and A. Karmakar, J. Sci. Res. **14**, 471 (2022). https://doi.org/10.3329/jsr.v14i2.55669
22. Q. Liu, Z. Tao, Y. Tse, and C. Wang, Financ. Res. Lett. Part A **46**, ID 102209 (2022). https://doi.org/10.1016/j.frl.2021.102209
23. A. Thakkar and K. Chaudhari, Procedia Comput. Sci. **167**, 616 (2020). https://doi.org/10.1016/j.procs.2020.03.328
24. L. Gu, Sci. Rep. **13**, 19330 (2023). https://doi.org/10.1038/s41598-023-46528-8
25. P. J. Werbos, in Proc. IEEE **78**, 1550 (1990). https://doi.org/10.1109/5.58337
26. A. Namdari and T. S. Durrani, Oper. Res. Forum. **2**, ID 38 (2021). https://doi.org/10.1007/s43069-021-00071-2
27. S. Agarwal and J. A. Khan, Theor. Econ. Lett. **9**, 511 (2019). https://doi/org/10.4236/tel.2019.93036
28. S. H. Jafar, S. Akhtar, H. El-Chaarani, P. A. Khan, and R. Binsaddig, J. Risk Financ. Manag. **16**, 423 (2023). https://doi.org/10.3390/jrfm16100423
29. D. Sulistiawan and F. A. Rudiawarni, Int. J. Bus. Glob. **24**, 139 (2020). https://doi.org/10.1504/IJBG.2020.104965
30. H. Xiao, BCP Business Manag. **38**, 255 (2023). https://doi.org/10.54691/bcpbm.v38i.3695
31. Z. Yudong and W. Lenan, Expert Syst. Appl. **36**, 8849 (2009). https://doi.org/10.1016/j.eswa.2008.11.028
32. A. K. Ramotra and V. Mansotra, Int. J. Eng. Trends Technol. **70**, 103 (2022). https://doi.org/10.14445/22315381/IJETT-V70I5P213
33. M. Thormann, J. Farchmin, C. Weisser, and A. Silbersdorff, Stat., Optimiz. Inf. Comput. **9**, 268 (2021). https://doi.org/10.19139/soic-2310-5070-1202
34. N. Rouf, M. Bashir Malik, S. Sharma, I. H. Ra, S. Singh, and A. Meena, Comput. Intell.

Neurosci. **2022**, ID 7097044 (2022). https://doi.org/10.1155/2022/7097044
35. D. Salama, A. Elminaam, A. M. M. El-tanany, M. Abd, E. Fattah, and M. A. Salam, Int. J. Adv. Comput. Sci. Appl. **15**, 442 (2024). https://doi.org/10.14569/IJACSA.2024.0150446
36. Z. Fathali, Z. Kodia, and L. B. Said, Appl. Artif. Intell. **36**, 3100 (2022). https://doi.org/10.1080/08839514.2022.2111134