Received: 20 February 2021 | Revised: 12 June 2021 | Accepted: 25 July 2021

CAAI Transactions on Intelligence Technology

DOI: 10.1049/cit2.12059

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Stock market prediction using deep learning algorithms

Somenath Mukherjee[1] | Bikash Sadhukhan[2] | Nairita Sarkar[2] | Debajyoti Roy[2] | Soumil De[2]

[1]Nazrul Center of Social and Cultural Studies, Kazi Nazrul University, Asansol, India

[2]Department of Computer Science and Engineering, Techno International New Town (Techno India College of Technology), Kolkata, India

Correspondence

Bikash Sadhukhan, Department of Computer Science and Engineering, Techno International New Town (Techno India College of Technology), 1/1, Service Rd, DG Block(Newtown), Action Area I, Newtown, Chakpachuria, West Bengal 700156, Kolkata, India.

Email: bikash.sadhukhan@tict.edu.in

Abstract

The Stock Market is one of the most active research areas, and predicting its nature is an epic necessity nowadays. Predicting the Stock Market is quite challenging, and it requires intensive study of the pattern of data. Specific statistical models and artificially intelligent algorithms are needed to meet this challenge and arrive at an appropriate solution. Various machine learning and deep learning algorithms can make a firm prediction with minimised error possibilities. The Artificial Neural Network (ANN) or Deep Feed-forward Neural Network and the Convolutional Neural Network (CNN) are the two network models that have been used extensively to predict the stock market prices. The models have been used to predict upcoming days' data values from the last few days' data values. This process keeps on repeating recursively as long as the dataset is valid. An endeavour has been taken to optimise this prediction using deep learning, and it has given substantial results. The ANN model achieved an accuracy of 97.66%, whereas the CNN model achieved an accuracy of 98.92%. The CNN model used 2-D histograms generated out of the quantised dataset within a particular time frame, and prediction is made on that data. This approach has not been implemented earlier for the analysis of such datasets. As a case study, the model has been tested on the recent COVID-19 pandemic, which caused a sudden downfall of the stock market. The results obtained from this study was decent enough as it produced an accuracy of 91%.

KEYWORDS

artificial neural network, convolutional neural network, nifty, stock market

1 | INTRODUCTION

The stock market refers to the collection of markets and exchange centres where economic activities like buying, selling, and deploying shares of publicly held companies take place. Such financial practices are conducted through institutionalised formal exchanges through over-the-counter marketplaces that function under a defined set of regulations. The stock market is a very dynamic and uncertain field, so the stock market's prediction naturally becomes a burning topic. Due to the advancement of computational power in recent times, predicting the stock market has been much faster and accurate. Artificial Intelligence and machine learning models play a crucial role in predicting the stock prices and, hence, determining an accurate result.

A large portion of the Indian stock market trading occurs on two stock exchanges: the Bombay Stock Exchange (BSE) and the National Stock Exchange (NSE). Both the exchanges pursue the same trading method, trading hours, and settlement process. BSE has more than 5000 listed firms, whereas its counterpart NSE has about 1600 enlisted firms. Out of all the listed firms on the BSE, only about 500 firms constitute more than 90% of its market capitalisation; the rest consists of highly illiquid shares. Almost all the major and giant trade firms of India are enlisted on both exchanges. Both exchanges compete for the order flow that results in reduced costs, market efficiency, and innovation. The presence of arbitrageurs keeps the prices on the two stock exchanges within a very close range.

The stock market is frequently used as a sentiment indicator and can have an impact on GDP (gross domestic product). GDP is a metric that measures an economy's total output of goods and services. As the stock market rises and falls, so does economic sentiment [1]. People's spending fluctuates in response to changes in emotion, which drives GDP growth [2]. The stock market, on the other hand, can have both positive and negative effects on GDP. GDP is usually expressed as a percentage increase from one period to the next. For example, if the quarter-to-quarter growth rate is 2%, that means the economy increased by 2% on an annualised basis in that quarter. It is important to remember that India's economy is mostly based on consumption. Private consumption spending, or money spent by the residents on goods and services, has accounted for approximately three-fifths of the Indian economy through the years [3].

Figures 1 and 2 describe the growth of the Indian Gross Domestic Product (GDP), which is analysed through machine learning algorithms. It shows the predictions on the GDP, leading to a better understanding of any country's socio-economic situation. Machine learning and deep learning algorithms [4] play a vital role in predicting a nation's economy. These predictions are essential for making correct financial decisions [5]. The motive of this study is to provide accurate predictions of the stock market values for consecutive days, even during extreme market fluctuations like in the COVID-19 pandemic during March and April 2020.

The whole work is categorised into several sections. The Related Work and the experimental dataset are discussed in Sections 2 and 3, followed by Section 4, which presents the proposed model used to generate the predictions. Section 5 gives a detailed description of the system configuration used for this study. The empirical and graphical results of the predictions, along with a case study, are shown in Section 6. Section 7 argues the advantages of the CNN method over the backpropagation ANN method and finally Section 8 provides the concluding remarks.

2 | RELATED WORK

Several researchers implemented their work to provide accurate solutions to this dynamic problem and have proposed various methods for predicting the stock market. Jayanth Balaji [6] performed a deep learning method to predict a company's stock price using 14 different deep learning methods. Similar work is implemented using Artificial Neural Networks (ANN) by Tsong Wuu Lin [7, 8]; his work tried to maximise the profitability using this model [9]. Autoregressive models are powerful models for predicting the stock market, they give a strong insight on time series analysis and make very accurate predictions [10, 11]. Sentiment analysis is also one of the strong ways to predict the stock market. Social media analytics plays a vital role in sentiment analysis. ARIMA model helps in

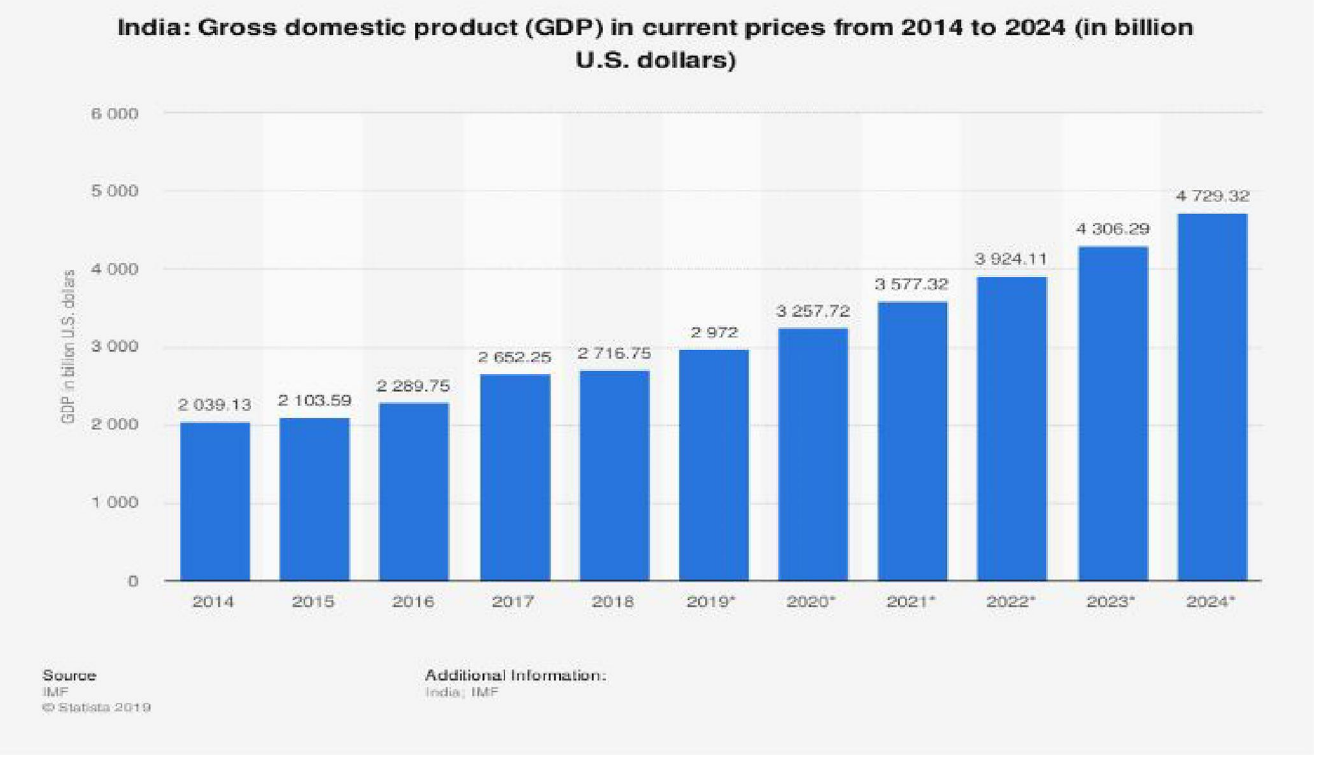

FIGURE 1 The pattern of growth of the Indian Gross Domestic Product (GDP) analysed through machine learning

