RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF
MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt24619

A Novel Deep Convolutional Neural Network Algorithm for Equity Price Prediction

A. Jesmine Mary [a], N. Sundaram [a, *]

[a] Department of Commerce, School of Social Sciences and Languages, Vellore Institute of Technology, Vellore-632014, Tamil Nadu, India

* Corresponding Author Email: nsundaram@vit.ac.in

DOI: https://doi.org/10.54392/irjmt24619

Received: 16-08-2024; Revised: 14-11-2024; Accepted: 22-11-2024; Published: 30-11-2024

Abstract: Predicting stock prices is one of the difficult issues for researchers and investors. The study suggests an equity price prediction based on feature neural network extraction. We expect the stock price using technovative forecasting from traditional Machine Learning (ML) models namely Linear Regression (LR), Autoregressive Integrated Moving Averages (ARIMA), and advanced Deep Learning (DL) algorithms such as Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM). We select seven features based on historical data: date, close, open, high, low, volume, and change %. The study's novelty is the prediction accuracy compared to the step-by-step backtesting methodology from ML to DL algorithms. We first use CNN to extract features from the data consisting of the items from the preceding 10 days to 100 days. After that the extracted feature data and LSTM to predict the stock price. Finally, the study used robotic error measure analysis, such as MAE, RMSE, and R2, to assess the forecasting accuracy of all four models. The CNN-LSTM model provides a consistent stock price forecast based on error measures with maximum prediction exactness ranging from 0 to 1, such as MAE-0.03, RMSE-0.04, and R^2-0.98. The proposed CNN-LSTM model maintained its efficiency throughout the process when compared to the LR, ARIMA, and LSTM-RNN models. The study conducts a robustness hypothesis check using the ANOVA test statistic for superior predictability accuracy. In addition, this forecasting technique gives academics real-world experience analyzing financial time series data and confident investment ideas to investors.

Keywords: Equity Prediction, Linear Regression, ARIMA, LSTM-RNN, Deep CNN-LSTM

1. Introduction

A stock market is a cluster of marketplaces and trading venues where people are involved in financial transactions such as purchasing, positioning, and investing in shares of publicly traded corporations. Predictions about the stock market inevitably become hot topics because it's a very dynamic and unpredictable area. Over the past ten years, academics have focused a great deal of emphasis on stock market prediction (SMP) because of the chaotic, non-parametric, high-noise, fluctuating, and non-linear characteristics of stock data [1]. The accurate forecast of stock values depends on the application of artificial intelligence (AI), particularly machine learning (ML) and deep learning (DL) technological models. Some academics utilize a linear forecasting model based on statistics and probability theory to predict short-term stock prices from long-term data, as the stock market covers time series data that can be seasonal [2-3]. Many use cases, including accurate stock price predictions, net asset value predictions [4], bond market predictions [5], image processing [6], learning in games [7], neuroscience [8], energy conservation [9–10], and skin cancer diagnostics [11–12], have demonstrated the remarkable effectiveness of Deep Learning models, particularly the Convolution Neural Network Long Short-Term Memory (CNN-LSTM) model. No other application has shown as much success in Deep Learning models [13-14]. The recent advancements in hybrid CNN-LSTM applications precisely predict stock prices with less execution time and multiple connections consisting of both LSTM and CNN layers. An investor's primary concern regarding the stock market is the fluctuating pattern of the stock price [15]. Most stock price changes are not linear. Stock price fluctuations have always been difficult for analysts to forecast in advance [16]. To improve stock price prediction, machine learning techniques such as CNN-LSTM, Autoregressive Integrated Moving Averages (ARIMA), Long Short-Term Memory-Recurrent Neural Networks (LSTM-RNN), and Linear Regressions (LR) are employed. According to Sharma *et al.* [17], DL and ML algorithms are essential for forecasting a country's economic future and require these forecasts to make wise investment decisions in the financial market. Even during periods of extreme market volatility, like the COVID-19 pandemic in March and April 2020, our research attempts to develop the finest forecasting algorithm for precise stock price predictions and investment decision-making for days in a row. The CNN-LSTM, LSTM-RNN, ARIMA, and LR algorithms are the most popular models for predicting stock return and volatility, while several models have been presented in the financial literature [18–19].

The study aims to improve the background and briefly discuss each baseline model's shortcomings. Linear regression's sensitivity to noise and outliers is one of its primary drawbacks when used for predictive analytics. Errors or random fluctuations in the data are referred to as noise, and data points that significantly deviate from the rest of the data are known as outliers [20]. The traditional ARIMA model is less able to adjust to new information, such as outside influences, holiday effects, and other erratic events, because it mostly depends on historical data [21]. The greater complexity compared to a basic RNN, the LSTM architecture is more difficult to comprehend and implement. Risk of overfitting due to their intricate architecture and numerous parameters, LSTMs are prone to overfitting, particularly in situations when there is a shortage of training data [22]. CNNs also have certain drawbacks that limit their efficacy and usefulness. The main disadvantage of CNNs is that they require a large amount of labeled data to train effectively, which can be costly and time-consuming to obtain and interpret [23]. The limited capacity of LR and ARIMA family ML models to adequately capture the nonstationary and nonlinearity characteristics of financial time series [24-27]. Numerous academics have used DL models in recent years. The literature has proposed AI technique-based deep learning networks (DLNs) for time series prediction, such as LSTM-RNNs and CNN-LSTMs knowledge-based expert system algorithms. CNNs have been used extensively in recent years to predict financial data [1, 28-29]. However, since the CNN-LSTM hybrid model has been widely applied to the prediction of interface depth from gravity data, text sentiment classification, predictive indoor temperature modeling, day-ahead peak load forecasting, risk assessment, thermal comfort prediction in buildings, and the freshness identification of leafy vegetables [30-36]. The researchers proposed the hybrid deep CNN-LSTM method to address the limitations of this restricted study and provide reliable stock price prediction for the NIFTY-100 Index.

The objective of stock price prediction in the real world is to generate significant profits. Predicting the stock market's performance is challenging. Other factors are also considered in the forecast, including physical and psychological traits, irrational and reasonable behavior, and so on. All of these factors combined result in dynamic and fluctuating share prices. It is quite difficult to generate precise stock price forecasts as a result. The present work aims to provide useful insights into NIFTY-100 index share price prediction, the literature, methodology, and results & discussion. Several scale-dependent error metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R2, will be used to assess the predictive models. These measurements serve as standards for evaluating how effectively AI systems forecast stock values. Multi-input layer extraction and parameter optimization will be used to refine the suggested deep CNN-LSTM models for stock market investment decision-making. In addition, the study wants to confirm the normality assumption i.e. hypothesis testing for the machine learning and deep learning model's prediction performance comparisons using MAE and RMSE metrics. Also, the research wants to confirm the robustness of the findings using a parametric approach.

The rest of the document is structured as follows. Section 2 summarizes the literature-relevant work; Section 3 explains the ideas involved in this study; and Section 4 proposes a multi-kernel hybrid CNN-LSTM, LR, ARIMA, and LSTM technique for stock price prediction. In the last section, conclusions and recommendations for additional research are provided after a comparison of the models and experimental results.

2. Related Work

This section discusses various technovative ML and DL models based on neural networks and features applied to BHARTIARTL (Bharti Airtel) telecommunication datasets that will be used to construct the models.

2.1 Proposed deep CNN-LSTM Algorithm

CNN is one of the deep learning techniques that was recently established and has shown remarkable effectiveness in solving the stock trend prediction problem. A hybrid CNN-BiLSTM-AM (attention mechanism) approach was projected by Lu *et al*. [37] to extract the closing price of the stocks the subsequent day. The study suggested seven additional models to compare the accuracy of the error metrics in the stock prices' forecast. When the suggested model was contrasted with other models, the researchers discovered that it generated insightful forecasts and provided a trustworthy framework for investors to choose stocks. A hybrid model that integrated an LSTM and CNN for a precise stock price prediction was presented by Wu *et al.* [38]. The stock sequence array convolutional LSTM (SACLSTM) model was created by the study. Convolutional and pooling layers were used to build the CNN model and extract it for the LSTM model's input vector. The work trials were conducted utilizing ten stock data sets from the US and Taiwan stock markets.

According to the researchers, the CNN-LSTM model generated more accurate predictions. Ishwarappa and Anuradha [39] suggested using deep CNN in conjunction with a reinforcement LSTM model to predict future stock prices using huge data. Several trials, including one month, one week, and one day ahead, were conducted by the study model. The model anticipated improved performance in terms of scale-dependence performance measures, according to the simulation findings. The paper claims that the hybrid CNN-TLSTM (tanh-LSTM) model is highly effective at forecasting stock prices for complicated nonlinear data and the USD/CNY exchange rate to determine the rate for the next trading day. The MLP, CNN, and LSTM models have two major shortcomings: they only have one model structure and have poorer predicting accuracy. TLSTM and CNN make up the model; the TLSTM component extracts feature factors from the TLSTM component, while the convolution layer extracts feature factors from the input data. With extremely low MSE and MAE errors, this model can achieve 0.00038 and 0.18945, respectively, with MAE and MSE errors of 0.18945 and 0.00038 [40].