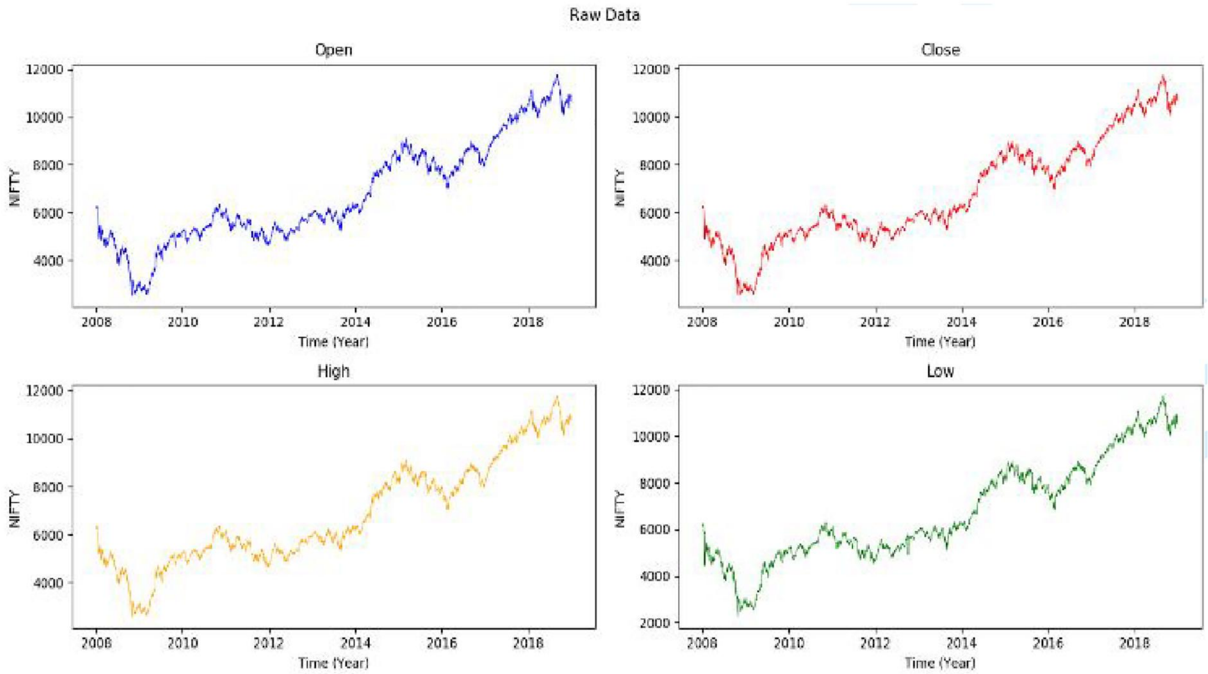

FIGURE 2 The time-series data that has been used in this study

sentiment analysis and predicting time series data [12–14]. Sentiment analysis can also be implemented by using deep learning models like CNN and LSTM [15]. For better accuracy and better use of features, deep learning methods are preferred over supervised machine learning methods. Boosted Decision Tree model [16, 17]and ELSTM model [18, 19] give valuable insight on the empirical results of the predictions but are unable to provide accurate results during fluctuating scenarios like in the COVID-19 case. Along with deep learning algorithms, a powerful model known as convolution neural network (CNN) is advantageous and accurate to solve problems of this genre [20, 21]. CNN plays a significant role in the fields of image processing, pattern recognition, and in analysing time series data. Traditionally, a Convolutional Neural Network input is often a 2-D image, but that does not mean this model cannot be used for predicting time series data [22]. There are two ways to pre-process the data, one way is to convert the 1-D-input data into a 2-D matrix, and the other way is to take advantage of the 1-D function to help do the convolutional computation [23].

Based on the above discussions, an endeavour has been taken to implement deep learning methods to predict the nature of stock market prices. National Stock Exchange (NSE) [24]stock market dataset is used for predicting the stock market values [25]. As the National Stock Exchange (NSE) dataset provides short settlement cycles and very high transaction time, the National Stock Exchange dataset can be computationally efficient for analysis purposes. Two conventional machine learning approaches are implemented. The first approach uses the backpropagation algorithm on a simple artificial neural network. While this approach is quite efficient in predicting future price values, another approach has been implemented, using a convolutional neural network (CNN) model. This model generates 2-D histograms and feeds them to the neural network. Such an approach increases the efficiency of training as well as prediction. LSTMs and Transformer models are used to generate context from sequences which are long. The efficiency of the LSTM models and Transformer models are significantly reduced when the sequence length is reduced. [26–29]. The sequence length used in the CNN model is considerably lesser than the ones used for LSTM and transformer models. The problem that has been addressed here is a regression problem but a model that has mostly been used for classification problems has been used for this purpose, that is without implementing known Recurrent Neural Networks like LSTM and Transformers, an endeavour has been taken to implement the same using CNN so that we can prove that we can obtain results of similar stature too. It has given accurate and concrete results even during extreme stock market fluctuations like the COVID-19 pandemic situation in March and April 2020. As compared to two recent works [30, 31], our proposed model gives an accuracy of 97.66% using Backpropagation Artificial Neural Network and 98.92% using Convolutional Neural Networks.

2.1 | Backpropagation on feed-forward neural network

The most fundamental unit of an Artificial Neural Network is an artificial neuron that takes in multiple inputs, multiplies them by the weights assigned, and adds a bias. The result is fed to an activation function whose output determines to what level the neuron will be triggered, and the corresponding value is passed on to the neurons in the next layer.

Figure 3 describes the model of an artificial neuron where $\{ x_1, x_2, x_3,\ldots, x_n \}$ is the set of inputs, $\{ w_1, w_2, w_3, \ldots,w_n, \}$ is the set of weights associated with the input nodes, φ is the activation function, and y is the output that can be calculated by applying the activation function over the net input.

The following equation governs the firing of an artificial neuron:

$$y = \varphi\left(\left(\sum_{i=1}^{n} x_i w_i\right) + bias\right) \tag{1}$$

x_i is the neurons' inputs in the previous layer, and w_i are the corresponding weights. φ is the activation function.

Considering an ANN with two hidden layers along with an input layer and an output layer, the equations would be as follows:

Notations:

- A^k represents the matrix containing the output of the neurons after activation at layer k.
- W^k represents the weight matrix in-between layer k-1 and k.
- b^k represents the bias matrix at hidden layer k.
- Z^k represents the matrix containing the neuron value at layer k.

Input layer:

$$x = a^1$$

Neuron value at hidden layer 1:

$$z^2 = w^1 x + b^1 \tag{2}$$

Activation value at hidden layer 1:

$$a^2 = \varphi(z^2)$$

Neuron value at hidden layer 2:

$$z^3 = w^2 a^2 + b^2 \tag{3}$$

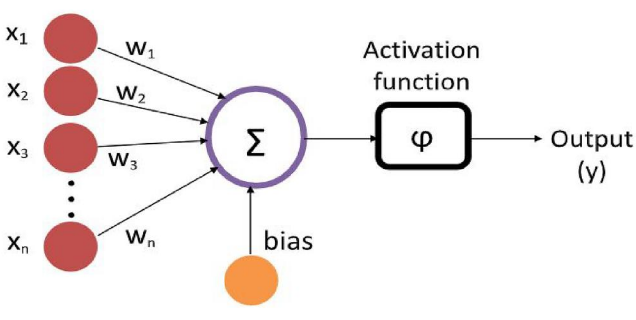

FIGURE 3 An artificial neuron

Activation value at hidden layer 2:

$$a^3 = \varphi(z^3)$$

Output layer:

$$y = w^3 a^3 \tag{4}$$

Once the output is generated, a cost function is evaluated to analyse the difference between the actual and predicted outputs.

Cost/Loss function:

$$Cost(C) = \frac{1}{N} \sum_{i=1}^{N} \left(y_{predicted} - y_{actual}\right)^2 \tag{5}$$

where, N is the number of inputs' combinations that have been fed forward through the network.

This cost function is used to backpropagate the errors and in turn tweak the weights and biases present in the network to minimise the cost function for better prediction. The concept of gradient descent is used to achieve it. The sensitivity of the cost function concerning a particular parameter is analysed by calculating the cost function's gradient for that parameter. The parameter is then tweaked accordingly.

$$x_{updated} = x - \eta \frac{\partial C}{\partial x} \tag{6}$$

where, η signifies the learning rate, C signifies the cost function, and x signifies any parameter (weight or bias) in the network.

This process of forwarding propagation and backpropagation continues until the cost function is minimised and the model can give accurate results.

In this project, the backpropagation algorithm is implemented on a deep feed-forward neural network to predict the upcoming days' time-series data value from the previous two days' data values.

2.2 | Convolutional neural network (CNN)

A convolutional neural network (CNN, or Conv-Net) is a type of deep neural network [33, 34] that is most typically used to analyse and conduct Artificial Intelligence operations on visual images.

Because of their shared-weights architecture and translation invariance qualities, they are also known as shift invariant or space invariant artificial neural networks (SIANN). They have image and video recognition applications, recommender systems, image classification, medical image analysis, and natural language processing.

The name "convolutional neural network" signifies that the neural network uses a mathematical operation known as convolution. Convolution is a specialised kind of linear operation. Convolutional networks are simply neural networks that employ convolution instead of general matrix multiplication in at least one of their layers.

A convolutional neural network consists of an input and an output layer and multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional and pooling layers arranged in a particular pattern that is eventually flattened and followed by a fully connected network. These layers are referred to as the hidden layers because the activation function and final convolution mask their inputs and outputs. The final convolution, in turn, often involves backpropagation to weight the end product more accurately.

The convolution layers consist of filters or kernels of specific dimensions smaller than the image dimensions, passed over the entire image to extract the features. The filters may be pre-determined like the ones to detect edges and various shapes or may be learnt by the model in the training process. A stride length is also applied to the image along with the filter. Stride is the number of pixels that can be shifted over the input image. Figure 4 shows the convolution operation on the image with the 7 × 7 dimension with filter size 3 × 3 and stride length = 2.

Pooling layers are introduced to extract essential features or downsize the extensive features extracted by the convolution layers. In most cases, max-pooling or average-pooling is used. The fully connected network then works on the reduced number of features hence producing efficient predictions quickly. Figure 5 describes the max pooling operation with filter size 3 × 3 and stride length = 2.

This study focusses on generating synthetic images in 2-D histograms [35] from stock market data spanning over a few days, passed to a convolution neural network to predict the time-series data value (stock prices) of the following day. The convolution neural network (CNN) model gives a new angle to look at stock market data and gives greater accuracy in the prediction. The system configuration for implementing these models is discussed in the section below.

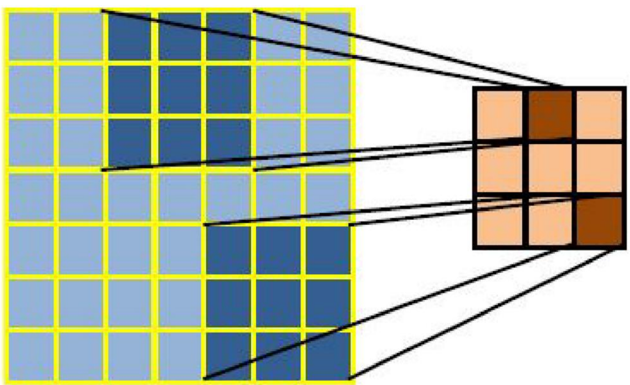

FIGURE 4 Convolution operation with kernel size = (3 × 3) and stride length = 2

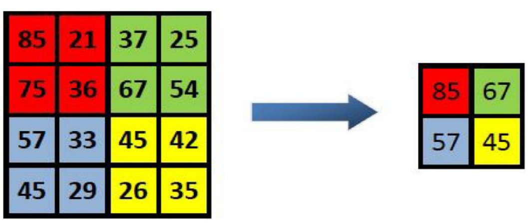

FIGURE 5 Max pooling with kernel size = (2 × 2) and stride length = 2

3 | EXPERIMENTAL DATASET

The experimental dataset that has been used is the National Stock Exchange (NSE) stock market dataset, specifically the NIFTY price index, ranging in the time frame of April 2008 to April 2018, collected from the National Stock Exchange (NSE) India website [32]. The recent dataset of NSE India from 1 November 2019 to 5 August 2020 is taken into account from the same source as a case study. The working dataset has seven features in it, out of which five features have been taken into consideration for this project. The features that have been taken into account are 'Date,' 'Open,' 'Close,' 'High,', 'Low'. A sample from the original dataset used in the work has been provided below in Table 1. Initially, all the features are plotted to generate a pattern on how the stock data moves along with time. The plots of each feature are shown above in Figure 2. The approach using 2-D histograms has not been implemented earlier; hence, an endeavour has been taken to predict a day's stock market prices based on the recent past data. These approaches proved to work well with the prediction of time-series data.

Based on this movement pattern, two deep learning models have been devised to predict future data values considering the present values. The models that have been used in this study are discussed in the following section.

4 | PROPOSED MODEL

4.1 | Using the ANN with backpropagation algorithm

This study implements the backpropagation algorithm on a deep feed-forward neural network to predict stock market data.

The model involves predicting the 'Open,' 'Close,' 'High,' and 'Low' values of NIFTY of the nth day based on these four

TABLE 1 Raw data sample

Date	Open	High	Low	Close
03-Jan-08	6184.25	6230.15	6126.4	6178.55
04-Jan-08	6179.1	6300.05	6179.1	6274.3
07-Jan-08	6271	6289.8	6193.35	6279.1
08-Jan-08	6282.45	6357.1	6221.6	6287.85
09-Jan-08	6287.55	6338.3	6231.25	6272
10-Jan-08	6278.1	6347	6142.9	6156.95

features of the $(n-1)^{th}$ and $(n-2)^{th}$ days. The model is designed to have eight input neurons to accept the values of each of the four features of the previous two days, as shown in Figure 6. The input layer is followed by two hidden layers consisting of 2 neurons, each, in turn, followed by the output layer with four nodes to predict the following day's values. The number of hidden layers and the number of neurons in each of the two hidden layers were fixed at two after careful experimentation with higher values, which often led to over-fitting and decreased prediction accuracy. The activation function that was used in the particular model is the sigmoid function. The model was trained with 2000 input combinations for more than 10,000 epochs with a learning rate of 0.1. The model was then tested with more than 2500 input combinations of stock price data.

The accuracy of each prediction was calculated using the following formula:

$$accuracy_i(in\ \%) = 1 - \left(\frac{|target_i - output_i|}{target_i}\right) * 100 \qquad (7)$$

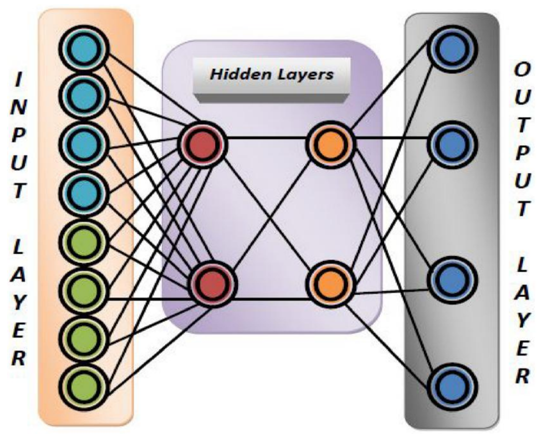

FIGURE 6 ANN model used in this prediction

4.2 | Using the convolutional neural network with 2-D histograms

The study was further taken to the implementation of deep convolution neural networks in the field of stock market prediction.

4.2.1 | Network Model

The network model designed for the prediction is shown in Figure 7. The model takes the 20 × 20 matrix generated from a 2-D histogram as input and generates a fraction as output. The predicted fraction multiplied with the $(n-1)^{th}$ day gives the predicted data value of the n^{th} day.