Nourbakhsh and Habibi [41] performed the model error in stock price trend prediction by combining the CNN and LSTM models with fundamental analysis elements such as profitability, P/E ratio, and the number of firm transactions. Four categories of finance, petroleum, basic metals, and non-metallic minerals were used in the study to assess the MAE and MAPE errors. According to the findings, the suggested model improved error prediction and performance. The CNN deep learning model was proposed by Kirisci and Yolcu [42] for stock price forecasting. Three convolutional layers, five fully connected layers, and Relu and Elu activation were used to select the composite model. The dataset was linked to the Financial Time Stock Exchange for London stock market data (FTSE) and the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX). When compared to ANN, LSTM, fuzzy-based approaches, and other conventional methods, the researchers discovered that the predicted CNN model showed an exceptional performance. A deep CNN network model, was employed by Mukherjee *et al*. [43] and was widely utilized to forecast stock market prices. Therefore, the accuracy of the CNN model was 98.92%, while the ANN model's accuracy was 97.66%. As an outcome, a better prediction was created using the data from the quantized dataset inside a specific time frame.

2.2 Deep LSTM-RNN Algorithm

The effectiveness of the LSTM depth network in the field of stock price prediction is worthy of further exploration. An LSTM deep learning model was used by Yadav *et al*. [44] to predict time series in the Indian stock market. According to the study, the LSTM model's performance was mostly determined by the selection of some hyper-parameters, which had to be done carefully to provide positive outcomes. According to the study's findings, stateless LSTMs are more stable than stateful LSTMs. Moghar and Hamiche [45] applied an RNN-based LSTM model to predict the two stocks in the New York Stock Exchange (NYSE) i.e., GOOGLE and NKE extracted from Yahoo finance stock price. The study covered the period from 8/19/2004 to 12/19/2019 and 1/4/2010 to 12/19/2019. The study discovered that both assets' opening price evolution could be tracked by the deep learning algorithm.

Bathla *et al.* [24], investigated the LSTM model with Adam optimizer, and the sigmoid activation function was tested and trained in the model. The duration of the study was conducted from Jan 2010 to March 2020. The proposed model MAPE score was NSE (3.89), BSE (1.21), NASDAQ (3.01), NYSE (1.19), Dow Jones (2.03), and Nikkei 225 indexes (0.86) respectively. The study used the 60-day window size for the stock price prediction. The experimental analysis proved that the LSTM model provided adequate accuracy. Perry and Jacqueline Rose [46] used a single-layer and multilayer LSTM neural network model to predict the next-day closing price of the S&P 500 index. The study used the correlation coefficient (R), RMSE, and MAPE to compare the performance. In comparison to multilayer LSTM models, the experimental result shows that the single-layer LSTM model offers a better fit and higher prediction accuracy. The LSTM and Support Vector Regression (SVR) models were trained and tested using the study. The study was carried out utilizing publicly available data, specifically Japanese stock data. MAE, RMSE, and MSE were the three metrics used to compare the model's accuracy. With an MAE of 0.009058733, an RMSE of 0.011169727, and an MSE of 0.000124763, the analysis revealed that the LSTM with the dropout strategy yielded the best results [47].

An empirical study of LSTM networks about time-series prediction was carried out by Billah *et al*. [48]. The study used a variety of real-world data and offered quantitative insights regarding LSTM performance. In contrast to conventional forecasting techniques, the LSTM models showed an impressive 23.4% decrease in MAE. In particular, LSTM significantly outperformed baseline models, achieving an average forecast accuracy of 89.7% in financial market estimates. Dhokane and Agarwal [49] focused on combining the LSTM algorithm for stock price prediction with some EMAs (50-day, 100-day, and 150-day EMA) and other potential stock market indicators, such as the relative strength index (RSI). Using Yahoo Finance, the chosen dataset was obtained from NSE India. The findings demonstrated that the stock closing price prediction model's performance was enhanced by the addition of technical indicators to the conventional open, high, low, close, and volume (OHLCV) model. The RMSE, MAPE, and R^2 scores were the assessment metrics employed in this study. The suggested technique performs well in stock closing price prediction, as seen by the decreases in the R^2 score for the RMSE and MAPE.

2.3 ARIMA Algorithm

The study looked at the ARIMA model's performance. This study sought to ascertain how well the ARIMA model predicted changes in the stock market. Accuracy was shown to be rather consistent across different volumes of the training dataset when the rank of the ARIMA model was determined using the Akaike information criterion (AICs). The model's accuracy is gauged by the MAE. Although accuracy varies by sector, the model achieved an accuracy of 85% [50]. The ARIMA forecasting model was employed by Dadhich *et al.* [51] to predict stock prices. The study was chosen for the time frame of March 1–28, 2018. According to the findings, the suggested model was the most effective predictive model for short-term stock price forecasting. To anticipate the time series in their stock market prediction, Sirisha *et al.* [52] used the neural network's LSTM method, ARIMA, and SARIMA. Not for SARIMA or LSTM, but only for ARIMA, the data has been converted into a stationary dataset. The approximate results of the study were 93.84% (ARIMA), 94.378% (SARIMA), and 97.01% (LSTM), with estimates for the next five years in mind. The results of the fitted analysis show that the LSTM performs better than both statistical models in terms of constructing the best model. The transformer model, a recently developed machine learning model, was applied by Muhammad *et al*. [53] to forecast the future stock prices of the Dhaka Stock Exchange (DSE), Bangladesh's top stock exchange. For the majority of the equities, the experiments showed encouraging outcomes and a manageable root-mean-squared error. The model's performance was also compared to that of the well-known benchmark stock forecasting model, ARIMA, and the findings showed that it performed satisfactorily.

2.4 Linear Regression Algorithm

The linear regression machine learning method was trained and evaluated with Yahoo Finance data to predict a numerical value. Means of Open, Low, High, and Close prices are the parameters. The multiple R-value, according to the result, is 0.612. A score close to 1 indicates that the regression line and the least squares value are appropriate and well-suited to the data [54]. Antad *et al.* [55] state that the main objective is to employ linear regression models to acquire more precise answers. The dataset that will be used to train the models can be altered to yield more accurate results from the linear regression models. This study showed that linear regression is the best and most appropriate technique for predicting stock market analysis. The results of the studies showed that the linear regression method outperformed other machine learning strategies in terms of accuracy. Nonetheless, other experts also mentioned that they intended to look into the possibility of using neural networks to anticipate stock market movements in the future. Sangeetha and Alfia [54] predicted a company's stock price with a higher level of accuracy and dependability using machine learning techniques. The incorporation of the assessed linear regression model as a method for determining stock prices was the experts' primary contribution.

This study intends to advance knowledge by enhancing Bharti Airtel stock forecasts by integrating ideas from these diverse areas of literature and providing a thorough framework that integrates input characteristics with CNN filter optimization techniques and LSTM network layers. A more sophisticated and reliable strategy for predicting NIFTY-100 index equity prices will result from combining these components.

3. Data & Methodology

3.1 Dataset

The proposed work involved gathering historical daily equity datasets of Bharti Airtel telecommunications sector (Index-BHARTIARTL) closing prices. The closing price for the following day is anticipated based on several influential characteristics, such as date, close, open, high, low, volume, and change %. The study used secondary data and the sources collected from published scientific articles. The datasets used for the work were gathered from Investment India's official website for the 15 years from 2009 to 2023. The data analysis method is used by both ML and DL algorithms. The study approached data preprocessing and converted the unstructured, raw data into a better, normalized, and more understandable format with the help of robust training models. The data was prepared to enhance its quality. We used the training set and test set data in the same operating environment to compare CNN-LSTM with linear, ARIMA, LSTM, and other methods to demonstrate the efficacy of CNN-LSTM. Every experiment uses an Intel Core i3 processor running in a Jupyter Notebook environment hosted on the Python 3 Google Cloud compute engine backend platform. Data processing and feature development are done with Python software packages such as NumPy, Pandas, Seaborn, OS, Math, Sklearn MinMaxScaler, etc. Additionally, the statistical validation was performed using the SciPy package. The Matplotlib, Pmdarima, Dese, Sequential, and Scikit libraries are used to build and train all other machine learning models, whereas Keras is used to build deep learning models using the TensorFlow backend.

3.2 Data frame efficiency and feature extraction

The NIFTY 100 index Bharti Airtel telecommunications equity sector data sets are chosen as the experimental data for this experiment.

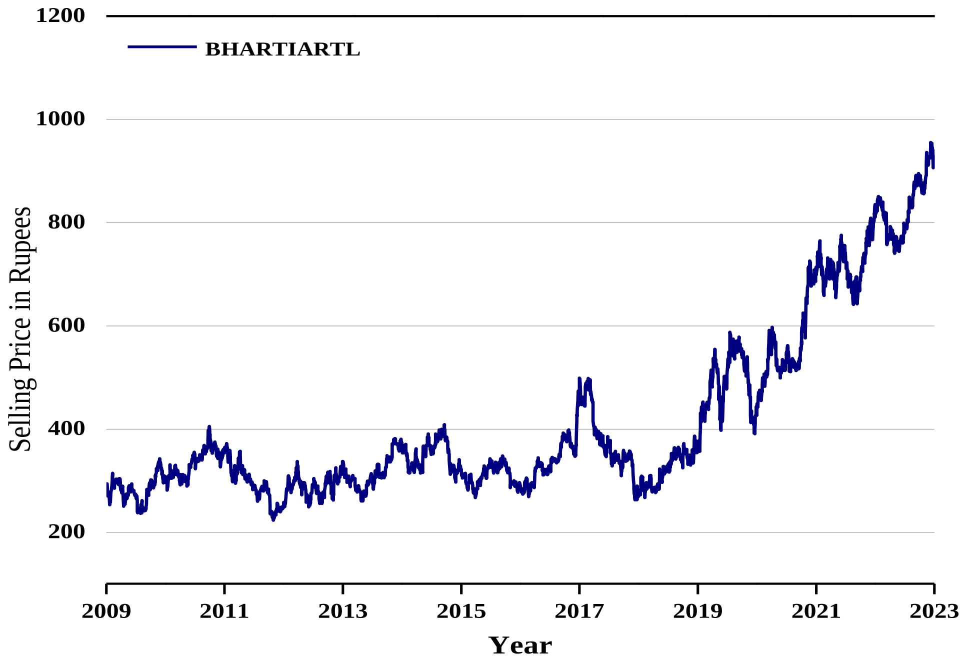

Figure 1. BHARTIARTL's price history from 2009 to 2023

Table 1. BHARTIARTL daily datasets available for training and testing

Index	Dataset Period	Model	Available Data	Produced Data	Trained Data	Testing Data
BHARTIARTL	01-01-2009 to 29-12-2023	Linear	3714	3714	2980	745
		ARIMA	3714	3714	2980	745
		LSTM	3714	3714	2980	745
		CNN-LSTM	3714	3614	2898	725
			14,856	14,756	11838	2960

Source: Compiled by Python

Table 2. Data unique value in the CNN-LSTM model

Date	3714	Change %	896
Close	2867	Daily return	3702
Open	2357	Moving average for 10 days	3665
High	2620	Moving average for 50 days	3657
Low	2699	Moving average for 100 days	3606
Volume	1439	Data type	Int64

Source: Compiled by Python

Table 3. Data frame efficiency statistics of the proposed model

BHARTIARTL- Daily Returns						
Data Unique	Min.	25%	50%	75%	Max.	S. D
MA for 10 days	227	300	341	475	1011	176
MA for 50 days	239	301	340	470	970	171
MA for 100 days	256	303	340	472	934	165

Source: Compiled by Python

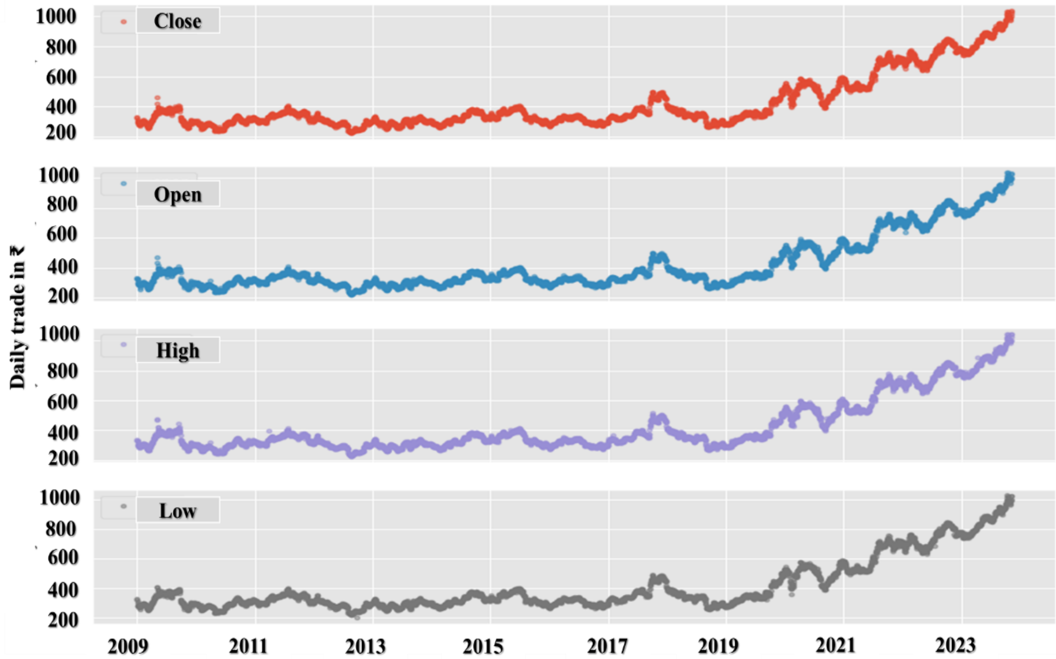

Figure 2. Data feature extraction of daily traded plots using CNN-LSTM

The Investment India website provided the daily available data for 3714 trading days between January 1, 2009, and December 29, 2023. Seven technical indicators make up each piece of data: date, close, open, high, low, volume, and change %. Figure 1 shows the historical price returns of BHARTIARTL from 2009 to 2023. The duration of data collection, the different scheme tags, the number of training and testing patterns, and the observed dataset that the models used are all listed in Table 3. 80% of the data was utilized as the training set for the LR, ARIMA, and LSTM models, and the remaining 20% was used as the testing set.

The proposed CNN-LSTM model produced 3614 data out of 3714 available data. The extracted dataset by the suggested model is 2898 for training and 725 for testing i.e., 80% and 20%. Table 1 displays the complete training and testing dataset available for the study, which is 14856, produced data is 14756, trained data is 11838, and tested data is 2960. The available data was entirely generated for the following step in this case after the CNN-LSTM model eliminated the null value and the remaining forecasting model. The BHARTIARTL's data frame efficiency from 2009 to 2023 is extracted in Figure 2. The data unique value of each extracted feature produced by the model is given in Table 2.

According to Table 3, using the proposed CNN-LSTM model, extending the data frame values from 10 days to 50 days the deviation level in daily return is 5, and from 50 days to 100 days, the deviation level in daily return is 6.

3.3 Model Implementation

The real issue in applied machine learning is model selection. As an easy-to-compute benchmark, we examine and contrast several predictive model types, such as deep CNN- LSTM, LSTM recurrent neural networks, ARIMA models, and linear regression. Because time series prediction involves the use of the deep CNN-LSTM model, it presents a particularly difficult task due to the presence of long-term trends, seasonal and cyclical oscillations, and random noise. The selected deep LSTM recurrent neural network model can be fitted using very straightforward criteria. Considering the extraordinary stochastic character of the training process, we use a range of training datasets to train residual models other than linear regression by averaging the cross-sectional ranks resulting from the projected possibilities. Every research period's hyperparameter is individually optimized based on the classification performance. Figure 3 shows the non-stationery dataset converted into the stationery return using the proposed CNN-LSTM hybrid model.

3.3.1 Linear regression

According to Larose et al. [56], linear regression is the first regression analysis technique that has been thoroughly studied and is frequently applied in real-world settings. The trend linear regression metrics were used to estimate the high-increase price of the stock over time, based on the association between selling price movement and period.

We use simple linear regression, which consists of a regression model with a single independent variable and dependent variables.

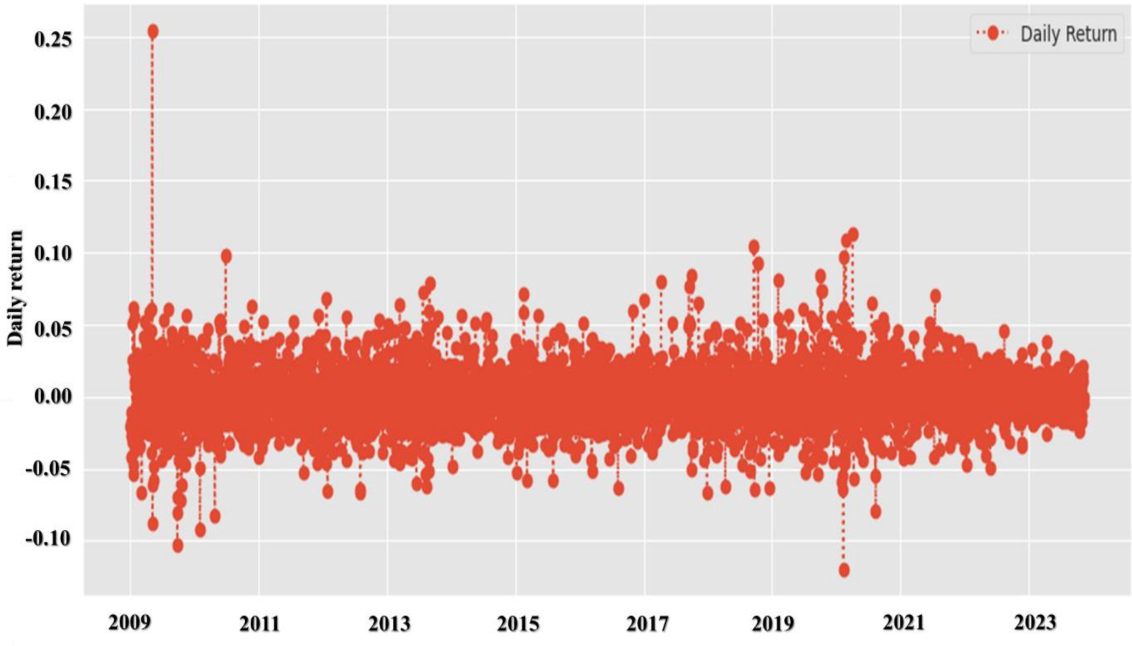

Figure 3. Daily return stationery from 2009 to 2023

Predicting the stock price in the best-fit line Y is difficult due to the dependent variables, while the independent variable, represented by the following equation, is time [57].

$$Y = \beta 0 + \beta 1X \quad (1)$$

Where the NIFTY-100 index equity share's closing period and daily selling price are represented by $\beta_0$and $\beta_1 X$. Given this, Y shows that, in addition to the capital investment and profit dividends, the expected value of the dependent variable equity shares between the purchase price and selling price has improved. X measures the daily movement of the selling price using the slope point.

$$y = mx + c \quad (2)$$

The final result, which quantifies the difference between each point in the dataset, is constant between β_0 and $\beta_1 X$, the total squared errors. With the linear system, a straight line is fitted using the following equation to allow for the maximum possible sum of the position points in the dataset [58].