4.2.2 | Training

This CNN model's training process was much faster and efficient than the previously discussed ANN model. Only 1000 input combinations and just about 150 epochs were sufficient for the desired training, which is way less than what was required for the ANN model. The Adam optimiser was used for the training, and a learning rate of 0.01 proved to be good enough for proper convergence of the cost function. The cost function that was used is the Mean Squared Error (MSE). Both MSE and Mean Absolute Error (MAE) were used to judge the model's performance during the training.

$$MSE = \frac{1}{n}\sum (y - \hat{y})^2$$

$$MAE = \frac{1}{n}\sum |y - \hat{y}|$$

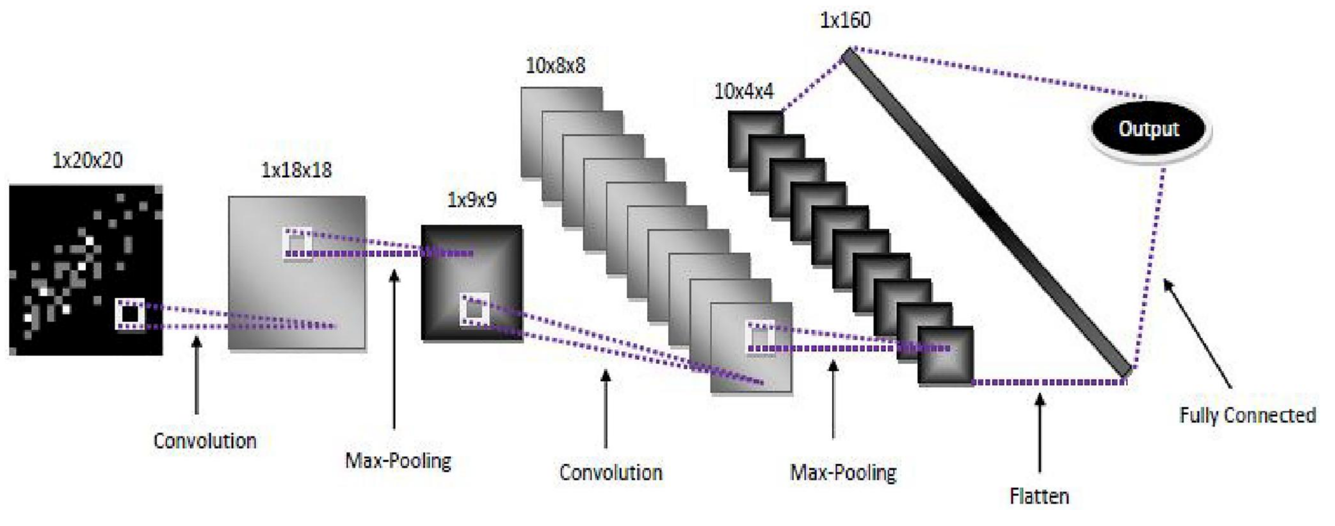

FIGURE 7 CNN model used in this study

Here n is the number of training examples, y is the actual output value, and $\hat{y}$ is the predicted output value.

5 | EXPERIMENTAL SETUP AND SIMULATION

The models were trained and tested using the Keras API on top of Theano in Python version 3.6.8. A machine with the Windows 7 Operating System was used for the entire study. The system used a dual-core Intel(R) Core (TM) 2 Duo processor coupled with 2 GB RAM and 829 MB of Graphics memory. The Graphics chipset used was an Intel(R) 4 Series Express Chipset. This hardware and software configuration took around 3 h to train the Artificial Neural Network model for 10,000 epochs, and it took about 1 h 30 min to train the CNN model for 150 epochs. The results obtained after implementing the model on this system is discussed below.

5.1 | Image Preparation

A 2-dimensional histogram also known as a density heatmap is generated by grouping into bins multiple points according to their x and y coordinates. In a 1-dimensional histogram, the frequency of points belonging to a bin is characterised by the bar's height corresponding to that bin. Whereas in a 2-D histogram, the colour corresponding to a bin signifies the frequency of points mapped to that bin. Thus, 2-D histograms are used to visualise the density of points in a 2-D plane. In this study, a 2-D histogram has been generated from the stock market data. Here, a point in the 2-D plane represents the stock price index of two consecutive days. The x-coordinate gives the price index of day one, and the y-coordinate gives the price index of day two. Fifty such sets of points are considered for generating a single 2-D histogram. A greyscale 2-D histogram has been used in this study. Hence, the whiter a bin, the denser it is. A 2-D histogram explains how the price index varied over consecutive days for the last 50 days. This insight helps to predict the stock price index of the next day.

The model that was designed can predict the 'Open,' 'Close,' 'High,' and 'Low' values of NIFTY, or in general, any time-series data value of the next day based on the corresponding values of the last 51 days.

In our work, a 2-D greyscale histogram is generated by plotting any of the 'Open,' 'Close,' 'High,' and 'Low' values of the $(n-51)^{th}$ day to $(n-2)^{th}$ day on the x-axis and the values of the corresponding succeeding day $((n-50)^{th}$ day to $(n-1)^{th}$ day) in the y-axis. Each 2-D histogram has 20 bins along the two axes, as shown in Figure 8a and b. Each such 2-D histogram gives rise to a 20×20 matrix where the $(i,j)^{\text{th}}$ cell corresponds to the pixel values in the ith and jth bins' intersection along the x and y-axis, respectively. Each such matrix is fed to the deep convolution network to predict the nth day's corresponding feature.

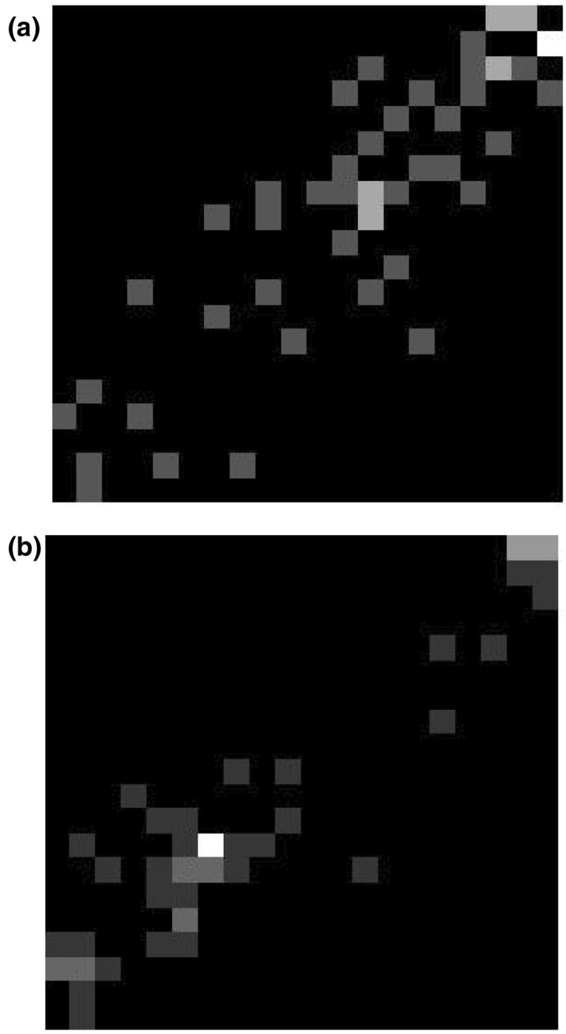

FIGURE 8 (a) and (b). Sample 2-D histograms generated from the time-series data

6 | RESULTS AND ANALYSIS

6.1 | Using the ANN with backpropagation algorithm

The model was tested with more than 2500 input combinations of stock price data, and it predicted the outputs with an accuracy of 97.66%.

The time versus feature plot shows how the data moves according to the prescribed model. It also correlates the model with the expected output against the received output. Its accuracy layup structure has accompanied each graph.

The time versus feature graphs based on the ANN model for the features 'Open,' 'Close,' 'High,' and 'Low' is shown in Figure 9a–d respectively.

Figure 10a–d shows the time versus prediction accuracy layup of the features 'Open,' 'Close,' 'High,' and 'Low,' respectively, based on the ANN model.

The above graphs clearly show that the model's predictions were very close to the actual values. The 'Open'

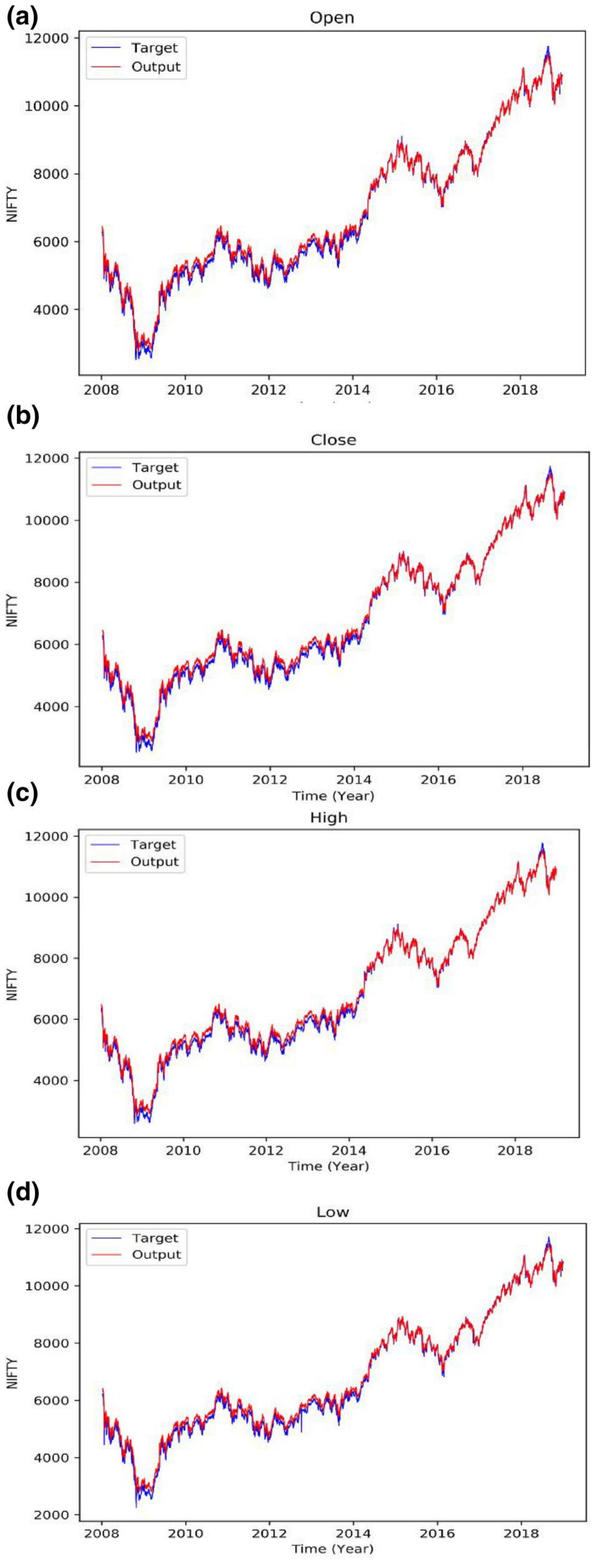

FIGURE 9 (a–d). The prediction of all the features based on the ANN model

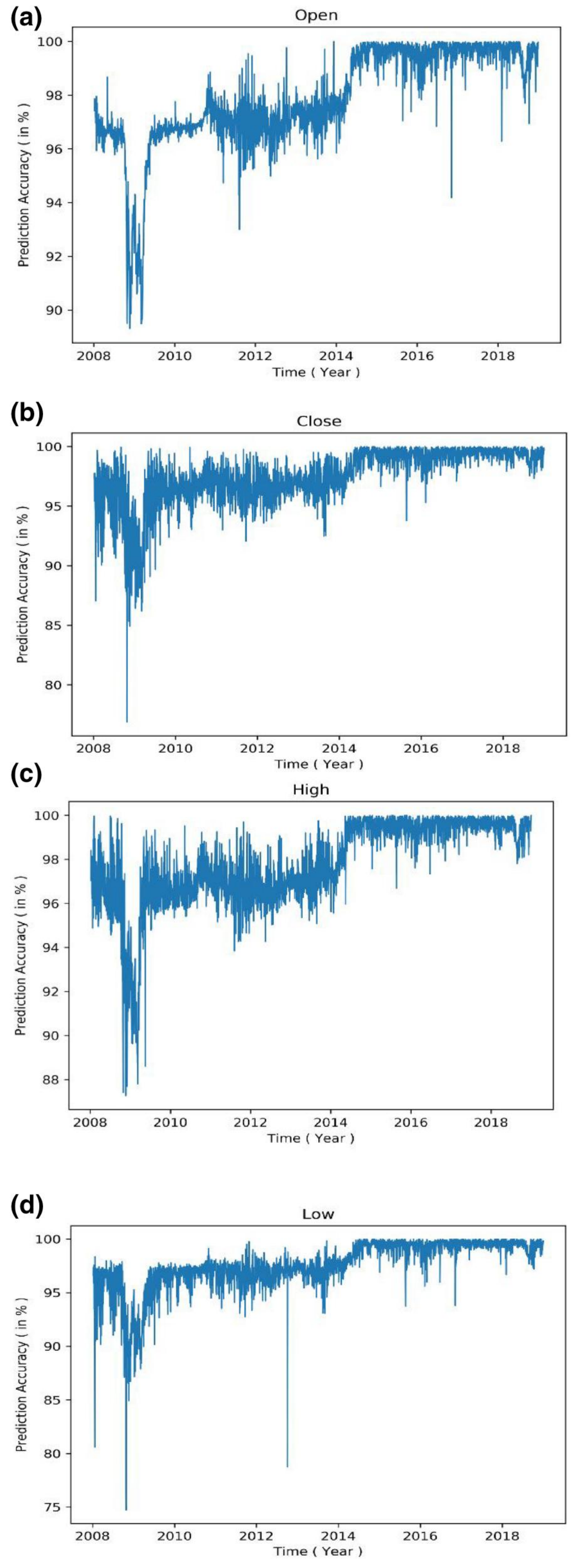

FIGURE 10 (a–d). The accuracy layup of all the features based on the ANN model

feature was predicted with an average accuracy of 97.87%, followed by the 'High,' 'Low,' and 'Close' features with accuracies of 97.77%, 97.53%, and 97.47%, respectively. Next, the same predictions were carried out using Convolutional Neural Networks to obtain even better results.

6.2 | Using the convolutional neural network with 2-D histograms

The MSE and MAE metrics were plotted against the number of epochs, and they converged to sufficiently small values, as shown in Figure 11 below.

6.2.1 | Testing

The model was tested with more than 1600 input combinations or matrices. The training data of 1000 data points was also put to test for analysing the performance of the model based on known and unknown data points. The accuracy was calculated in the same way as it was done for the ANN model. Accuracy on the training set and the testing set was found to be close. The 'Low' feature was predicted most accurately with a prediction accuracy of 99.06%, followed by 'Close' at 99.03%, 'High' at 98.94%, and 'Open' at 98.63%.

The results that were obtained after the prediction of the features 'Open,' 'Close,' 'High,' and 'Low' are depicted in Figure 12a–d, respectively.

Figure 13a–d show time versus prediction accuracy layup of the features 'Open,' 'Close,' 'High,' and 'Low,' respectively, based on the CNN model.

The prediction graphs show that the CNN model used the 2-D histograms as input predicted even better than the ANN model. The 'target' and 'output' plots seem to almost coincide with each other. The accuracy graphs validate the same as we see that the accuracy remained between 98% and 100% for most of the time.

6.3 | Quantisation table

Table 2 shows that a clear picture is obtained on how the accuracy of the CNN model on the given dataset is obtained. The entire dataset is broken down into 15 segments randomly. The CNN model is implemented on each segment to determine the highest accuracy in that particular segment. In each segment, the maximum and minimum value of the feature ('High' and 'Low') is taken into consideration within that time frame, which is then compared with the corresponding predicted maximum and minimum value. Based on this, the accuracy is generated. Taking the 'High' and 'Low' features into consideration, the CNN model's accuracy was found to be 99.34%. The total accuracy is calculated by only taking the arithmetic mean of the accuracy of all segments.

Statistically, overfitting can be defined as producing an analysis that corresponds precisely to a particular dataset and may therefore fail to fit additional data or fail to predict future observations on unknown data reliably. An overfitted model is a statistical model that contains more parameters than that can be justified by the data and hence has higher model complexity. The essence of overfitting is to have unknowingly extracted some of the residual variations as if that variation represented the underlying model structure.