3.3.2 ARIMA

The traditional notation (p, d, q) is used to easily classify the ARIMA model, where the parameters are replaced by integer values. The standard equation provided by Seabold and Perktold [59] for the p autoregressive model representation.

$$yt = c + \phi 1yt - 1 + \phi 2yt - 2 + \cdots + \phi pyt - p + ct \quad (3)$$

In a pure AR model, yt is completely dependent on its previous value yt − 1, yt − 2 and ct stands for background noise.

A typical representation of a d in the model does such, where I in ARIMA refers to Integrated. A non-seasonal ARIMA model is created using the following formula by Seabold and Perktold [59], in which time series are different to create the stationary and coupled AR and MA models.

$$y't = c + \phi 1y't - 1 + \cdots + \phi py't - p + \theta 1ct - 1 + \cdots + \theta qct - q + ct \quad (4)$$

Where the equation additionally included lag errors from the MA model and insulated values for yt from the AR model. $y't$ indicates a series of differences made more than one. When data progress is steady, the ARIMA model's order differencing (d) takes a value of 1.

The Moving Average (MA) in the ARIMA equation, however, is represented by q in the model. Reversing to the AR model, the MA model is dependent on earlier forecast errors. Utilizing the equation provided by Seabold and Perktold [59] is a common technique for producing a moving average of q.

$$yt = c + ct + \theta 1ct - 1 + \theta 2ct - 2 + \cdots + \theta qct - q \quad (5)$$

Where ct stands for anticipated errors in the future as well as white noise (amplitude of the spectral noise, η, for gravity measurements). The moving average window size approach, as explained by Harrop and Velicer, is commonly known as the q. Equations (3) through (5) then explained the ARIMA technique used in the observations.

3.3.3 LSTM-RNN

Selecting the data to be deleted from the cell state is the first stage in the LSTM process. The formula provided by Lin et al. [60] is used by the forget gate (ft) to determine the subsequent choice.

$$ft = \sigma(XtUf + St - 1Wf + bf) \quad (6)$$

LSTMs have one tanh layer ($\tilde{C}t$) and three logistic sigmoid gates, whereas RNNs only have one tanh layer. By looking at the next cell, it will determine which information is relevant and which may be disregarded. Zhang et al. [61] utilized mathematical formulas to calculate the two-input gate and tanh layer.

RNNs only have one tanh layer, but LSTMs have three logistic sigmoid gates and one tanh layer ($\tilde{C}t$). It will determine which information should be discarded and which is necessary based on the next cell. Zhang et al.'s mathematical formulas are used to calculate the tanh layer and two-input gate [61].

$$it = \sigma(XtUi + St - 1Wi + bi) \quad (7)$$

$$\tilde{C}t = \tanh(XtUc + St - 1Wc + bc) \quad (8)$$

The next step is to modify the prior cell state, C_{t-1}, and then use Zhang et al. 2021's formula to evaluate the new cell state, C_{t-1} given by Zhang et al. [61].

$$C1 = Ct - 1 \otimes ft \oplus it \otimes \tilde{C}t \quad (9)$$

The ultimate determination concerning the result will be produced. Even so, the cell state will determine the filtered output that is produced. In this stage, the output gate (o_t) determines how one component of the cell state will be fashioned as output. Once the cell has passed through the output gate and the tanh layer, which forces the values to be between -1 and 1, the cell state is determined using the following mathematical calculations by Zhang et al. [61].

$$(ot) = \sigma(XtUo + St - 1Wo + bo) \quad (10)$$

$$St = ot \otimes \tanh(Ct) \quad (11)$$

The following terms were indicated by the aforementioned (6) – (11) formulations. The input weights are represented by U^f, U^i, U^c, and U^o represents the input weights, where

W^f, W^i, W^c, and W^o represents the weight matrix, where b_f, b_i, b_c, and b_o denotes the bias vectors, while the sigmoid function is denoted by σ.

3.3.4 DCNN-LSTM

The CNN-LSTM parameters for the current investigation are shown in Table 4. According to the model, the network's parameter values are built as follows: The data in the input training set is a four-dimensional data vector (None, 1, 100, 1), where 100, 1 represents the characteristic of the input dimension and 1 is the timestep size.

Table 4. CNN-LSTM BHARTIARTL parameter configuration

Parameters	Value
Convolution layer filters	23
Convolution layer kernel size	3
Convolution layer activation function	tanh
Convolution layer padding	Same
Pooling layer pool size	2
Pooling layer padding	Same
Pooling layer activation function	Relu
Number of hidden units in the LSTM layer	40
CNN layer activation function	Relu
LSTM layer activation function	tanh
Final layer activation function	Linear
Time step	10
Batch size	40
Learning rate	0.0023
Optimizer	Adam
Loss function	RMSE & MAE
Epochs	73

Source: Compiled by Python

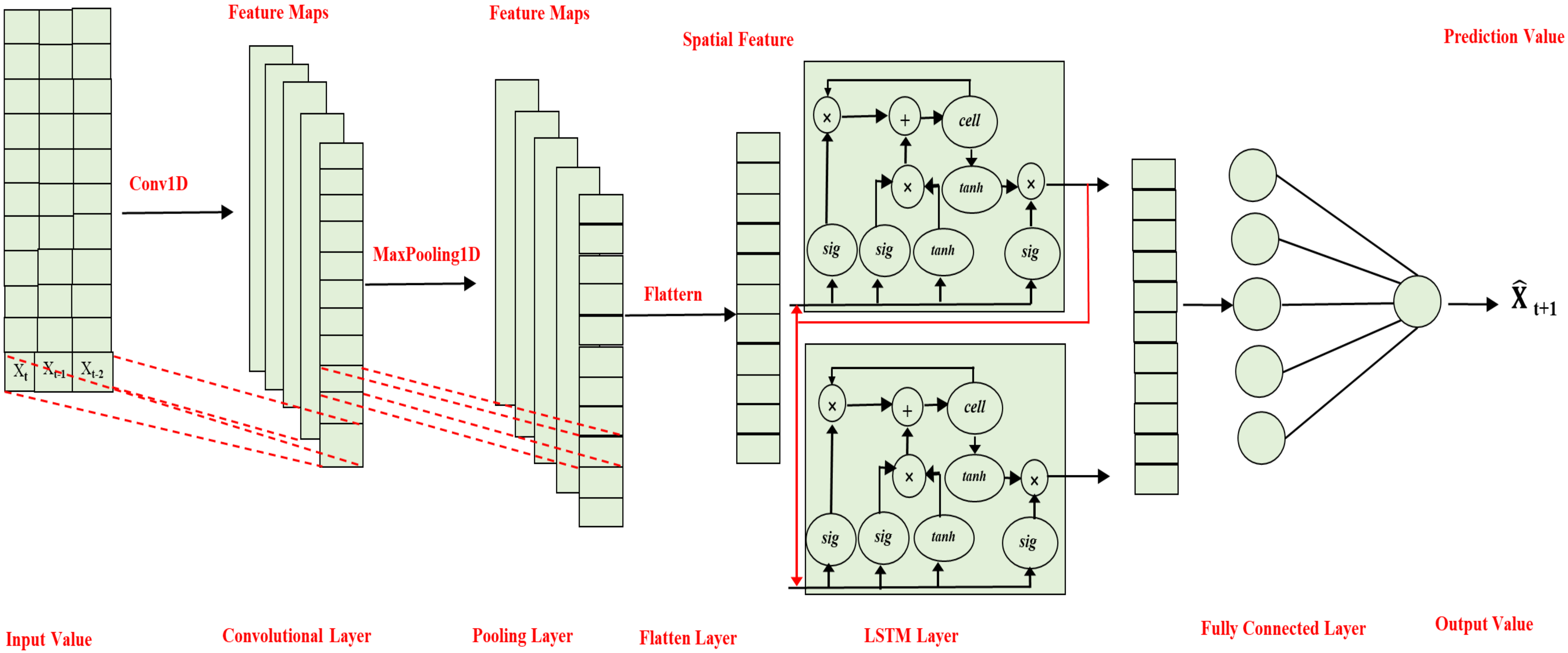

Figure 4. The CNN-LSTM model architecture

To extract added features and create a four-dimensional output vector (None, 1, 98, 64), where the convolution layer's filters have a size of 64. the one-dimensional convolution layer receives the input first. When the vector moves on to the pooling layer, it similarly yields a four-dimensional output vector (None, 1, 98, 64). Extending the model architecture in line with Sezer and Ozbayoglu [62]. To create new feature indices, the original image was convolved twice using a particular filter. The kernel size was about one, and the number of filters assigned was 64 and 128. In a random processing mode, every filter produced a separate convolution operation to provide a different filter effect. The greatest value in the convolutional matrix was then chosen using max-pooling. In the end, overfitting was avoided by flattening and connecting the neurons to the subsequent layer using a fully linked layer. Voting was then conducted to classify the retrieved feature information. The input layer, convolutional layer, pooling layer, flatten layer, LSTM layer, and fully connected layer make up the CNN-LSTM model, as seen in Figure 4. Consequently, parameter optimization was employed to get the optimal accuracy value

3.4 Evaluation Metrics

Regression metrics are quantitative tools for evaluating the goodness-of-fit of regression models and quantifying errors in values that differ from predictions. When assessing the effectiveness of prediction models that concentrate on continuous numerical outputs, several indicators are essential.