Regularisation is a process in which extra information is added to the current analysis to solve an ill proposed (incorrect model usage) problem or prevent overfitting. The Regularisation method is used extensively in this study to address the problem of overfitting. The model is experimented with a range of λ values, which is known as the regularisation parameter. An optimal λ value was chosen based on the

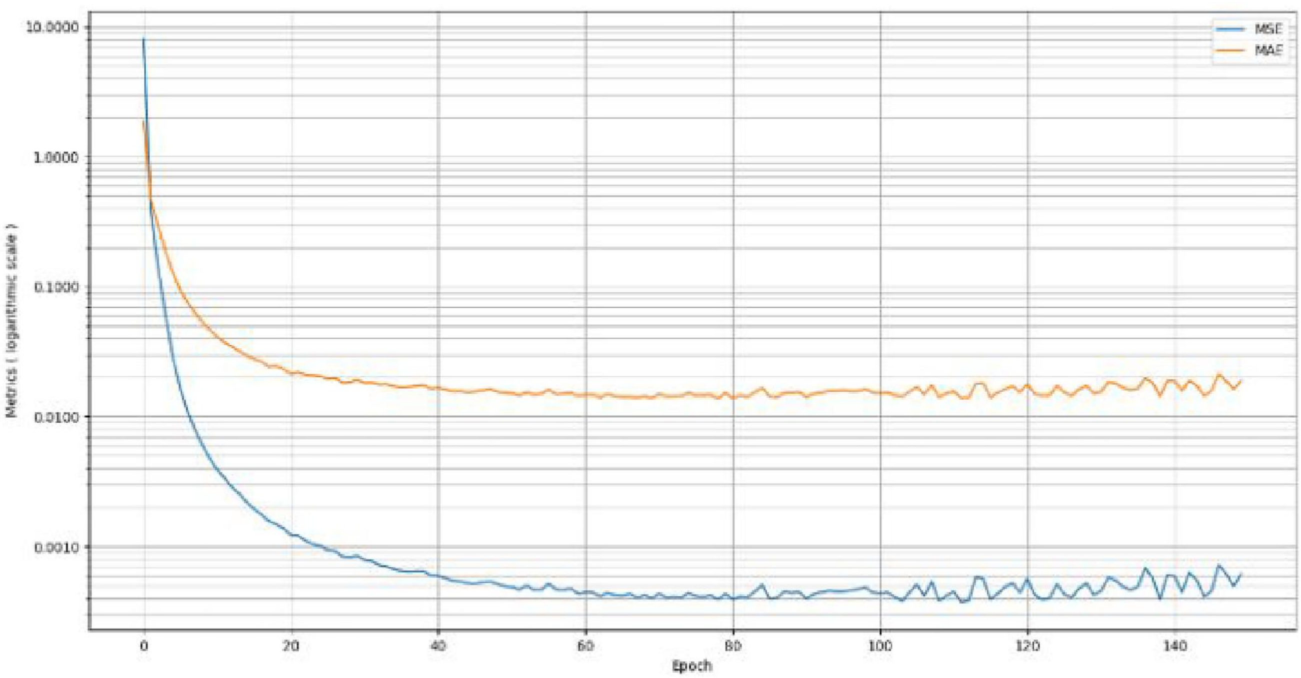

FIGURE 11 The convergence of Mean Squared Error (MSE) and Mean Absolute Error (MAE)

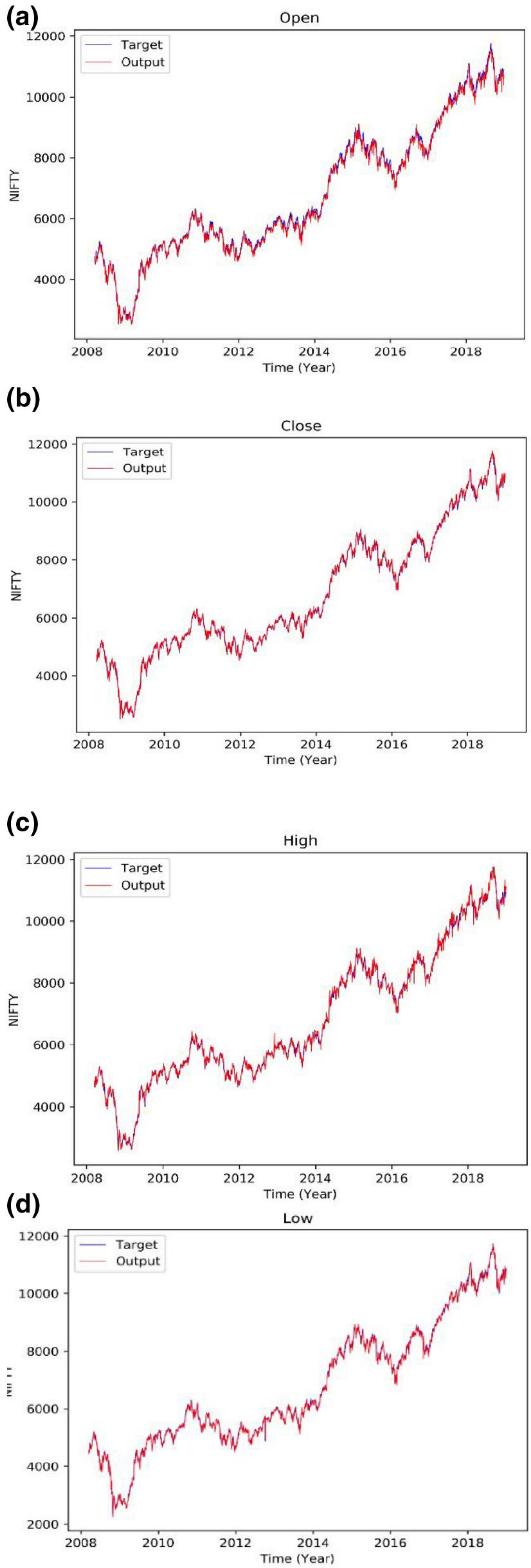

FIGURE 12 (a–d). The prediction of all features based on the CNN model

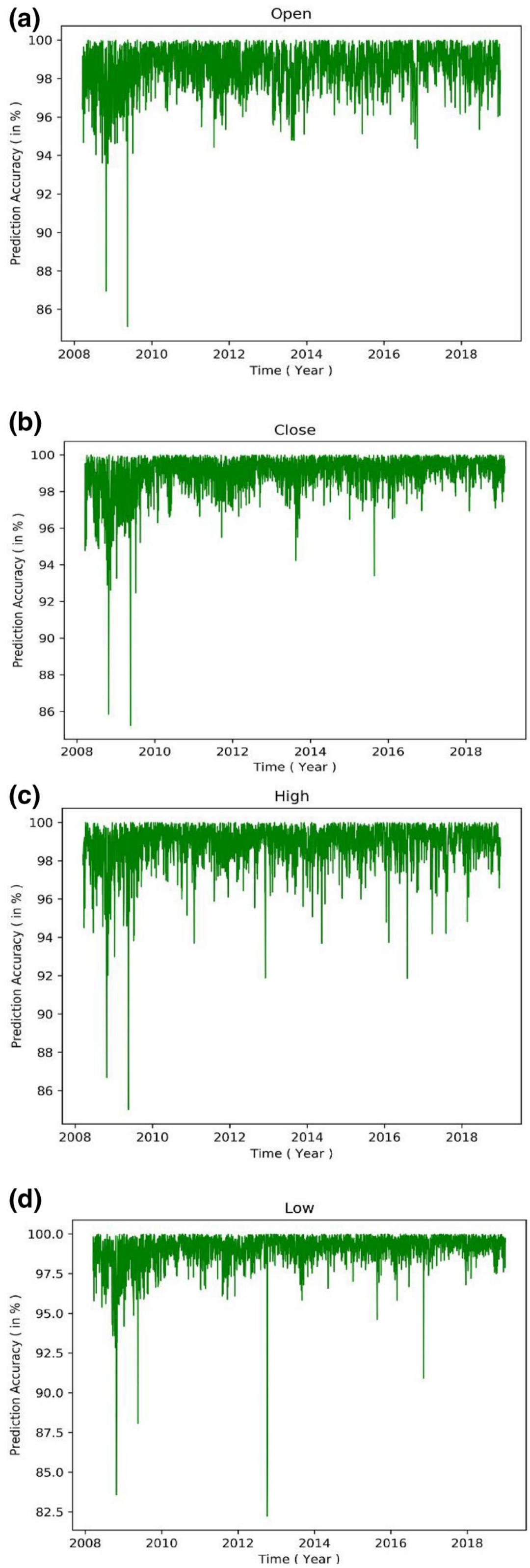

FIGURE 13 (a), (b), (c), (d). The accuracy layup of all the features based on the CNN model

performance of the model on the cross-validation set. This eliminated the chances of the model having high variance. The model's training accuracy and testing accuracy had a negligible difference and were found to be sufficiently high. This finding validates the fact that the model neither suffered from overfitting nor underfitting.

TABLE 2 Accuracy determination by quantisation of the dataset into 15 segments

Stock market data	'High' price index (Rs)			'Low' price index (Rs)		
	Actual value	Predicted value	Accuracy (%)	Actual value	Predicted value	Accuracy (%)
(Apr-2008)–(Nov-2008)	5298.85	5297.73	99.97886334	2252.75	2251.84	99.95960493
(Dec-2008)–(Jun-2009)	4693.2	4666.45	99.43002642	2539.45	2541.03	99.93778180
(Jul-2009)–(Dec-2009)	5221.85	5183.20	99.25984086	3918.75	3886.11	99.16708134
(Jan-2010)–(Jul-2010)	5447.5	5477.13	99.99324509	4675.4	4658.97	99.64858622
(Aug-2010)–(Apr-2011)	6338.5	6436.02	98.46146565	5177.7	5176.29	99.79894548
(May-2011)–(Dec-2011)	5775.25	5845.08	98.79087485	4531.15	4498.00	99.26839765
(Jan-2012)–(Aug-2012)	5629.95	5639.80	99.82504285	4675.8	4610.96	98.61328543
(Sep-2012)–(May-2013)	6229.45	6377.19	97.62836205	4888.2	4870.71	99.64219958
(Jun-2013)–(Jan-2014)	6415.25	6476.07	99.05194653	5118.85	5114.17	99.90857321
(Feb-2014)–(Dec-14)	8626.95	8713.91	98.99199601	5933.3	5922.29	99.81443716
(Jan-2015)–(Oct-2015)	9119.2	9130.92	99.87147995	7539.5	7524.16	99.79653823
(Nov-2015)–(Sep-2016)	8968.7	9059.88	98.98335322	6825.8	6828.34	99.96278824
(Oct-2016)–(Apr-2017)	9367.15	9623.66	97.26160038	7893.8	7737.58	98.02097849
(May-2017)–(Feb-2018)	11,171.55	11,180.32	99.92149702	9269.9	9243.23	99.71229463
(Mar-2018)–(Dec-2018)	11,760.2	11,758.02	99.98146290	9951.9	9904.56	99.52431194

The same model has been implemented for predicting the stock market values during a situation of extreme market fluctuation, that is, during the peak phase of the COVID-19 pandemic in March and April 2020. During that time, the stock market was extraordinarily motile, and severe fluctuation was observed. The study shown below provides an insight into how a situation of this sort can be predicted.

6.4 | Case study

To test the effectiveness and accuracy of the model in unexpected situations, it was made to predict every day 'Open,' 'Close,' 'High,' 'Low' features of NIFTY during the Covid-19 pandemic in March and April 2020. NIFTY plunged drastically during this period, and thus it would be a difficult test for the model. The model performed pretty well in this case study. Although the model's average prediction accuracy decreased to around 91%–94% during the first few days of the downfall, it soon adapted to the situation. The model adjusted quickly, and the average prediction accuracy increased to a range of 95%-98%. Therefore, even though the model suffered from an unexpected scenario, it did well to cope with it efficiently. The graphs showing the prediction value and actual value of NIFTY from 15 January 2020 to 5 August 2020 for the features 'Open', 'Close', 'High', 'Low' are given in Figure 14a–d, respectively.

The changes in the prediction accuracy of the four features and the average prediction accuracy over the period are given in Figure 15 below.

The graph shown in Figure 15 depicts the movement of prediction accuracy for the prediction that took place in the time frame of 15th January to 5th August. The sudden downfall in the accuracy occurred because of the heavy market fluctuations during late March. However, the model adjusted with the fluctuations and provided mostly accurate predictions, resulting in an aggregate of 91% accuracy.

6.5 | Comparison table

From the comparison Table 3 shown above, a clear picture is received on the level of accuracy obtained from the model used in this study compared to the studies performed earlier on similar topics. The boosted tree and the ELSTM model provide valuable insight on the predictions, but the accuracy is considerably less compared to that of the proposed model in this study. The model used in this study obtained a significantly higher level of accuracy. The Convolutional Neural Network model proved to give a significantly good result even during extreme market fluctuations like in the case of the COVID-19 pandemic.

7 | DISCUSSION

Artificial Neural Networks which have been used before for stock market prediction gives about 97.66% accuracy, whereas the Convolutional Neural Network model gives 98.92% accuracy on the dataset. The working of CNN model is clearly better than that of the ANN model. This is because the ANN model loses context from the time series data before the CNN model. The dense connections with regularisation are not

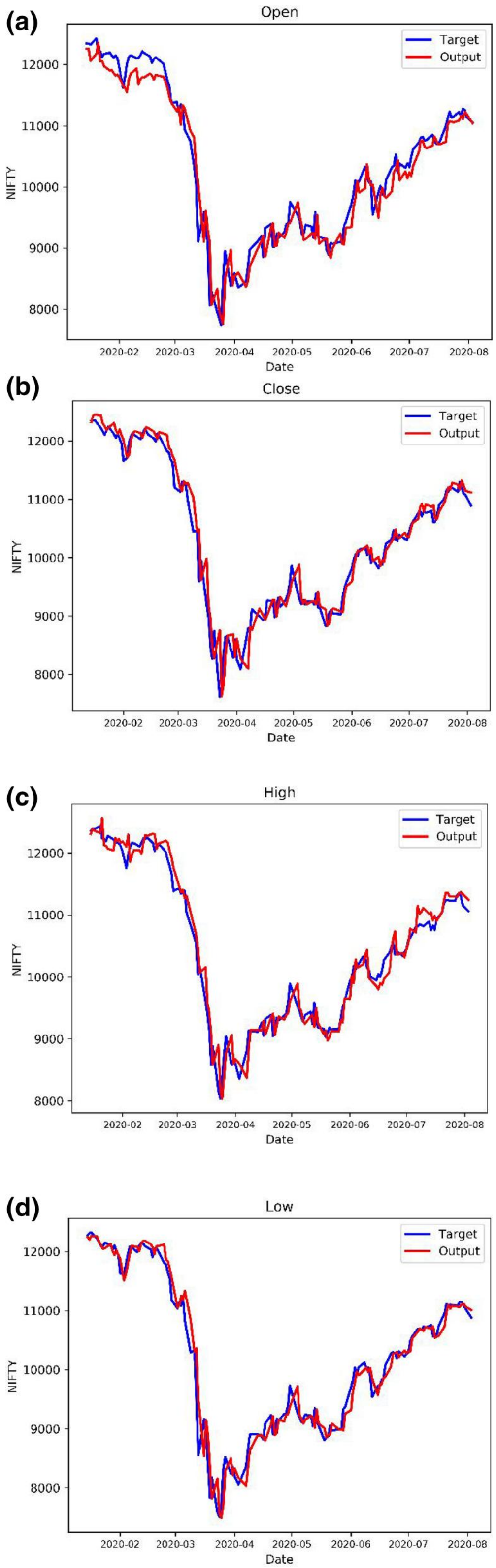

FIGURE 14 (a–d). The predicted value versus actual value of NIFTY dataset from 15 January 2020 to 5 August 2020 for all the features

capable of retaining the time series context. This leads to the CNN model performing better than the ANN model, as it has more connection to get context from the dataset.