The forecasting effect of CNN-LSTM is evaluated using the mean absolute error (MAE), root mean square error (RMSE), and R-square (R^2) evaluation criteria. Forecast accuracy is often measured using the Mean Absolute Error metric. Regression model performance is frequently assessed using the MAE. The mean absolute difference between the projected and actual values is computed. A lower MAE indicates that the model is more accurate. Regression model performance is commonly assessed using the Root Mean Squared Error measure. It calculates the average size of the discrepancies between the actual and projected numbers. R^2 is a statistical measure of how well a regression model fits data. It displays the extent to which the independent variable or variables can be used to predict the variance in the dependent variable. Better matches are indicated by higher R-squared values, which range from 0 to 1.

The following formula is used to determine the MAE:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}i - tyi| \quad (12)$$

The formula for calculating the RMSE is as follows:

$$\mathrm{RMSE} = \sqrt{(\sum(Yi - Yi)^2/n)} \quad (13)$$

Where $\hat{yi}$ is the predicted value and yi is the actual value. The lesser the value of

RMSE, the healthier the prediction. The formula for manipulative the R^2 is as follows:

$$R^2 = (1 - (SSR/SST) \quad (14)$$

The sum of squared deviations between the actual values and the projected values from the regression model is known as the SSR. The variance that remains unresolved is calculated. The SST (Total Sum of Squares) is the sum of the squared differences between the mean and actual values of the dependent variable. It calculates the overall variance. $R2$ has a value range of (0,1). The forecasting accuracy increases with the distance between the predicted and real values from 0, as indicated by the values of RMSE and MAE. The model's degree of fitting is higher when R^2 is nearer 1.

4. Results and Discussion

In the BHARTIARTL datasets, 80% and 20% of the datasets have been selected for training and testing by all four models including the proposed CNN-LSTM. The test set data is forecasted using the output of four training models, and the true value is presented in Figures 5–8 in comparison to the predicted value. This is carried out following the training of CNN-LSTM, LSTM, ARIMA, and Linear utilizing the processed training set data. To minimize the error in the results, the BHARTIARTL dataset has been subjected to an LSTM feature scaling of crucial data preprocessing techniques using mini-max normalization and a CNN powerful panda's library for data cleaning, transformation, data analysis, data visualization, and reading the export & import databases. Additionally, the datasets used for testing and training were split into 80% and 20% components. Using genetic algorithm (GA) optimization techniques for tuning the hyperparameters, the researchers used a variety of kernel filter functions including linear, ARIMA, LSTM, and CNN-LSTM ML & DL models. Hence, the BHARTIARTL kernel and GA achieve a better regression result with a CNN-LSTM model MAE is 0.03, RMSE 0.04, and R^2 0.98 which is significantly less than Wang *et al.* [63], Wang et al. [40], and most other published articles. The similar study error metrics of MAE, RMSE, and R^2 results in the literature were 1.36, 11.73, and 1.02 was significantly more than the proposed CNN-LSTM models [64]. In Table 5, the proposed model score is in bold and it reveals less than 1 is an acceptable error. The existing model error score is more than 1 which is not advisable.

Table 5. Comparison of BHARTIARTL's four methods evaluation indexes

Method	MAE	RMSE	R^2
Linear	88.98	112.21	0.58
ARIMA	230.62	267.55	-0.52
LSTM	15.53	19.32	0.93
CNN-LSTM	**0.03**	**0.04**	**0.98**