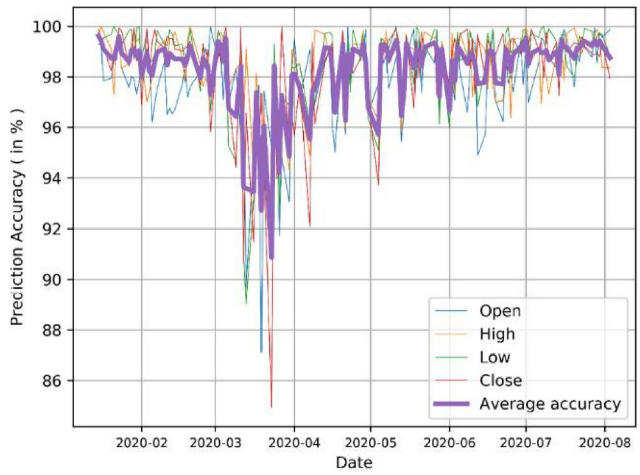

FIGURE 15 Accuracy in predicting the features

TABLE 3 Comparison of accuracy of our proposed model with two recent works

Works	Accuracy in (%)
Aparna Nayak et al. Work done: Boosted decision tree	76.90
Xiongwen Pang et al. Work done: Neural network with embedded layer (ELSTM)	57.20
Proposed work:	
(I) Artificial neural network (with 2 hidden layers)	97.66 (ANN)
(II) CNN (with 2-D histograms generated from stock market data)	98.92 (CNN)
(III) Case study on severe market fluctuations due to COVID-19	91 (under COVID situation)

8 | CONCLUSION

This study proposes two approaches to predict stock market indices and stock prices. This study first uses a Feed-forward Neural Network and performs the backpropagation algorithm for the training process. This model gave a fundamental insight into the prediction trend and provided a graphical result on how the prediction should look. This model gave satisfactory results with an average prediction accuracy of 97.66%, but it required many training data and epochs to get to the above accuracy. Moreover, it also suffered from overfitting to some extent, which was handled using the regularisation method. Thus, to overcome the cost of extensive training, this study further gives another cleaner approach. The Convolution Neural Network model proved to give better results on a given dataset. The study finds a different approach to analyse time-series data. It utilises greyscale 2-D histograms generated from time-series data for prediction. The entire dataset is broken down into 15 segments, every segment is fed into the CNN model, and maximum accuracy is obtained for each of them. This increase in input to the model makes a vast difference in the training time and prediction accuracy. This model required way less training data and time than the previous model. After the training phase and quantisation, the average prediction accuracy of 98.92% is obtained, which is also better than that of the previous model.

On the other hand, the generation of synthetic images in 2-D histograms adds an overhead that is not present in the first approach. Both the methods have their advantages and disadvantages, but both can provide nearly accurate predictions for a dataset of this sort. Such an efficient prediction of the stock market is essential in today's world. It would be of great use to stock market analysts who would be able to develop solutions to help companies and economies by predicting future patterns in the stock market.

ACKNOWLEDGEMENTS

None.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Somenath Mukherjee https://orcid.org/0000-0003-1729-0502

Bikash Sadhukhan https://orcid.org/0000-0001-5469-0711

REFERENCES

1. Statman, M., Fisher, K.L.: Consumer Confidence and Stock returns. SSRN Journal. https://doi.org/10.2139/ssrn.317304
2. Wilcox, J.: Consumer Sentiment and Consumer Spending. FRBSF Economic Letter (2008)
3. Kaul, V. https://www.deccanherald.com/specials/sunday-spotlight/what-s-in-store-for-the-indian-economy-in-2020-789545.html
4. Sharma, A., Bhuriya, D., Singh, U.: Survey of stock market prediction using machine learning approach. In: International conference of Electronics, Communication and Aerospace Technology (ICECA). (2017). https://doi.org/10.1109/ICECA.2017.8212715
5. Nivetha, R.Y., Dhaya, C.: Developing a prediction model for stock analysis. In: 2017 International Conference on Technical Advancements in Computers and Communications (ICTACC), IEEE (2017). https://doi.org/10.1109/ICTACC.2017.11
6. Jayanth Balaji, A., Harish Ram, D.S., Nair, B.B.: Applicability of deep learning models for stock price forecasting an empirical study on BANKEX data. Procedia Comput. Sci. 143, 947–953 (2018). https://doi.org/10.1016/j.procs.2018.10.340
7. Lin, T.-W., Yu, C.-C.: Forecasting stock market with neural networks. SSRN Journal. https://doi.org/10.2139/ssrn.1327544
8. Shah, V.S., et al.: Stock market prediction using neural networks. Int. J. Soft Comput. Eng. 6(1), (2016)
9. Kalyvas, E.: Using Neural Networks and Genetic Algorithms to Predict Stock Market Returns. University of Manchester (2001)
10. Girija, V.A., et al.: Stock Market Prediction: A Big Data Approach. IEEE (2015). https://doi.org/10.1109/TENCON.2015.7373006
11. Ertuna, L.: Stock Market Prediction Using Neural Network and Time Series Forecasting. IEEE (2016). https://doi.org/10.13140/RG.2.1.1954.1368
12. Wen, M., et al.: Stock market trend prediction using high-order information of time series. IEEE Access. 7, 28299–28308 (2019). https://doi.org/10.1109/access.2019.2901842
13. Usmaini, M., et al.: Stock Market Prediction Using Machine Learning Techniques. ICCOINS (2016). https://doi.org/10.1109/ICCOINS.2016.7783235
14. Das, N., Ghosh, P., Roy, D.: Effect of Demonetization on stock market correlated with geo twitter sentiment analysis. In: Learning and Analytics in Intelligent Systems. Springer, Siliguri (2020). https://doi.org/10.1007/978-3-030-42363-6_92
15. Li, W., et al.: User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models. Appl. Soft Comput. 94, 106435 (2020). https://doi.org/10.1016/j.asoc.2020.106435
16. Nayak, A., Pai, M.M.M., Pai, R.M.: Prediction models for Indian stock market, In Procedia Comput. Sci. 89, 441–449 (2016). https://doi.org/10.1016/j.procs.2016.06.096
17. Drucker, H., Cortes, C.: Boosting decision trees. Adv Neural Inf Process Syst. 0, 6 (1995). https://dl.acm.org/doi/10.5555/2998828.2998896
18. Pang, X., et al.: An innovative neural network approach for stock market prediction. J. Supercomput. 76(3), 2098–2118 (2020). https://doi.org/10.1007/s11227-017-2228-y
19. Lien, M.D.: Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. IEEE Access. 6, 55392-55404 (2018). https://doi.org/10.1109/access.2018.2868970
20. Chen, J.-F., et al.: Financial time-series data analysis using deep convolutional neural networks. In International conference of cloud Computing and Big Data (2016). https://doi.org/10.1109/CCBD.2016.027
21. Naeini, M.P., Taremian, H.: Stock Market Value Prediction Using Neural Networks. IEEE (2010). https://doi.org/10.1109/CISIM.2010.5643675
22. Chen, S., He, H.: Stock prediction using convolutional neural network. In IOP conference Series: Materials Science and Engineering (2018)
23. Du, X., et al.: Overview of deep learning. In Youth Academic Annual conference of Chinese Association of Automation, Wuhan (2016). https://doi.org/10.1109/YAC.2016.7804882
24. N.S Exchange: NSE India, [Online]. https://www.nseindia.com/
25. Hiransha, M., et al.: NSE stock market prediction using deep-learning models. Procedia Comput. Sci. 132, 1351–1362 (2018). https://doi.org/10.1016/j.procs.2018.05.050
26. Ding, G.J., et al.: Hierarchical multi-scale Gaussian transformer for stock movement prediction. International Joint Conference on Artificial Intelligence Organization (2020). https://doi.org/10.24963/ijcai.2020/640
27. Li, S.Q., et al.: Modeling the stock relation with graph network for overnight stock movement prediction. International Joint Conference on Artificial Intelligence Organization (2020). https://doi.org/10.24963/ijcai.2020/626
28. Lu, W., et al.: A CNN-BiLSTM-AM method for stock price prediction. Neural. Comput. Appl. 33(10), 4741–4753 (2021). https://doi.org/10.1007/s00521-020-05532-z
29. Liu, W.X., et al.: Multi-scale two way deep neural network for stock trend prediction. International Joint Conference on Artificial Intelligence Organization (2020). https://doi.org/10.24963/ijcai.2020/628
30. Nayak, A., Pai, M.M.M., Pai, R.M.: Prediction models for Indian stock Market. Procedia Comput. Sci. 89, 441–449 (2016). https://doi.org/10.1016/j.procs.2016.06.096
31. Pang, X., et al.: An innovative neural network approach for stock market prediction. J. Supercomput. 22 (2018)
32. B.S Exchange: NSE India, national stock exchange. [Online]. https://www1.nseindia.com/
33. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best Practices for Convolutional Neural Networks.MicrosoftRedmond, WA. (2003). https://doi.org/10.1109/ICDAR.2003.1227801
34. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional. Adv. Neural Inf. Process. Syst. 25(NIPS), 9 (2012). https://dl.acm.org/doi/10.5555/2999134.2999257
35. 2D Histograms in Python, Plotly, [Online]. Available: https://plotly.com/python/2D-Histogram/

How to cite this article: Mukherjee, S., et al.: Stock market prediction using deep learning algorithms. CAAI Trans. Intell. Technol. 8(1), 82–94 (2023). https://doi.org/10.1049/cit2.12059