Source: Compiled by Python

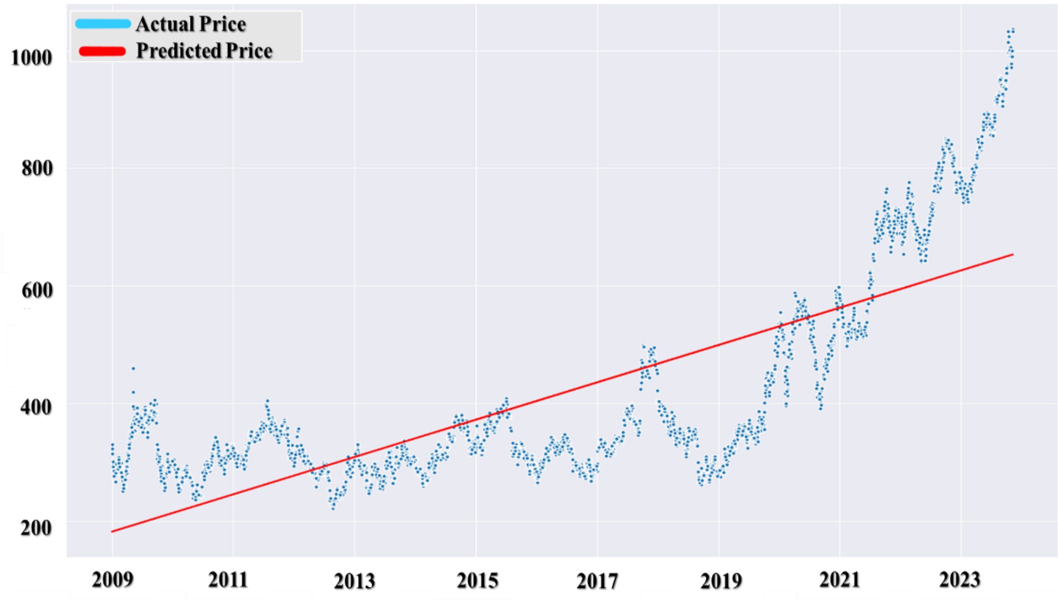

Figure 5. Predicting BHARTIARTL's stock price with the linear regression model

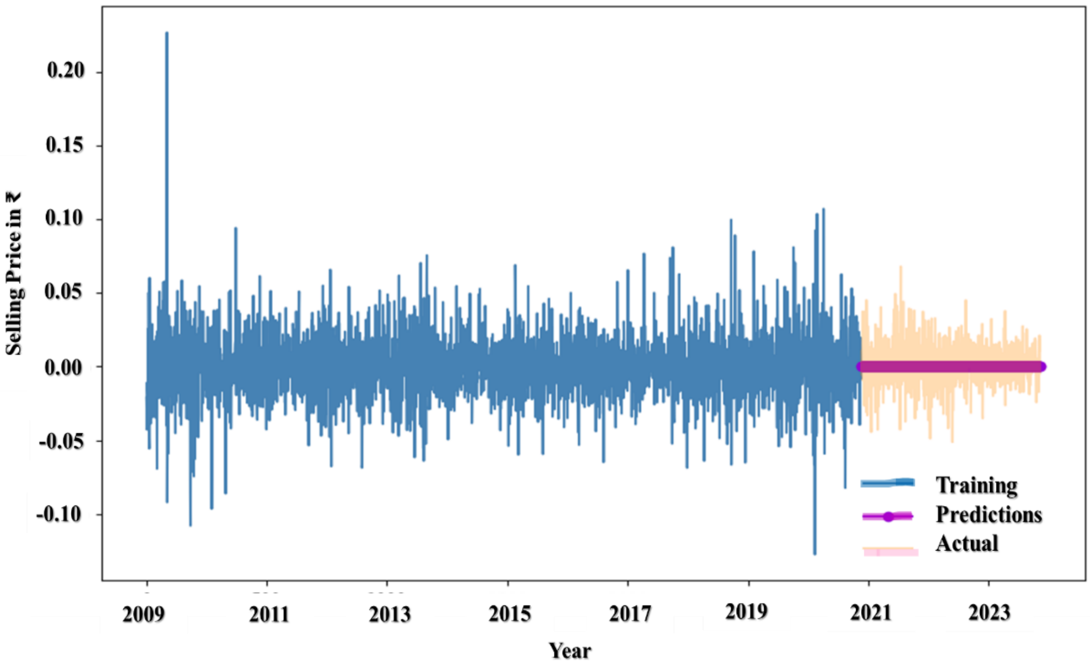

Figure 6. Predicting BHARTIARTL's stock price with the ARIMA model

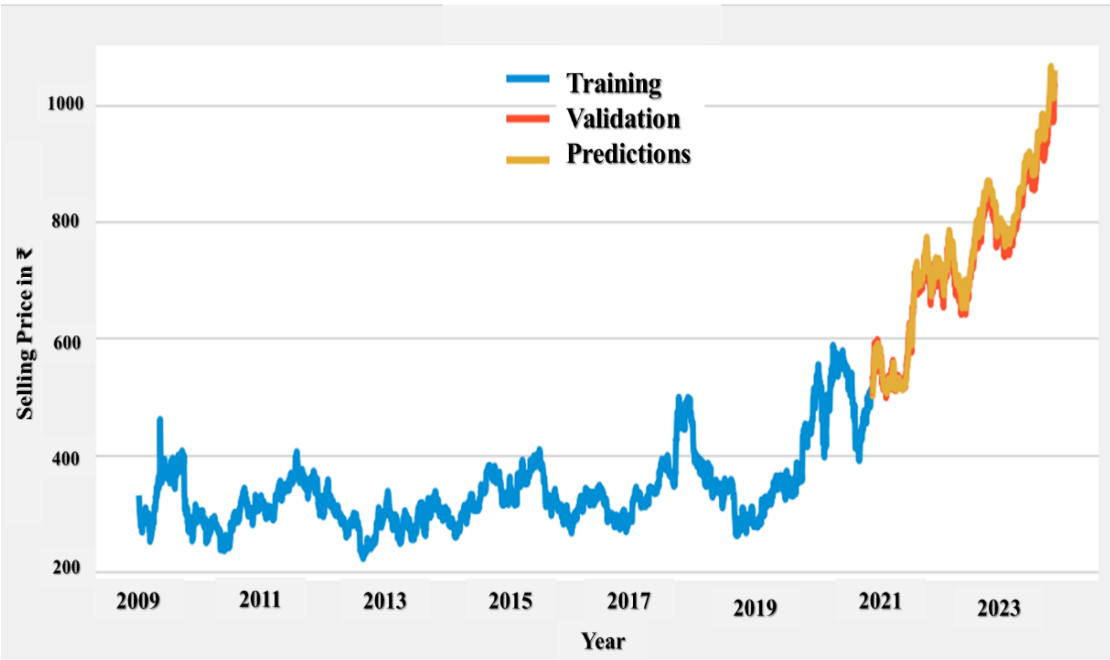

Figure 7. Predicting the price of BHARTIARTL stocks using the LSTM-RNN model

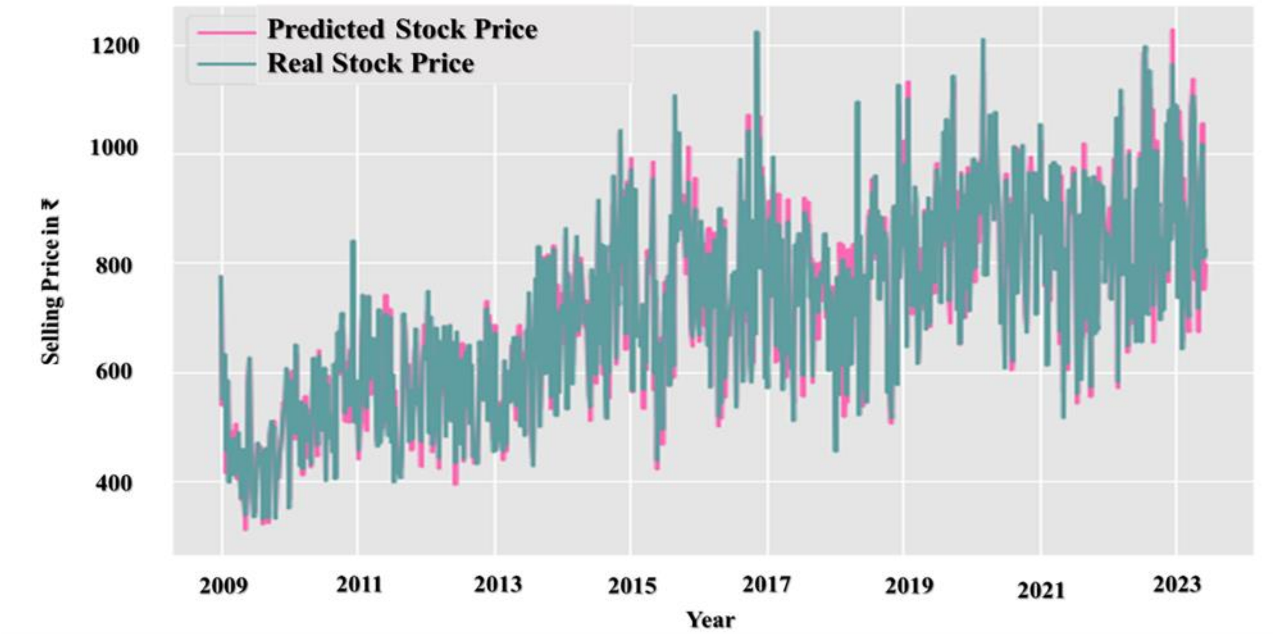

Figure 8. BHARTIARTL stock price prediction using CNN-LSTM

Table 6. Comparing the CNN-LSTM model's MAE and RMSE to baseline methods

Model Comparison	Metrics	Test	F-value	P-value	Sig. (α)	Conclusion
CNN-LSTM vs. Linear Regression	MAE & RMSE	ANOVA	74.96	0.01	0.05	Reject H0: CNN-LSTM has significantly lower MAE & RMSE than Linear Regression.
CNN-LSTM vs. ARIMA	MAE & RMSE	ANOVA	181.92	0.005	0.05	Reject H0: CNN-LSTM has significantly lower MAE & RMSE than ARIMA.
CNN-LSTM vs. LSTM-RNN	MAE & RMSE	ANOVA	84.21	0.01	0.05	Reject H0: CNN-LSTM has significantly lower MAE & RMSE than LSTM-RNN.

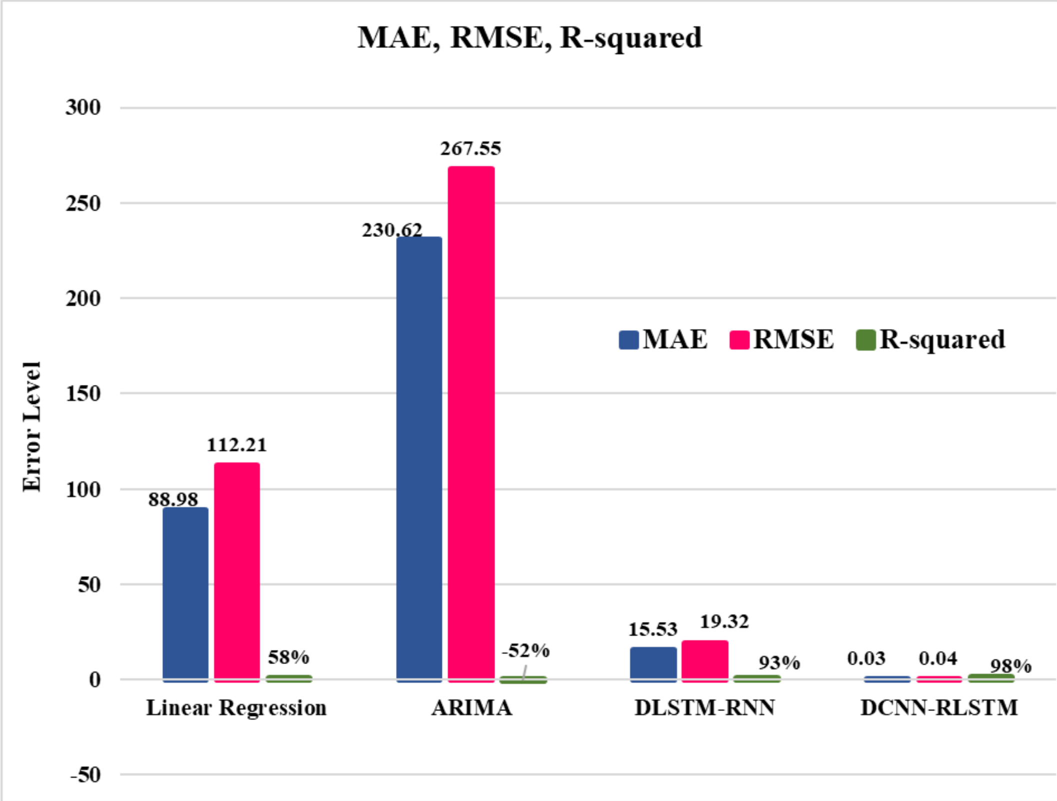

Figure 9. Robotic baseline models error metrics comparison

The expected and actual values of each method can be used to determine how well it performs. The CNN-LSTM model has the lowest MAE, RMSE, and R^2 values, whereas the ARIMA model has the highest error metrics, according to the data in Figure 9. When CNN-LSTM and linear are compared, the liner's R^2 value is low. However, the CNN-LSTM has the smallest error measurement metrics, approaching 0 to 1. The CNN-LSTM designed in this work has a lower MAE, RMSE, and R^2 following the CNN layer compared to deep LSTM-RNN. When comparing the LSTM-RNN error score with the other two machine learning models such as LR and ARIMA which have higher error scores.

Using CNN to extract data features can significantly enhance LSTM's predicting performance. According to the results, CNN-LSTM outperforms the other four methods. Hence, the investors of the firm and funding agency should take the correct predictive measure for forecasting the equity market by way of using an AI-based deep learning model that can be forecast well. Compared to the other three linear, ARIMA, and LSTM comparative models, the CNN-LSTM proposed model in this work performs better with fitting degree and error values. As such, it accurately indicates the day's closing price and is a reference point for investors. Figure 9 presents the robotic error metrics findings of the four technovative models.

The study wants to confirm the normality assumption for the Machine Learning and Deep Learning model's prediction performance comparisons using daily MAE and RMSE. It also intends to verify the robustness of the findings using a parametric (ANOVA) test. Table 6 compares the CNN-LSTM model's significant difference between the metrics against linear regression, ARIMA, and DLSTM-RNN models.

H0: There is no significant change in MAE & RMSE among the CNN-LSTM and Linear Regression algorithms.

H0: There is no significant change in MAE & RMSE among the CNN-LSTM and ARIMA algorithms.

H0: There is no significant change in MAE & RMSE among the CNN-LSTM and LSTM-RNN algorithms.

The ANOVA test consistently demonstrated statistically significant differences in various evaluation metrics (MAE, RMSE, and R^2) between CNN-LSTM and the baseline models, with lower values observed for CNN-LSTM. Therefore, the evidence from the statistical tests strongly suggests that the DCNN-RLSTM model is a superior choice for the given dataset. Also, the proposed model's capacity to capture both spatial and temporal dependencies in data leads to extra accurate and reliable predictions than the traditional statistical and Machine Learning models considered in this study.

5. Conclusion

This study uses the chronological aspects of stock price data to propose a CNN-LSTM projection of the closing price of equities for the subsequent day. This research uses the Bharti Airtel NIFTY- 100 Index's daily equity price data as a case study to confirm the experimental findings. The research proposes step-by-step backtesting from Machine Learning (Linear Regression and ARIMA) to Deep Learning (LSTM-RNN and CNN-LSTM) algorithms. In comparison to the LSTM-RNN, ARIMA, and Linear Regression models, the suggested CNN-LSTM model remained efficient throughout the procedure. In comparison to previous forecasting models, the CNN-LSTM scale-dependent error metrics of MAE, RMSE, and R^2 remain below one, at 0.03 and 0.04 respectively. By comparing the R^2 value, the CNN-LSTM is 98%, LSTM-RNN is 93%, Linear Regression is 58%, and ARIMA is -52%. This indicates that the proposed CNN-LSTM model shows higher (98%) efficiency than baseline models. The ARIMA model is not advised because it produces negative predictions. The Linear Regression model and LSTM-RNN models are less appropriate than the CNN-LSTM model. In addition, the study conducts a robust hypothesis ANOVA test for scientifically proven forecast performance of the projected CNN-LSTM model with superior predictability accuracy than the existing algorithms. Thus, CNN-LSTM is an influential tool for stock price prediction and supervised learning can improve the accuracy of share price forecasts to help society anticipate market trends and investors in society the ability to make informed decisions about their investments in advance. However, it ignores emotional variables including news and national politics.

References

[1] W. Chen, M. Jiang, W.-G. Zhang, Z. Chen, A novel graph convolutional feature based convolutional neural network for stock trend prediction. Information Sciences, 556, (2021) 67–94. https://doi.org/10.1016/j.ins.2020.12.068

[2] J.H. Stock, Forecasting economic time series. A Companion to Theoretical Econometrics, (2001) 562-84. https://doi.org/10.1002/9780470996249.ch28

[3] D.C. Montgomery, C.L. Jennings, M. Kulahci, (2015) Introduction to time series analysis and forecasting. Wiley, 672.

[4] J.M. Antony, S. Natarajan, Influence of artificial intelligence on forecasting net asset value and return volatility in Indian Mutual Fund. Advances in Artificial Intelligence and Machine Learning, 4(1), (2024) 1892–1924. https://doi.org/10.54364/AAIML.2024.41109

[5] J.M. Antony, S. Natarajan, Neural network and machine learning use cases: Indian bond market predictions. The Economics and Finance Letters, 11(1), (2024) 57–79. https://doi.org/10.18488/29.v11i1.3667

[6] M.E. Korkmaz, M.K. Gupta, E. Çelik, N.S. Ross, M. Günay, Tool wear and its mechanism in turning aluminum alloys with image processing and machine learning methods. Tribology International, 191, (2023) 109207. https://doi.org/10.1016/j.triboint.2023.109207

[7] A.S. Chivukula, W. Liu, (2017) Adversarial learning games with deep learning models. International Joint Conference on Neural Networks, IEEE, USA. https://doi.org/10.1109/IJCNN.2017.7966196

[8] F. Badrulhisham, E. Pogatzki-Zahn, D. Segelcke, T. Spisak, J. Vollert, and Machine learning and artificial intelligence in neuroscience: A primer for researchers. Brain Behavior and Immunity, 115, (2023) 470–479. https://doi.org/10.1016/j.bbi.2023.11.005

[9] X. Liu, Z. Gou, Occupant-centric HVAC and window control: A reinforcement learning model for enhancing indoor thermal comfort and energy efficiency. Building and Environment, 250, (2024) 111197. https://doi.org/10.1016/j.buildenv.2024.111197

[10] H.P. Das, Y.W. Lin, U. Agwan, L. Spangher, A. Devonport, Y. Yang, J. Drgoňa, A. Chong, S. Schiavon, C.J. Spanos, Machine learning for smart and energy-efficient buildings. Environmental Data Science, 3, (2024) e1. https://doi.org/10.1017/eds.2023.43

[11] T. Mazhar, I. Haq, A. Ditta, S.A.H. Mohsan, F. Rehman, I. Zafar, J.A. Gansau, L.P.W. Goh, The role of machine learning and deep learning approaches for the detection of skin cancer. In Healthcare, 11(3), (2023) 415. https://doi.org/10.3390/healthcare11030415

[12] M.K. Monika, N.A. Vignesh, Ch.U. Kumari, M.N.V.S.S. Kumar, E.L. Lydia, Skin cancer detection and classification using machine learning. Materials Today Proceedings, 33, (2020) 4266–4270. https://doi.org/10.1016/j.matpr.2020.07.366

[13] S. Dong, P. Wang, K. Abbas, A survey on deep learning and its applications. Computer Science Review, 40, (2021) 100379. https://doi.org/10.1016/j.cosrev.2021.100379

[14] M.F. Dixon, I. Halperin, P. Bilokon, (2020) Machine learning in finance. Springer International Publishing, New York. https://doi.org/10.1007/978-3-030-41068-1

[15] T.N. Gongada, B.K. Babu, N. Ramesh, J. Venkata, M, P. N. V. Syamala Rao, K.A. Saravanan, K. Swetha, M.A. Tripathi, Enhanced Quantitative Financial Analysis Using CNN-LSTM Cross-Stitch Hybrid Networks for Feature Integration. International Journal of Advanced Computer Science & Applications, 15(9), (2024). https://doi.org/10.14569/IJACSA.2024.0150977

[16] J. Li, S. Pan, L. Huang, A machine learning based method for customer behavior prediction. Tehnički vjesnik, 26(6), (2019) 1670-1676.

https://doi.org/10.17559/TV-20190603165825

[17] C. Xiao, W. Xia, J. Jiang, Stock price forecast based on combined model of ARI- MA-LS-SVM. Neural Computing and Applications, 32(10), (2020) 5379–5388. https://doi.org/10.1007/s00521-019-04698-5

[18] A. Sharma, D. Bhuriya, U. Singh, (2017) Survey of stock market prediction using machine learning approach. International Conference of Electronics, Communication and Aerospace Technology (ICECA), IEEE, India. https://doi.org/10.1109/ICECA.2017.8212715

[19] S. Ahmed, R.K. Chakrabortty, D.L. Essam, W. Ding, Poly-linear regression with augmented long short term memory neural network: Predicting time series data. Information science, 606, (2022) 573–600. https://doi.org/10.1016/j.ins.2022.05.078

[20] M. Yang, G. Wan, E. Zheng, A predictive DEA model for outlier detection. Journal of Management Analytics, 1(1), (2014) 20–41. https://doi.org/10.1080/23270012.2014.889911

[21] A.A. Adebiyi, A.O. Adewumi, C.K. Ayo, Comparison of ARIMA and artificial neural networks models for stock price prediction. Journal of Applied Mathematics, (2014) 1–7. https://doi.org/10.1155/2014/614342

[22] A. Dželihodžić, A. Žunić, E.Ž. Dželihodžić, Predictive modeling of stock prices using machine learning: A comparative analysis of LSTM, GRU, CNN, and RNN models. Lecture notes in networks and systems, (2024) 447–467. https://doi.org/10.1007/978-3-031-71694-2_33

[23] W. Lu, J. Li, Y. Li, A. Sun, J. Wang, A CNN-LSTM-Based model to forecast stock prices. Complexity, (2020) 1–10. https://doi.org/10.1155/2020/6622927

[24] G. Bathla, R. Rani, H. Aggarwal, Stocks of year 2020: prediction of high variations in stock prices using LSTM. Multimedia Tools and Applications, 82(7), (2022) 9727– 9743. https://doi.org/10.1007/s11042-022-12390-5

[25] C.N. Babu, B.E. Reddy, Prediction of selected Indian stock using a partitioning– interpolation based ARIMA–GARCH model. Applied Computing and Informatics, 11(2), (2015) 130–143. https://doi.org/10.1016/j.aci.2014.09.002

[26] T. Fang, R. Lahdelma, Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system. Applied Energy, 179, (2016) 544–552. https://doi.org/10.1016/j.apenergy.2016.06.133

[27] A.M. Ashik, K.S. Kannan, Time series model for stock price forecasting in India. Logistics, Supply Chain and Financial Predictive Analytics: Theory and Practices, (2019) 221-231. https://doi.org/10.1007/978-981-13-0872-7_17

[28] M.L. Challa, V. Malepati, S.N.R. Kolusu, S&P BSE Sensex and S&P BSE IT return forecasting using ARIMA. Financial Innovation, 6(1), (2020) 47. https://doi.org/10.1186/s40854-020-00201-5

[29] S. Sinha, S. Mishra, V. Mishra, T. Ahmed, Sector influence aware stock trend prediction using 3D convolutional neural network. Journal of King Saud University - Computer and Information Sciences, 34(4), (2022) 1511–1522. https://doi.org/10.1016/j.jksuci.2022.02.008

[30] M. Jiang, W. Chen, H. Xu, Y. Liu, A novel interval dual convolutional neural network method for interval-valued stock price prediction. Pattern Recognition, 145, (2024) 109920. https://doi.org/10.1016/j.patcog.2023.109920

[31] S. Maiti, R.K. Chiluvuru, "deep CNN-LSTM model for predicting interface depth from gravity data over thrust and fold belts of North East India. Journal of Asian Earth Sciences, 259, (2024) 105881. https://doi.org/10.1016/j.jseaes.2023.105881

[32] A. Onan, Bidirectional convolutional recurrent neural network architecture with group- wise enhancement mechanism for text sentiment classification. Journal of King Saud University - Computer and Information Sciences, 34(5), (2022) 2098–2117. https://doi.org/10.1016/j.jksuci.2022.02.025

[33] F. Elmaz, R. Eyckerman, W. Casteels, S. Latré, P. Hellinckx, CNN-LSTM architecture for predictive indoor temperature modeling. Building and Environment, 206, (2021)108327. https://doi.org/10.1016/j.buildenv.2021.108327

[34] N. Mughees, S.A. Mohsin, A. Mughees, A. Mughees, Deep sequence to sequence Bi-LSTM neural networks for day-ahead peak load forecasting. Expert Systems with Applications, 175, (2021) 114844. https://doi.org/10.1016/j.eswa.2021.114844

[35] C. Wang, C. Rao, F. Hu, X. Xiao, M. Goh, Risk assessment of customer churn in telco using FCLCNN-LSTM model. Expert Systems with Applications, 248, (2024) 123352. https://doi.org/10.1016/j.eswa.2024.123352

[36] N. Somu, A. Sriram, A. Kowli, K. Ramamritham, A hybrid deep transfer learning strategy for thermal comfort prediction in buildings. Building and Environment, 204, (2021)108133. https://doi.org/10.1016/j.buildenv.2021.108133

[37] W. Lu, J. Li, J. Wang, L. Qin, A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications, 33(10), (2020) 4741–4753. https://doi.org/10.1007/s00521-020-05532-z

[38] J.M.T. Wu, Z. Li, Y. Djenouri, D. Polap, G. Srivastava, J.C.W. Lin, A graphic CNN-LSTM model for stock price predication. Lecture notes in computer science, (2021) 258–268.

https://doi.org/10.1007/978-3-030-87986-0_23

[39] N. Ishwarappa, J. Anuradha, Big data based stock trend prediction using deep CNN with reinforcement-LSTM model. International Journal of Systems Assurance Engineering and Management, (2021) 1-11. https://doi.org/10.1007/s13198-021-01074-2

[40] J. Wang, X. Wang, J. Li, H. Wang, A prediction model of CNN-TLSTM for USD/CNY exchange rate prediction. IEEE Access, 9, (2021) 73346–73354. https://doi.org/10.1109/ACCESS.2021.3080459

[41] Z. Nourbakhsh, N. Habibi, Combining LSTM and CNN methods and fundamental analysis for stock price trend prediction. Multimedia Tools and Applications, 82(12), (2022) 17769–17799. https://doi.org/10.1007/s11042-022-13963-0

[42] M. Kirisci, O.C. Yolcu, A new CNN-Based model for financial time series: TAIEX and FTSE stocks forecasting. Neural Processing Letters, 54(4), (2022) 3357–3374. https://doi.org/10.1007/s11063-022-10767-z

[43] S. Mukherjee, B. Sadhukhan, N. Sarkar, D. Roy, S. De, Stock market prediction using deep learning algorithms. CAAI Transactions on Intelligence Technology, 8(10), (2021) 82–94. https://doi.org/10.1049/cit2.12059

[44] A. Yadav, C. K. Jha, A. Sharan, Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science, 167, (2020) 2091–2100. https://doi.org/10.1016/j.procs.2020.03.257

[45] A. Moghar, M. Hamiche, Stock market prediction using LSTM Recurrent Neural network. Procedia Computer Science, 170, (2020) 1168–1173. https://doi.org/10.1016/j.procs.2020.03.049

[46] J.R. Perry, (2023) Predictive AI for the S&P 500 Index. Computer Science Senior Theses, 23.

[47] Q.Q. Abuein, M.Q. Shatnawi, E.Y. Aljawarneh, Y.E. Manasrah, A. Ahmed Time Series Forecasting Model for the Stock Market using LSTM and SVR. International Journal of Advances in Soft Computing and its Applications, 16(10), (2024) 1852024.

[48] M.M. Billah, A. Sultana, F. Bhuiyan, M.G. Kaosar, Stock price prediction: comparison of different moving average techniques using deep learning model. Neural Computing and Applications, 36(11), (2024) 5861–5871. https://doi.org/10.1007/s00521-023-09369-0

[49] R.M. Dhokane, S. Agarwal, A Predictive Model of the Stock Market Using the LSTM Algorithm with a Combination of Exponential Moving Average (EMA) and Relative Strength Index (RSI) Indicators. Journal of the Institution of Engineers (India) Series B, 105, (2024) 1145–1157. https://doi.org/10.1007/s40031-024-01037-8

[50] P. Mondal, L. Shit, S. Goswami, Study of Effectiveness of Time Series Modeling (ARIMA) in forecasting stock Prices. International Journal of Computer Science Engineering and Applications, 4(2), (2014) 13–29. https://doi.org/10.5121/ijcsea.2014.4202

[51] M. Dadhich, M.S. Pahwa, V. Jain, R. Doshi, Predictive Models for Stock Market Index using Stochastic Time Series ARIMA Modeling in Emerging Economy. Lecture notes in mechanical engineering, (2021) 281–290. https://doi.org/10.1007/978-981-16-0942-8_26

[52] U.M. Sirisha, M.C. Belavagi, G. Attigeri, Profit prediction using ARIMA, SARIMA and LSTM models in Time Series Forecasting: A comparison. IEEE Access, 10, (2022) 124715–124727. https://doi.org/10.1109/ACCESS.2022.3224938

[53] T. Muhammad, A.B. Aftab, M. Ibrahim, M.M. Ahsan, M.M. Muhu, S.I. Khan, M.S. Alam, Transformer-based deep learning model for stock price prediction: A case study on Bangladesh stock market. International Journal of Computational Intelligence and Applications, 22(03), (2023) 2350013. https://doi.org/10.1142/S146902682350013X

[54] J.M. Sangeetha, K.J. Alfia, Financial stock market forecast using evaluated linear regression based machine learning technique. Measurement Sensors, 31, (2023) 100950. https://doi.org/10.1016/j.measen.2023.100950

[55] S. Antad, S. Khandelwal, A. Khandelwal, R. Khandare, P. Khandave, D. Khangar, R. Khanke, Stock Price Prediction Website Using Linear Regression-A Machine Learning Algorithm. In ITM Web of Conferences, 56, (2023) 05016. https://doi.org/10.1051/itmconf/20235605016

[56] D.T. Larose, C.D. Larose, (2014) Discovering knowledge in data: an introduction to data mining John Wiley & Sons, 336. https://doi.org/10.1002/9781118874059

[57] J.S. Long, J. Freese, (2006) Regression models for categorical dependent variables using Stata. Stata press.

[58] G. Maji, D. Mondal, N. Dey, N.C. Debnath, S. Sen, Stock prediction and mutual fund portfolio management using curve fitting techniques. Journal of Ambient Intelligence and Humanized Computing, 12(10), (2021) 9521–9534. https://doi.org/10.1007/s12652-020-02693-6

[59] S. Seabold, J. Perktold, Statsmodels: Econometric and Statistical Modeling with Python. Proceedings of the Python in Science Conferences, 7(1), (2010). https://doi.org/10.25080/Majora-92bf1922-011

[60] Y. Lin, Y. Yan, J. Xu, Y. Liao, F. Ma, Forecasting stock index price using the CEEMDAN-LSTM model. The North American Journal of

Economics and Finance, 57, (2021) 101421. https://doi.org/10.1016/j.najef.2021.101421

[61] W. Zhang, X. Gong, C. Wang, X. Ye, Predicting stock market volatility based on textual sentiment: A nonlinear analysis. Journal of Forecasting, 40(8), (2021) 1479–1500. https://doi.org/10.1002/for.2777

[62] O.B. Sezer, A.M. Ozbayoglu, Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Applied Soft Computing, 70, (2018) 525–538. https://doi.org/10.1016/j.asoc.2018.04.024

[63] Z. Wang, Z. Hu, F. Li, S.B. Ho, E. Cambria, Learning-Based stock trending prediction by incorporating technical indicators and social media sentiment. Cognitive Computation, 15(30), (2023)1092–1102. https://doi.org/10.1007/s12559-023-10125-8

[64] J. Zhang, L. Ye, Y. Lai, Stock price prediction using CNN-BILSTM-Attention model. Mathematics, 11(9), (2023) 1985. https://doi.org/10.3390/math11091985

Acknowledgment

We are grateful to the editors and reviewers for many constructive and insightful comments. The authors would like to acknowledge VIT, Vellore, for providing all the essential facilities for this study.

Authors Contribution Statement

A. Jesmine Mary: Conceptualization, Investigation, Literature review, Methodology, Data Analysis, and Writing -Original draft. N. Sundaram: Conceptualization, Methodology, Critical review, Editing, and Supervision.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes