ORIGINAL ARTICLE

Stock Market Prediction Using Optimized Deep-ConvLSTM Model

Amit Kelotra* and Prateek Pandey

Abstract

Stock market prediction acts as a challenging area for the investors for obtaining the profits in the financial markets. A greater number of models used in stock market forecasting is not capable of providing an accurate prediction. This article proposes a stock market prediction system that effectively predicts the state of the stock market. The deep convolutional long short-term memory (Deep-ConvLSTM) model acts as the prediction module, which is trained by using the proposed Rider-based monarch butterfly optimization (Rider-MBO) algorithm. The proposed Rider-MBO algorithm is the integration of rider optimization algorithm (ROA) and MBO. Initially, the data from the live stock market are subjected to the computation of the technical indicators, representing the features from which the necessary features are obtained through clustering by using the Sparse-Fuzzy C-Means (Sparse-FCM) followed with feature selection. The robust features are given to the Deep-ConvLSTM model to perform an accurate prediction. The evaluation is based on the evaluation metrics, such as mean squared error (MSE) and root mean squared error (RMSE), by using six forms of live stock market data. The proposed stock market prediction model acquired a minimal MSE and RMSE of 7.2487 and 2.6923 that shows the effectiveness of the proposed method in stock market prediction.

Keywords: rider optimization algorithm; monarch butterfly optimization; stock; forecast; deep learning

Introduction

The extraction of information from the data is performed with the concept of predictive analysis, which is an area in data mining that can be used for the prediction of the behavior trend patterns.¹ The predictive analytics is defined as the prediction of each individual, organizational element with a more detailed level of granularity.² The working of the predictive analytics is based on observing the relationship among the predicted elements and the explanatory elements of the occurrences that happened in the past and then uses it for the prediction of the unknown outcome. The present and the past facts are analyzed for predicting the future, and the models are developed for the prediction of future behaviors and the events. Most of the models of prediction are capable of producing improved results that represent improved likelihood about the given event. The project risk management,

fraud detection, marketing, customer retention, clinical decision support systems, and analytical customer relationship management are some of the applications of the predictive analysis.³ Machine learning is a type of artificial intelligence that contains various advanced statistical methods of classification and regression that are useful in the predictive analysis.⁴ Neural networks (NNs) are used in predicting, forecasting, and analysis, and the deep NN is a variant of the multilayer perceptrons used by various researchers in predictive analytics.⁵ Predictive analytics acts as a highly demanding area in the field of business intelligence, and it is important in the basic growth of the organization.⁶

The stock market forecasting is very tedious as compared with the other areas due to the changing behavior of the stock market.⁷ The prediction of the stock market by using intelligent systems was developed to make the decision about the trade. In recent years, various

Jaypee University of Engineering & Technology, Raghogarh, Guna, Madhya Pradesh, India.

*Address correspondence to: Amit Kelotra, Jaypee University of Engineering & Technology, Raghogarh, Guna, Madhya Pradesh-473226, India, E-mail: kelotra.amit@gmail.com

hybrid intelligent systems were developed to perform complicated automation works, modeling expertise, and decision support aided with the common analytical methods to perform the analysis of the stock market.⁸ The current stock data prediction acts as a major need for the research due to the presence of a number of optimization tools for the prediction of the future market. The artificial NN (ANN)⁹ is considered as an efficient optimization tool for predicting the time series, and also, it predicts the hidden and the unknown records.¹⁰ The advancements in the deep learning algorithm have produced various trading algorithms to predict the stock price changes with more accuracy. The deep learning method¹¹ reshapes the basics of the predictive analysis and has achieved various advantages in the fields of machine translation,¹² question answering,¹³ speech recognition,¹⁴ and in various other areas.¹⁵ In recent years, the support vector machine (SVM),¹⁶ which is a supervised learning method, has achieved popularity. This method makes use of its ability for learning the rule changes presently in the time series and also, it is used in the prediction of the future.¹⁷

The statistical-time series models, such as the Kalman filter, state space, ARIMA, exponential smoothing, and the regression, are used in the analysis of the stock market. These models make use of a single point and find it difficult to fit the data in nonlinear nature. Also, the stock market is a complex system with noisy characteristics.¹⁸ For solving these problems, the genetic algorithm, SVM, and the NN based on statistics are recommended. These methods provide better values of the forecast but are not suitable for the prediction of the fuzzy data.¹⁹ The fuzzy models are capable of tolerating the problem of approximation, uncertainty, and imprecision.²⁰ The excess of information produced by using the stock market is used to improve the knowledge of the investors.²¹ The prediction of the characteristics of the prices of the stock market acts as an important area for most of the financial scientists and investors. The investment in the stock market paves a way to obtain profit that requires the correct data from the stock market, trend forecasting, and stock changes. Thus, it is necessary to make the use of reliable and powerful tools to predict stock prices.¹⁷ The efficient prediction algorithm provides a considerable benefit to traders. The accuracy of prediction models depends on the features selected, which are the input to the prediction model. Though different methods are developed for the prediction of the stock market, the feature selection from the financial news is

complex. Deep learning methods can extract features from a large set of raw data, which makes it potentially attractive for stock market prediction. By considering the drawbacks in the existing stock market prediction techniques, this article introduces the stock market prediction technique to effectively predict the stock price.

The main aim of this article is to design and develop a system named Rider-monarch butterfly optimization (MBO)-based deep-convolutional long short-term memory (ConvLSTM) model, for predicting the state of the stock market. The inputs to the proposed system are the current and past status of the stock market. Initially, the input data obtained from the live stock market are fed to the feature extraction process to extract the required features based on the technical indicators, namely as double exponential moving average (DEMA), rate of change (ROCR), average true range (ATR), relative strength index (RSI), simple moving average (SMA), average directional index (ADX), commodity channel index (CCI), moving average convergence divergence (MACD), on balance volume (OBV), Parabolic SAR (SAR), exponential moving average (EMA), and triple exponential moving average (T3). The technical indicators are responsible for the provision of the data needed to predict the stock market effectively. The data features obtained from the technical indicators are clustered by using the Sparse-Fuzzy C-Means (FCM) algorithm. Then, the highly significant features are given to the proposed Rider-MBO-based Deep-ConvLSTM model, which acts as the prediction module. Here, the Deep-ConvLSTM model is trained by using the proposed Rider-MBO algorithm for predicting the stock market.

The major contributions of the article

Rider-MBO-based Deep-ConvLSTM model for predicting the stock market. The Rider-MBO-based Deep-ConvLSTM model is newly developed for predicting the future of the stock market.

Rider-MBO algorithm. The Rider-MBO algorithm is developed by integrating the rider optimization algorithm (ROA) and MBO to train the Deep-ConvLSTM model.

The organization of the article is as follows: Introduction section gives the introduction to stock market prediction. Motivation section discusses the literature survey of various existing methods of stock market prediction. Proposed Methodology: Stock Market Prediction Using Rider-MBO-based Deep-ConvLSTM

Model section deals with the proposed Rider-MBO-based Deep-ConvLSTM model for stock market prediction. Results and Discussions sections appear next. Finally, Conclusion section concludes the article.

Motivation

In this section, the survey of various techniques used in stock market prediction and the challenges of the existing methods are discussed.

Literature review

The review of the eight existing methods is discussed: Weng et al.²² used the decision trees, NNs, and SVMs for a financial expert system that was capable of improving the prediction accuracy. In addition, effective decisions on the commodity, index, or stock were made by using the intelligent decision-making tool. The drawback of this method was the exclusion of the other online data sources, and the applicability of the sources was not clear. Efendi et al.²⁰ designed the Improved Fuzzy Random Auto-Regression (FR-AR) model, as the forecasting accuracy of the FR-AR model was enhanced by using this method and was capable of solving the biasness among the machine and human error, but it cannot be applied to handle the nonstationary time series data of the various area. Chen et al.²³ modeled the improved fundamental analysis-based approach, which was useful for the investors to choose the optimal stocks and predict the future price trend. In addition, this system improved forecasting accuracy and helped in the enhancement of decision quality. The drawback of this system was the inability to supervise the learning process, and the failure in considering the sentence meaning of the headlines of financial news. Pang et al.²⁴ developed the Deep LSTM neural network, which offered improved performance in prediction for the Shanghai A-share composite index, but the expansion of the research-related datasets for checking the applicability of the models in another stock market with increased accuracy was not possible in this method.

Chandar²⁵ modeled the Subtractive clustering-based adaptive neuro-fuzzy approach, as this method was suitable with respect to the number of parameters, number of rules, testing error, and training error, but the problems cannot be solved individually, and as a result there was a possibility for the development of imperfect, imprecise, and vague data. Singh and Srivastava²⁶ designed the two-directional two-dimensional principal component analysis (2D)2PCA+Deep NN method,

which enhanced the stock multimedia prediction accuracy, but the deep learning approach in case of large window size and high dimension offered only low performance. Pathak and Shetty²⁷ developed the combination of sentiment analysis and NN techniques for the prediction model that predicted the beneficial financial returns and the market trend with high accuracy but did not consider the refined fuzzy rules for the improvement in prediction. Zhang et al.²⁸ designed the Coupled matrix and tensor factorization method that was capable of predicting the changes in the stock market with the completion of the missing values of the sparse tensor, but it was not capable of considering the other data sources, such as knowledge graph information.

Challenges

The major challenges associated with the existing methods of stock market prediction are as follows:

- The Coupled matrix and tensor factorization method²⁸ makes use of the tensor to predict the stock market. The decomposition of the tensor in the presence of event sparsity and of obtaining improved accuracy acts as a major challenge with regard to this method.
- The stock market is considered dynamic, volatile, complex, and chaotic, and the prediction of the stock market is one of the tedious tasks in terms of time series forecasting. The ANN methods were not capable of performing this task.²⁶
- One of the challenging fields for investors is the stock market prediction for obtaining the profits of the financial markets. It is necessary for investors to know about the financial markets that violate the external factors.²⁵
- The prediction of the stock market with more accuracy is a challenging task due to the presence of nonstatic stock prices and noisy nature. The macro-economic factors, such as the company's policy, political events, product price indexes, general economic conditions, the expectation of investors, and the psychological factors, affect the stock prices.¹⁹
- The commercial financial expert systems²² were designed for the trading on the stock exchange with the limitation in the prediction accuracy due to the dependence of them in the time-series analysis of the market.

By considering the challenges of the existing stock market prediction methods as the motivation, this article

develops a stock market prediction system, which tries to overcome the drawbacks of the existing methods.

Proposed Methodology: Stock Market Prediction Using Rider-MBO-based Deep-ConvLSTM Model

The primary intention of this article is to design and develop a stock market prediction system to ensure the effective prediction of the stock market. An approach is developed to predict the future market based on the past and the current status of the market. Initially, the input data are subjected to the extraction of the features based on the technical indicators, such as DEMA, ROCR, ATR, RSI, SMA, ADX, CCI, MACD, OBV, SAR, EMA, and T3. The technical indicators ensure the information of the data that significantly contributes toward predicting the stock market. The Sparse-FCM algorithm is employed for clustering the data features that are generated by using the technical indicators. The highly significant features of the data are fed to the prediction module, for which the Deep-ConvLSTM model is employed. The Deep-ConvLSTM model uses the

Rider-MBO algorithm that is developed by the integration of the ROA²⁹ and the MBO.³⁰ Thus, the proposed Rider-MBO-based Deep-ConvLSTM model performs the stock market prediction with the considered technical indicators. Figure 1 depicts the block diagram of the proposed stock market prediction model.

Data model

The input data consist of both the open and close prices of the stock market for all the days in a year. In addition to these data, the high value, low value, and the change in market cap of the stock market are available in the data model. The data model is of the size $b_{j,i}$, $i \leq n$ where the j th row contains the data attributes such as open price, close price, and high value; the i th column contains the days in a year; and n is the total number of days.

Technical indicators

The stock market prediction is carried out through the use of the technical indicators that are focused on the historical trading information, and these indicators

are simply the mathematical model depending on the open and closing prices. Through the analysis of the historical data, the future of the stock market is predicted; moreover, these technical indicators assist the long-term investors in determining the entry and exit prices. The 12 technical indicators,³¹ such as DEMA, ROCR, momentum (MOM), RSI, SMA, ADX, triple exponential moving average oscillator (TRIX), MACD, OBV, time series forecast (TSF), triple exponential moving average (TEMA), and T3 are utilized in this article; these are considered the features for initiating the stock market prediction and they are grouped on the basis of their similarity.

- (i) DEMA: The DEMA is a rate of trending average price of a security that offers the most weight to the recent price data. Similar to EMA, the DEMA is more reactive to the fluctuations in price compared with that of the SMA, and it thus helps the short-term traders, who aim at attempting to pinpoint trend changes.

$$D = (2 * E(n)) - (E(E(n))) \quad (1)$$

where D indicates the DEMA, E is the EMA, and n is the period.

- (ii) ROCR: The ROCR indicator is otherwise termed as momentum indicator; it is used in the comparison of the current and the previous price for a certain number of periods, and the current price is split with the previous price.

$$ROCR = \frac{P(t)}{P(t-n)} \times 100 \quad (2)$$

where $P(t)$ denotes the current price, and $P(t-n)$ denotes the price of a certain number of periods a while ago.

- (iii) ATR: The ATR is used to estimate the size of the period range, and it considers the gap from the close of the previous period.

$$T = A(T_{\text{ran}}, n) \quad (3)$$

where A is the ATR, and T_{ran} is the true range.

- (iv) RSI: The RSI is one of the momentum oscillators that measures how often the stock was bought or sold, and it assumes that stock brought or sold eventually balances out. The relationship for the RSI is expressed as,

$$RSI = A(PU) / (A(PU) + A(PD)) * 100 \quad (4)$$

where,

$$\begin{cases} PU(t) = 1 * (P(t) - P(t-1)), & \text{for } P(t) - P(t-1) > 0 \\ PD(t) = 1 * (P(t-1) - P(t)), & \text{for } P(t) - P(t-1) < 0 \end{cases} \quad (5)$$

where $A(PU)$ denotes the average value of price up, and $A(PD)$ is the average value of price down.

- (v) SMA: The SMA is the average calculation of the closing price of any security for a selected number of days. The short-term SMA uses only a small amount of days as input; on the other hand, the long-term SMA uses a larger sample of days.

$$M = (\text{Sum}(P, n)) / n \quad (6)$$

where P is the price, and n is the period.

- (vi) ADX: The ADX is used to measure the trend strength with the rate of price movement in a single direction. Also, the ADX is responsible for the discovery of the developing trends.

$$ADX = \text{sum} (+I - (-I)) / (+I + (-I)) / n \quad (7)$$

where I is the directional index.

- (vii) CCI: The CCI compares the current and the average mean price for about 20 periods.

$$Q = \frac{(S - A)}{0.015 * \bar{h}} \quad (8)$$

where $S = \frac{\text{highest price} + \text{lowest price} + \text{closing price}}{3}$, A is the average of S , and $\bar{h}$ is the mean deviation of the term $S - A$.

- (viii) MACD: The MACD is the commonly used indicator for revealing the changes in direction, strength, and duration of the stock's price. In general, it is the difference between fast and slow moving average.

$$MACD = F_{MA} - S_{MA} \quad (9)$$

where F_{MA} is the shorter moving average, and S_{MA} is the longer moving average.

- (ix) OBV: The OBV relates trading volume to the price change, and it is expressed as,

$$B(t) = B(t-1) \pm V(t) \quad (10)$$

where $B(t)$ is the OBV, $B(t-1)$ is the OBV of the previous period, and $V(t)$ is the volume.

- (x) SAR: The SAR is used in the trailing stop points to make it usable with the short and long positions. In the case of new long positions, the SAR is given as,

$$\text{SAR} = \text{PPS} + \text{AF}(\text{Highest price} - \text{PPS}) \quad (11)$$

In the case of new short positions, the P_s is given as,

$$\text{SAR} = \text{PPS} + \text{AF}(\text{Lowest price} - \text{PPS}) \quad (12)$$

where PPS is the previous period SAR, and AF is the acceleration factor.

- (xi) EMA: The EMA indicates the average prices, and it places more weight on the recent prices. The weighting given to the most recent price always relies on a certain period of moving average.

$$E = (\text{Current price} - \text{EM}) * \text{SC} + \text{EM} \quad (13)$$

where EM indicates the EMA for a certain period, and SC is the smoothing constant, which is equal to $\frac{2}{n+1}$.

- (xii) T3: The T3 is the one that tries to provide moving average with enhanced smoothing compared with that of the conventional EMA.

$$T3 = 3 * E1 - 3 * E2 + E3 \quad (14)$$

$$E1 = E(t, z) \quad (15)$$

$$E2 = E(E1, z) \quad (16)$$

$$E3 = E(E2, z) \quad (17)$$

where z is the period, and E is the EMA.

The dimension of each technical indicator is $[1 \times 1]$, and the extracted feature is of the size $[1 \times 12]$. The extracted features are fed to the process of clustering to group the features based on their similarities.

Clustering of features using Sparse-FCM algorithm

The process of clustering is performed to cluster the features that are obtained as the output of the technical indicators. The highly significant features are collected by using the clustering process, which is done by using the Sparse-FCM algorithm. The features with similar features are grouped into a single cluster, and thus a

number of cluster groups are framed. The Sparse-FCM algorithm³² is used for clustering the features that are obtained from the technical indicators. The features are grouped as clusters based on the similarities in the characteristics of the features. It is obvious that the prediction accuracy depends on the proper clustering for which the Sparse-FCM is used. The input of the Sparse-FCM is a feature H , which is denoted as $q = [H_{bc}] \in b^{i \times j}$, where $[i \times j]$ is denoted such that i indicates the row and j indicates the column of H . The algorithmic steps of the Sparse-FCM Clustering in clustering the features are detailed as follows:

Step 1: Initialization: In the initial step, the weights of the features are initialized as, $w = w_1^d = w_2^d = \dots = w_j^d = \frac{1}{\sqrt{j}}$.

Step 2: Partition matrix update: For each cluster center a and attribute weights w , the $\gamma(P)$ is minimized if

$$p_{uv} = \begin{cases} \frac{1}{N_v} & ; \text{ if } U_{uv}=0 \text{ and } N_i = \text{card} \{l : U_{ul}=0\} \\ 0 & ; \text{ if } U_{uv} \neq 0 \text{ but } U_{uv}=0 \text{ for some } t, t \neq i \\ \frac{1}{\sum_{t=1}^i \left(\frac{U_{uv}}{U_{tv}}\right)^{\frac{1}{\epsilon-1}}} & ; \text{ Otherwise} \end{cases} \quad (18)$$

where $\text{card}(r)$ denotes the cardinality of the set r . The distance measure of the standard Sparse-FCM is expressed as,

$$U_{uv} = \sum_{v=1}^i w_l (a_{ul} - a_{vl})^2 \quad (19)$$

Step 3: Update of the cluster center a : Let w and p be fixed, and $\gamma(a)$ is minimized if

$$a_{vl} = \begin{cases} 0 & ; \text{ if } w_l = 0 \\ \frac{\sum_{u=1}^s p_{uv}^e \cdot a_{ul}}{\sum_{u=1}^s p_{uv}^e} & ; \text{ if } w_l \neq 0 \end{cases} \quad (20)$$

where $v = 1, \dots, i$ and $l = 1, \dots, j$; ϵ refers to the weight component that is used for controlling the degree of membership, which is shared among the fuzzy clusters. The contribution of l th feature to the objective function is represented as w_l .

Step 4: Class determination: The class value is estimated with the use of the fixed clusters $c = \{c_1, c_2, \dots, c_c, \dots, c_i\}$ and the membership function σ . The class u_c is calculated based on the following objective that is given next as, $\max_w \sum_{l=1}^f w_l \cdot h_l$, where

$\|w\|_2^2 \leq 1$, $\|w\|_r \leq \tau$ to obtain the value of w^* , where τ indicates the tuning parameter and $(0 \leq r \leq 1)$;

$$\|w\|_r^r = \sum_{c=1}^f |w_c|^r.$$

Step 5: Condition for termination: The steps just provided are repeated until the stopping condition is satisfied. The stopping condition is expressed as,

$$\frac{\sum_{l=1}^f |w_l^* - w_l^d|}{\sum_{l=1}^f |w_l^d|} < 10^{-4} \quad (21)$$

The output of the Sparse-FCM comprises the clusters containing similar features, the highly essential features are chosen based on the minimal distance between the cluster, and the feature is selected to predict the stock market. The clusters obtained by using the Sparse-FCM are given as,

$$f = \{f_1, f_2, \dots, f_{\hat{o}}, \dots, f_{\ell}\} \quad (22)$$

where $f_{\hat{o}}$ is the $\hat{o}^{th}$ cluster consisting of v number of features. The v number of features in the cluster $f_{\hat{o}}$ needs to be selected, which is done based on the minimal distance match.

Feature selection using minimal distance match

The main aim of feature selection is to reduce the dimension of the feature vector to minimize the computational complexity associated with the prediction. The cluster groups with significant features are selected to perform the prediction process, from which the effective

features are selected based on the minimum distance between the cluster and the individual feature in the cluster. The figure depicts the selection of the feature from the centroid based on minimum distance. Figure 2 depicts the selection of the feature based on minimum distance, in which a centroid with three features is considered. The distance between the centroid and each feature is calculated, and the feature possessing minimal distance is selected. The highly significant features of the data are given to the prediction module, for which the Deep-ConvLSTM model is used.

The selection of features from each cluster is done based on minimal distance. t_c number of features are selected in such a way that $t_c < r_d$ indicates that the feature dimension is reduced, which, in turn, reduces the computational complexity.

Stock market prediction using deep learning based on the optimization algorithm

The stock market prediction is carried out by using the Deep-ConvLSTM classifier, which obtains the effective features as the input. The Deep-ConvLSTM classifier is trained by using the proposed classifier, namely the Rider-MBO algorithm, which is developed by the integration of MBO and ROA.

Architecture of Deep-ConvLSTM. The ConvLSTM has various advantages as compared with those of the Fully Connected LSTM (FC-LSTM) that require unfolding the input data as the 1D vector before processing, which leads to the loss in spatial information. The ConvLSTM is grouped as 3D tensors of inputs $Y_1, \dots, Y_z$, cell outputs $Z_1, \dots, Z_z$, hidden

FIG. 2. Feature selection based on minimum distance.

states $C_1, \dots, C_\alpha$, gates J_α , K_α , and L_α of ConvLSTM. The spatial data that remain in the ConvLSTM act as the most important advantage as compared with the FC-LSTM. Thus, the ConvLSTM³³ is capable of predicting the future with the use of the inputs and the past states with respect to its neighbors. This process is carried out by using the Hadamard product and the Convolutional operator*. The Hadamard product is used to protect the constant error carousel nature of the cells. The ConvLSTM provide effective feature patterns with the employment of the large transitional kernel. To perform the effective process of prediction, the ConvLSTM structure is grouped as forecasting layers and encoding layers. The encoding network's initial states and the cell outputs of the final stage are copied to the forecasting network. The prediction output is of the same dimension, similar to the input, and thus, all the states of the forecast network are joined together and given to the $[1 \times 1]$ conv layer for performing the final prediction process. Figure 3 shows the architecture of ConvLSTM networks that differentiates between the forecast and the encoding layers.

The encoding LSTM compresses the input inside the hidden states to forecast the final stock market prediction. The LSTM network comprises memory units that consist of gates and cell, which are depicted in Figure 3. In addition, the calculation of the output from each memory unit component is demonstrated.

The memory cell and the gates control the flow of information, and also, the encoding-forecasting structure of ConvLSTM guarantees the forecast depending on the spatiotemporal sequences. The ordinal neuron is replaced by the three gates and the memory cell in case of the other deep learning methods. Thus, depending on the new input, without considering the unwanted contents, the memory cell state of the ConvLSTM is updated, and finally, the output is provided. These features of the ConvLSTM make the possibility of the prediction in case of time-series data. The estimation of the input gates and the output of the memory unit are detailed.

The change in the functioning of the ConvLSTM⁸ from that of the other deep learning networks depends on the way of using the feedback loops for holding the past memory. The output of the input gate is obtained as,

$$J_\alpha = \beta (\delta_f^Y * Y_\alpha + \delta_f^C * C_{\alpha-1} + \delta_f^Z \circ Z_{\alpha-1} + \eta^I) \quad (23)$$

where Y_α indicates the input vector, δ_f^Y indicates the weight between the input layer and the input gate, β is the gate activation function that is mostly a sigmoid function, δ_f^C indicates the weight between the input layer and the memory output, and δ_f^Z is the weight vector between the input layer and the cell output. $C_{\alpha-1}$ and $Z_{\alpha-1}$ are the previous output of the cell and

FIG. 3. Architecture of ConvLSTM.

memory unit, respectively, and η^J is the bias of the input layer. $*$ is the convolutional operator, and $\circ$ is the element-wise multiplication. The result from the forget gate is derived as,

$$K_\alpha = \beta (\delta_K^Y * Y_\alpha + \delta_K^C * C_{\alpha-1} + \delta_K^Z \circ Z_{\alpha-1} + \eta^K) \quad (24)$$

where δ_K^Y indicates the weight between the input layer and forget gate, δ_K^C indicates the weight that is responsible for the connection between the output gate and the memory unit of the previous layer, δ_K^Z is the weight between the output gate and the cell, and η^K is the bias with respect to the forget gate. The output of the output gate is given as,

$$L_\alpha = \beta (\delta_L^Y * Y_\alpha + \delta_L^C * C_{\alpha-1} + \delta_L^Z \circ Z_\alpha + \eta^L) \quad (25)$$

where δ_L^Y is the weight that is responsible for the link between the output gate and the input layer, δ_L^C is the weight between the output gate and the memory unit, and δ_L^Z is the weight between the output gate and the cell. The output gate is indicated as η^L . The output of the temporary cell state is calculated depending on the activation function of the weights with respect to the cell and is given as,

$$\tilde{Z}_\alpha = \tanh (\delta_F^Y * Y_\alpha + \delta_F^C * C_{\alpha-1} + \eta^F) \quad (26)$$

where δ_F^Y is the weight between the cell and the input layer, and δ_F^C is the weight between the cell and memory unit. The bias of the cell is represented as η^F .

The output of the cell is termed as the sum of the temporary cell state and the difference between the memory unit of the previous and the current layers that is expressed as,

$$Z_\alpha = K_\alpha \circ Z_{\alpha-1} + J_\alpha \circ \tilde{Z}_\alpha \quad (27)$$

$$Z_\alpha = K_\alpha \circ Z_{\alpha-1} + J_\alpha \circ \tanh (\delta_F^Y * Y_\alpha + \delta_F^C * C_{\alpha-1} + \eta^F) \quad (28)$$

The output from the memory unit is given as,

$$C_\alpha = L_\alpha \circ \tanh (Z_\alpha) \quad (29)$$

where C_α is the output of the memory block, and L_α is the output gate. Thus, the output of the output layer is expressed as,

$$G_\alpha = \mu (\delta_G^C \cdot C_\alpha + \eta^G) \quad (30)$$

where Z_τ indicates the output vector, and δ_G^C indicates the weight between the output vector and memory unit. The bias of the output layer is indicated as η^G .

Thus, from Figure 4, it is clear that the output of the output layer is fed to the succeeding LSTM, and the copies of the output from the LSTMs in the encoding layer are given to the forecasting layer to obtain the final prediction about the stock market. Thus, the bias and the weights of the ConvLSTM are represented as $\eta \in \{\eta^F, \eta^K, \eta^L, \eta^J\}$ and $\delta \in \{\delta_G^C, \delta_F^Y, \delta_F^C, \delta_L^Y, \delta_L^C, \delta_L^Z, \delta_K^Y, \delta_K^C, \delta_K^Z, \delta_J^Y, \delta_J^C, \delta_J^Z\}$. The tuning of the weights δ and biases η of the ConvLSTM is determined optimally by using the proposed Rider-MBO algorithm.

Generation of the weights for training the ConvLSTM using the proposed Rider-MBO algorithm. The weights of the ConvLSTM are obtained by using the proposed Rider-MBO algorithm, which is developed by the integration of MBO and ROA. The ROA algorithm follows a new approach, termed as fictional computing, which is developed to solve the optimization problems based on imaginary thoughts and ideas. This algorithm works based on the four groups of riders who try to reach the destination. Some groups of riders are formed in such a way that they travel to the same destination to become the winner. For instance, four groups are considered, and the number of riders in each group is selected from the total number of riders equally. The bypass rider, follower, overtaker, and attacker are the four groups of riders, where each group follows a unique idea to reach the destination. The bypass rider bypasses the leading path for reaching the target and does not take the leading rider in concern, whereas the follower follows the leading rider. The overtaker follows his position to reach the destination with respect to the neighboring location of the leading rider, and the attacker is the one who reaches the destination by the utilization of the maximum speed. Finally, the winner is the rider who wins the race. The algorithmic steps of the ROA are as follows:

- (a) *Group and rider parameter initialization*: The algorithm is initialized by the four rider groups, represented by parameter initialization (PI), whose position is initialized randomly. The initialization of the group is expressed as,

$$l_t = l_t(x, y); \quad 1 \leq x \leq g; \quad 1 \leq y \leq k \quad (31)$$

where $l_t(x, y)$ is the x th rider position at the time t , g is the total number of riders, which is equal to that of PI, and k is the dimension or the number of co-ordinates. The number of riders in each group is used in the calculation of the total number of riders, and it is expressed as,

FIG. 4. Memory unit of the ConvLSTM. MSE, mean squared error; RMSE, root mean squared error.

$$g = B_R + F_L + O_{TK} + A_{TT} \quad (32)$$

where B_R is the number of bypass riders, F_L is the number of followers, O_{TK} is the number of overtakers, and A_{TT} is the number of attackers. The relationship among these parameters is expressed as,

$$B_R = F_L = O_{TK} = A_{TT} = \frac{g}{4} \quad (33)$$

The positions of the riders are calculated by using the relation just given. The initialization of the rider parameters, namely the steering, gear, accelerator, and the gear, is done after the initialization of the group. The steering angle at a time t is expressed as,

$$S_t = \{S_{x,y}^t\}; \quad 1 \leq x \leq g; \quad 1 \leq y \leq k \quad (34)$$

where $S_{x,y}^t$ is the steering angle of the x^{th} rider's vehicle. At zero initial time, the steering angle is the initial steering angle, and it is given as,

$$S_{x,y} = \begin{cases} \phi_x & \text{for } y=1 \\ S_{x,y+\theta} & \text{for } y \neq 1, S_{x,y+\theta} \leq 360 \\ S_{x,y+\theta-360} & \text{else} \end{cases} \quad (35)$$

where ϕ is the position angle, and θ_x is the co-ordinate angle. The estimation of the position angle depends on

the maximum angle of 360° , and the number of riders and the co-ordination angle is used in the calculation of the steering angle.

The gear of the vehicle in a group is expressed as,

$$Ge = \{Ge_x\}, \quad 1 \leq x \leq g \quad (36)$$

where Ge_x is the x th rider's vehicle's gear angle. The initial value of the gear angle is 0, and at the time t , it takes any one value of the set $\{0, 1, 2, 3, 4\}$.

The accelerator of the vehicle is expressed as,

$$AC = \{AC_x\}; \quad 1 \leq x \leq g \quad (37)$$

The initial value of the accelerator is set to 0, and the value of the accelerator lies between 0 and 1.

The representation of the brake is expressed as,

$$BR = \{BR_x\}, \quad 1 \leq x \leq g \quad (38)$$

The brake value lies between 0 and 1, and the initial value of the brake is set to 1.

Depending on the two boundary values in space, the speed of the vehicle is adjusted by the rider on the way of reaching the destination. Thus, the maximum speed of the vehicle at which the rider can drive the vehicle is represented as,

$$TM^x = \frac{TM_{UPP}^x - TM_{LOW}^x}{T_O} \quad (39)$$

where TM_{UPP}^x and TM_{LOW}^x are the maximum and the minimum values of the x th rider's position, and T_O is the off-time that is at its maximum at the end of the iteration. The speed limit of the gear is expressed as,

$$TM_x^{Ge} = \frac{TM^x}{|Ge|} \quad (40)$$

where $|Ge|$ is the count of gears, and TM^x is the maximum speed.

- (b) *Calculation of success rate*: The term “success rate” or otherwise the fitness measure is calculated based on the mean squared error (MSE), and it is expressed as,

$$MSE = \frac{1}{np} (\text{Target value} - \text{Estimated value}) \quad (41)$$

where np is the number of iterations.

- (c) *Determination of the leading rider*: The success rate is important in determining the position of the leading rider. The rider who is nearer to the location of the target is considered as the leading rider, as he has the possibility to reach the destination at first. The rider's position changes with respect to time; the leading rider cannot be fixed; and thus based on the success rate, any rider can obtain the leading position.
- (d) *Rider position update*: In each set, the rider position is updated to find the leading rider, which means the rider who has the highest possibility of reaching the destination at first.
- (i) *Position update of bypass rider*: The update of the position of the bypass rider occurs randomly as the bypass rider uses the common path without considering the leading riders. The update equation is expressed as,

$$l_{t+1}^{Br}(x, y) = \omega[l_t(\tau, y) * \vartheta(y) + l_t(\psi, y) * [1 - \omega(y)]] \quad (42)$$

where ω is a random number that lies between 0 and 1, and τ and ψ are the random numbers that lie between 1 and g . ϑ is the random number that lies between 0 and 1, and it is of the size $1 \times k$.

- (ii) *Position update of follower*: The co-ordinate selector is responsible for the update equation of the follower and is carried out for certain values of k .

$$l_{t+1}^{Fl}(x, e) = l^{Le}(Le, e) + [\cos(S_{x,e}^t) * l^{Le}(Le, e) * dist_x^t] \quad (43)$$

where e is the co-ordinate selector, $S_{x,e}^t$ is the steering angle of the x th rider in the e th co-ordinate, l^{Le} is the leading rider position, L is the leading rider index, and $dist_x^t$ is the distance needed to be traveled by the x th rider.

- (iii) *Position update of overtaker*: The position update process of the overtaker depends on three main factors, such as direction indicator, relative success rate, and the co-ordinate selector, and the relation is expressed as,

$$l_{t+1}^{OTK}(x, e) = l_t(x, e) + [DI_t(x) * l^{Le}(Le, e)] \quad (44)$$

where $l_{t+1}(x, e)$ is the position of the x th rider in the e th co-ordinate, and $DI_x(x)$ is the direction indicator of the x th rider.

- (iv) *Position update of the attacker*: The attacker follows the same procedure as that of the follower to reach the destination to be a winner. It updates all the co-ordinate values, and not only the selected values. The update equation of the attacker is expressed as,

$$l_{t+1}^{ATT}(x, y) = l^{Le}(Le, y) + [\cos(S_{x,y}^t) * l^{Le}(Le, y) * dist_x^t] \quad (45)$$

where $S_{x,y}^t$ is the steering angle of the x th rider in the y th co-ordinate.

- (e) *Calculation of success rate*: Once the update process is completed, the success rates of the riders are calculated. The rider position is updated with the new position of the rider who possesses the maximum success rate.
- (f) *Parameter update for the riders*: The optimal solution is obtained by updating the parameters of the rider.
- (g) *Activity counter*: In addition to the parameters, the activity counter is included during the position update of the rider. The value of the activity counter is 1, and in case the success rate at a particular time exceeds the previous time, the value of the activity counter is set to 0. The relation for the activity counter is expressed as,

$$ACT(x) = \begin{cases} 1 & \text{for } ss_{t+1}(x) > ss_t(x) \\ 0 & \text{else} \end{cases} \quad (46)$$

- (i) *Steering angle update*: Based on the activity counter, the steering angle is updated.

$$S_{x,y}^{t+1} = \begin{cases} S_{x+1,y}^t; & \text{for } \text{ACT}(x)=1 \\ S_{x-1,y}^t; & \text{for } \text{ACT}(x)=0 \end{cases} \quad (47)$$

- (ii) *Gear update*: The gear of the vehicle is updated based on the activity counter and the maximum possible gear a rider can apply, and it is expressed as,

$$G_x^{t+1} = \begin{cases} G_x^t + 1; & \text{for } \text{ACT}(x)=1, \text{ and } G_x^t + 1 \neq |G| \\ G_x^t - 1 & \text{for } \text{ACT}(x)=0, \text{ and } G_x^t + 1 \neq 0 \\ G_x^t & \text{else} \end{cases} \quad (48)$$

- (iii) *Acceleration update*: The acceleration is updated based on the gear of the vehicle, and it is expressed as,

$$AC_x^{t+1} = \frac{G_x^{t+1}}{|G|} \quad (49)$$

- (iv) *Brake value update*: The brake value is updated similar to that of the accelerator being subtracted from one.

$$BR_x^{t+1} = 1 - \frac{G_x^{t+1}}{|G|} \quad (50)$$

- (h) *Off-time riding*: The steps just cited are repeated till reaching the off-time, before which the leading rider is found.

To inherit the advantages of both the ROA and the MBO algorithms, the MBO is integrated into the ROA algorithm. The MBO is a nature-inspired metaheuristic algorithm that is generated with the mitigation characteristics associated with the butterflies. This algorithm is used, as it is effective in finding the solutions in case of continuous optimization problems. The position update of the monarch butterflies is done in two different ways, such as (i) the position update (generation of the off springs) with the use of the migration operator that can be adjusted by using the adjusting operator; (ii) the position of all the other butterflies is updated with the use of the butterfly adjusting operator. The implementation of the migration operator and the butterfly adjusting operator is done simultaneously. Hence, the MBO suits well for parallel processing, and in addition, it is capable of de-

veloping the trade-off between the diversification and the intensification that acts as an important aspect of metaheuristics. The standard equation of the MBO is given as,

$$l_{t+1}(x, e) = l_t(x, e) + \varsigma \times (WS_e - 0.5) \quad (51)$$

where $l_{t+1}(x, e)$ is the position of the monarch butterfly x in $(t+1)^{th}$ iteration, $l_t(x, e)$ is the position of the monarch butterfly x in t^{th} iteration, WS_e is the walk distance of the monarch butterfly, and ς is the weighting factor. Assuming Equation (44), the position update equation of the overtaker is the best solution, as it is determined with the maximum success rate as compared with other solutions of the ROA.

Substituting Equation (51) in (44),

$$l_{t+1}^{OTK}(x, e) = l_t(x, e) + [DI_t(x) * (l_t(x, e) + \varsigma \times (WS_e - 0.5))] \quad (52)$$

$$l_{t+1}^{OTK}(x, e) = l_t(x, e) + [DI_t(x) * l_t(x, e) + DI_t(x) * \varsigma \times (WS_e - 0.5)] \quad (53)$$

$$l_{t+1}^{OTK}(x, e) = [1 + DI_t(x)] l_t(x, e) + [DI_t(x) * \varsigma \times (WS_e - 0.5)] \quad (54)$$

Thus, using Equation (54), the position of the rider can be updated, based on the position of the rider, direction indicator of the rider, watch distance of the monarch butterfly, and the weighting factor. The pseudocode of the proposed Rider-MBO algorithm is detailed in Algorithm 1.

Algorithm 1: Pseudocode for the proposed Rider-MBO algorithm

Location estimation using Rider-MBO Algorithm

- 1 Input: positions of the riders in random $\rightarrow l_t$
 - 2 Output: Rider in a leading position $\rightarrow l^{Le}$
 - 3 **Start**
 - 4 **Population initialization**
 - 5 Rider parameter initialization: Steering angle $S_{x,y}^{t+1}$, Gear G_x^{t+1} , Accelerator AC_x^{t+1} , Brake BR_x^{t+1}
 - 6 Calculate the success rate ss_x
 - 7 **While** $t < T_O$
 - 8 **For** $x=1$ to g
 - 9 The position of the bypass rider is updated based on Equation (42)
 - 10 The position of the follower is updated based on Equation (43)
 - 11 The position of the overtaker is updated based on Equation (44)
 - 12 The position of the attacker is updated based on Equation (45)
 - 13 Ranking of riders based on the success rate
 - 14 Selection of rider with a maximum success rate
 - 15 Update steering angle, gear, accelerator, and brake
 - 16 Return the leading rider based on Equation (54)
 - 17 $t = t + 1$
 - 18 **End while**
 - 19 **Stop**
-

Results and Discussions

The results obtained by using the proposed Rider-MBO method and the existing methods of stock market prediction are explained in this section. The performance of the proposed Rider-MBO algorithm is analyzed by using two metrics, such as MSE and root mean squared error (RMSE).

Experimental setup

The experiment is carried out by using the MATLAB tool, which offers simple and effective implementation of the proposed system.

Simulation results

Figure 5 depicts the sample simulation results established by using the proposed method with a variation in the number of training iterations. Figure 5a shows the sample result at the training iteration of 19 among the 250 training iterations. With the increase in a number of training iterations, the accuracy of the proposed method increases in prediction. Figure 5b shows the sample result at the training iteration of 217 among the 250 training iterations. On comparing both the figures, Figure 5b provides better prediction and is observed to be close to the actual value.

Figure 6 shows the prediction analysis of the stock market open price with the original open price for the comparative methods and the proposed method. From the figure, it is clear that the proposed Rider-MBO-based Deep-ConvLSTM provides better prediction equal to that of the original open price of the stock market as compared with that of the conventional methods.

Database description

The stock market data of six companies,³⁴ such as Infosys, HCL Technologies, Wipro, Bharti Infratel, Larsen & Toubro Infotech, and Siemens from January 1, 2017 to December 31, 2018, are taken as the dataset to perform the analysis of the proposed system.

Evaluation metrics. The effectiveness of the proposed stock market prediction method is analyzed with the use of two measures, such as MSE and RMSE.

- (a) **MSE:** The MSE is the rate of an average of the errors in squares between the actual value and the estimated value, and is always positive, but not 0 due to its randomness. The expression for the MSE is given in Equation (41).
- (b) **RMSE:** RMSE is nothing, but the square root of the MSE, and thus, it is an estimation of the

square root of the difference between the actual value and the observed value.

$$\text{RMSE} = \sqrt{\frac{1}{np} (\text{Actual value} - \text{observed value})} \quad (55)$$

Comparative methods of stock market prediction

The comparative methods involved are the proposed Rider-MBO-based Deep-ConvLSTM, and the existing methods, such as Deep learning,²⁷ Auto Regression model,²⁰ and the nonlinear autoregressive network with exogenous inputs (NARX).³⁵ The performance of the proposed Rider-MBO algorithm is compared with the existing methods. The proposed Rider-MBO algorithm is developed by using the integration of the existing ROA and the MBO algorithms.

Comparative analysis of the proposed method

The comparative analysis of the proposed Rider-MBO and the existing methods is described in this section based on the evaluation metrics, such as MSE and RMSE.

Comparative analysis based on the data of Infosys. The comparative analysis of the proposed Rider-MBO method and the existing methods of stock market prediction using the data of Infosys is shown in Figure 7. Figure 7a shows the comparative analysis based on MSE, and Figure 7b shows the comparative analysis based on RMSE. For the delay of 10 days, the MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 14.8181, 57.0662, 159.6425, and 184.6425, respectively. Similarly, for the delay of 10 days, the RMSE of the methods, namely Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 3.8494, 7.5542, 12.635, and 17.6350, respectively. From the figure, it is clear that with the increase in delay, the MSE and the RMSE increases.

Comparative analysis based on the data of HCL Technologies. The comparative analysis of the proposed Rider-MBO method and the existing methods of stock market prediction using the data of HCL Technologies is shown in Figure 8. Figure 8a shows the comparative analysis based on MSE, and Figure 8b shows the comparative analysis based on RMSE. For the delay of 15 days, the MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning,

FIG. 5. Simulation results, **(a)** at iteration number 19, **(b)** at iteration number 217.

FIG. 6. Prediction analysis. MBO, monarch butterfly optimization; NARX, nonlinear autoregressive network with exogenous inputs.

Auto Regression model, and the NARX, is 12.8858, 323.5855, 1256386, and 1256411, respectively. Similarly, for the delay of 15 days, the RMSE of the methods, namely Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 3.5897, 17.9885, 1120.886, and 1125.886, respectively. From the figure, it is clear that with the increase in delay, the MSE and the RMSE increases.

Comparative analysis based on the data of Wipro. The comparative analysis of the proposed Rider-MBO method and the existing methods of stock market prediction using the data of Wipro is shown in Figure 9. Figure 9a shows the comparative analysis based on MSE, and Figure 9b shows the comparative analysis based on RMSE. For the delay of 15 days, the MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 23.4555, 68.1896, 1952.211, and 1977.211, respectively. Similarly, for the delay of 15 days, the RMSE of the methods, namely Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 4.8431, 8.2577, 44.1838, and 49.1838, respectively. From the figure, it is clear that with the increase in delay, the MSE and the RMSE increases.

Comparative analysis based on the data of Bharti Infratel. The comparative analysis of the proposed

Rider-MBO method and the existing methods of stock market prediction using the data of Bharti Infratel is shown in Figure 10. Figure 10a shows the comparative analysis based on MSE, and Figure 10b shows the comparative analysis based on RMSE. For the delay of 10 days, the MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 74.4353, 96.2252, 658.8153, and 683.8153, respectively. Similarly, for the delay of 10 days, the RMSE of the methods, namely Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 8.6276, 9.8094, 25.6674, and 30.6674, respectively. From the figure, it is clear that with the increase in delay, the MSE and the RMSE increases.

Comparative analysis based on the data of Larsen & Toubro Infotech. The comparative analysis of the proposed Rider-MBO method and the existing methods of stock market prediction using the data of Larsen & Toubro Infotech is shown in Figure 11. Figure 11a shows the comparative analysis based on MSE, and Figure 11b shows the comparative analysis based on RMSE. For the delay of 20 days, the MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 57.61, 1176.479, 1538.304, and 1563.304, respectively. Similarly, for the delay of 20 days, the RMSE of the methods, namely Rider-MBO-based Deep-ConvLSTM,

FIG. 7. Comparative analysis on the basis of data from Infosys, (a) based on MSE, (b) based on RMSE.

Deep learning, Auto Regression model, and the NARX, is 7.5901, 34.2999, 39.2212, and 44.2212, respectively. From the figure, it is clear that with the increase in delay, the MSE and the RMSE increases.

Comparative analysis based on the data of Siemens. The comparative analysis of the proposed Rider-MBO method and the existing methods of stock market prediction using the data of Siemens is shown in Figure 12. Figure 12a shows the comparative analysis

FIG. 8. Comparative analysis on the basis of data from HCL Technologies, (a) based on MSE, (b) based on RMSE.

based on MSE, and Figure 12b shows the comparative analysis based on RMSE. For the delay of 20 days, the MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 53.2538, 193.9928, 24884.26, and 24909.26, respectively. Similarly, for the delay of 20 days, the RMSE of the methods, namely Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, is 7.2975, 13.9281, 157.7475, and 162.7475, respectively. From the figure, it is clear that with the increase in delay, the MSE and the RMSE increases.

FIG. 9. Comparative analysis on the basis of data from Wipro, (a) based on MSE, (b) based on RMSE.

Comparative discussion

Table 1 shows the comparative analysis of the stock market prediction methods in the presence of data obtained from six live stock markets. The MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Infosys is 13.8647, 54.8042, 68.5545, and 93.5545, respectively. Similarly, the RMSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Infosys is 3.7235, 7.4029, 8.2797, and 13.2797, respectively. The MSE of the methods, such as Rider-MBO-based

FIG. 10. Comparative analysis on the basis of data from Bharti Infratel, (a) based on MSE, (b) based on RMSE.

Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from HCL Technologies is 7.2487, 173.1286, 69648.38, and 69673.38, respectively. Similarly, the RMSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from HCL Technologies is 2.6923, 13.1578, 263.9098, and 268.9098, respectively. The MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from

FIG. 11. Comparative analysis on the basis of data from Larsen & Toubro Infotech, (a) based on MSE, (b) based on RMSE.

FIG. 12. Comparative analysis on the basis of data from Siemens, (a) based on MSE, (b) based on RMSE.

Wipro is 11.4972, 52.7737, 1144.569, and 1169.569, respectively. Similarly, the RMSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Wipro is 3.3908, 7.2646, 33.8315, and 38.8315, respectively.

The MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Bharti Infratel is 58.2697, 65.98136, 240.0854, and

265.0854, respectively. Similarly, the RMSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Bharti Infratel is 7.6335, 8.1229, 15.4947, and 20.4947, respectively. The MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Larsen & Toubro Infotech is 52.1032, 695.9763, 1159.382, and

Table 1. Comparative analysis of stock market prediction methods

Companies	Evaluation metrics	Proposed Rider-MBO-based Deep-ConvLSTM	Deep learning	Auto regression model	NARX
Infosys	MSE	13.8647	54.8042	68.5545	93.5545
	RMSE	3.7235	7.4029	8.2797	13.2797
HCL technologies	MSE	7.2487	173.1286	69648.38	69673.38
	RMSE	2.6923	13.1578	263.9098	268.9098
Wipro	MSE	11.4972	52.7737	1144.569	1169.569
	RMSE	3.3908	7.2646	33.8315	38.8315
Bharti Infratel	MSE	58.2697	65.98136	240.0854	265.0854
	RMSE	7.6335	8.1229	15.4947	20.4947
Larsen & Toubro infotech	MSE	52.1032	695.9763	1159.382	1184.382
	RMSE	7.2182	26.3813	34.0496	39.0496
Siemens	MSE	19.5524	169.7411	554.5325	579.5325
	RMSE	4.4218	13.0284	23.5485	28.5485

MBO, monarch butterfly optimization; MSE, mean squared error; NARX, nonlinear autoregressive network with exogenous inputs; RMSE, root mean squared error.

1184.382, respectively. Similarly, the RMSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Larsen & Toubro Infotech is 7.2182, 26.3813, 34.0496, and 39.0496, respectively. The MSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Siemens is 19.5524, 169.7411, 554.5325, and 579.5325, respectively. Similarly, the RMSE of the methods, such as Rider-MBO-based Deep-ConvLSTM, Deep learning, Auto Regression model, and the NARX, for the data obtained from Siemens is 4.4218, 13.0284, 23.5485, and 28.5485, respectively.

Conclusion

The accurate stock market prediction is performed by using the proposed Rider-MBO-based Deep-ConvLSTM model. Initially, the data that are obtained from the live stock market are subjected to the feature extraction process, from which the required features needed in the prediction of the stock market are extracted. The features are extracted based on the technical indicators, which is responsible for the provision of the data required in the prediction of the stock market. The effective features are fed to the Deep-ConvLSTM model that is trained by using the proposed Rider-MBO algorithm, which is developed from the integration of the ROA and the MBO. The experimentation of the proposed method is carried out by using the metrics, such as MSE and RMSE, in the presence of the data collected from six live stock markets. The proposed method produces the minimum MSE of 7.2487 and RMSE of 2.6923 and proves the superiority of the method in effective stock market prediction as compared with that of the existing

methods. The proposed method of stock market prediction helps to invest cleverly to make good profits. Also, it helps to earn good profit.

Author Disclosure Statement

No competing financial interests exist.

Funding Information

No funding was received for this article.

References

- Buettner R. Predicting user behavior in electronic markets based on personality-mining in large online social networks: A personality-based product recommender framework. *The International Journal on Networked Business*. Springer, 2016. pp. 1–19.
- Nyce C. Predictive Analytics White Paper (PDF). American Institute for Chartered Property Casualty Underwriters/Insurance Institute of America. 2007.
- Mosavi A, Vaezipour A. Developing effective tools for predictive analytics and informed decision. University of Tallinn, Technical Report. 2013.
- Jones ER. Neural Networks' Role in Predictive Analytics. DM Review Special Report. 2008.
- Krauss C, Do XA, Huck N. Deep neural networks, gradient boosted trees, random forests: Statistical arbitrage on the S&P 500. *Eur J Oper Res*. 2017;259:689–702.
- Kumar V, Garg ML. A hybrid deep learning model for predictive analytics. *Int J Res Advent Technol*. 2018;6:27–31.
- Hiransha M, Gopalakrishnan EA, Menon VK, Soman KP. NSE stock market prediction using deep-learning models. *Procedia Comput Sci*. 2018;132: 1351–1362.
- Abraham A, Nath B, Mahanti PK. Hybrid intelligent systems for stock market analysis. In: *Proceedings of International Conference on Computational Science*. San Francisco, CA: Springer-Verlag, Berlin Heidelberg, 2001. pp. 337–345.
- Vijaya P, Raju G, Ray SK. Artificial neural network-based merging score for Meta search engine. *J Central South Univ*. 2016;23:2604–2615.
- Ding X, Zhang Y, Liu T, Duan J. Using structured events to predict stock price movement: An empirical investigation. In: *Empirical Methods in Natural Language Processing*. Doha, Qatar: Association for Computational Linguistics, 2014. pp. 1415–1425.
- Cun YL, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521:436–444.
- Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: *ICLR*. 2015. pp. 1–15.
- Bordes A, Chopra S, Weston J. Question answering with sub-graph embeddings. In: *Empirical Methods in Natural Language Processing*. Doha, Qatar: Association for Computational Linguistics, 2014. pp. 615–620.

14. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst*. 2012;25: 1097–1105.
15. Shi L, Teng Z, Wang L, et al. DeepClue: Visual interpretation of text-based deep stock prediction. *IEEE Trans Knowl Data Eng*. 2019;31:1094–1108.
16. Thomas R, Rangachar MJS. Fractional bat and multi-kernel-based spherical SVM for low resolution face recognition. *Int J Pattern Recogn Artif Intel*. 2017;31.
17. Ahmadi E, Jasemi M, Monplaisir L, et al. New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the Support Vector Machine and Heuristic Algorithms of Imperialist Competition and Genetic. *Expert Syst Appl*. 2018;94:21–31.
18. Evans C, Pappas K, Xhafa P. Utilizing artificial neural networks and genetic algorithms to build an algo-trading model intra-day foreign exchange speculation. *Math Comput Model*. 2013;58:1249–1266.
19. Sun BQ, Guo H, Karimi HR, et al. Prediction of stock market futures based on fuzzy sets and multivariate fuzzy time series. *Neurocomputing*. 2015; 151:1528–1536.
20. Efendi R, Arbaiy N, Deris MM. A new procedure in stock market forecasting based on fuzzy random auto-regression time series model. *Inform Sci*. 2018;441:113–132.
21. Khedr AE, Yaseen N. Predicting stock market behavior using data mining technique and news sentiment analysis. *Int J Intell Syst Appl*. 2017;9:22.
22. Weng B, Ahmed MA, Megahed FM. Stock market one-day ahead movement prediction using disparate data sources. *Expert Syst Appl*. 2017; 79:153–163.
23. Chen YJ, Chen YM, Lu CL. Enhancement of stock market forecasting using an improved fundamental analysis-based approach. *Soft Comput*. 2017; 21:3735–3757.
24. Pang X, Zhou Y, Wang P, et al. An innovative neural network approach for stock market prediction. *J Supercomput*. 2018;1–21.
25. Chandar SK. Stock market prediction using subtractive clustering for a neuro fuzzy hybrid approach. *Cluster Comput*. 2019;22:13159–13166.
26. Singh R, Srivastava S. Stock prediction using deep learning. *Multimed Tools Appl*. 2017;76:18569–18584.
27. Pathak A, Shetty NP. Indian stock market prediction using machine learning and sentiment analysis. In: *Computational Intelligence in Data Mining*, Springer, 2019. pp. 595–603.
28. Zhang X, Zhang Y, Wang S, et al. Improving stock market prediction via heterogeneous information fusion. *Knowl Based Syst*. 2018;143:236–247.
29. Binu D, Kariyappa BS. RideNN: A new rider optimization algorithm-based neural network for fault diagnosis in analog circuits. *IEEE Trans Instrum Meas*. 2019;68:2–26.
30. Wang GG, Deb S, Zhao X, Cui Z. A new monarch butterfly optimization with an improved crossover operator. *Oper Res*. 2018;18:731–755.
31. List of technical indicators. Available online at www.tradingtechnologies.com/help/xstudy/technical-indicator-definitions/triple-exponential-moving-average-tema/ (last accessed October 20, 2018).
32. Chang X, Wang Q, Liu Y, Wang Y. Sparse regularization in fuzzy C-means for high-dimensional data clustering. *IEEE Trans Cybern*. 2017;47:2616–2627.
33. Shi X, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation Nowcasting. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*. 2015, pp. 802–810.
34. Stock Market Data. Available online at www.moneycontrol.com/stocks/histstock.php?ex=N&sc_id=AMF&mycomp=Apple%20Mutual%20Fund (last accessed August 18, 2018).
35. Ercan H. Baltic Stock Market Prediction by Using NARX. In: *Proceedings on 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*. Lviv, Ukraine: IEEE, 2017.

Cite this article as: Kelotra A, Pandey P (2020) Stock market prediction using optimized deep-ConvLSTM Model. *Big Data* 8:1, 5–24, DOI: 10.1089/big.2018.0143.

Abbreviations Used

ADX	=	average directional index
AI	=	artificial intelligence
ANN	=	artificial neural network
ATR	=	average true range
CCI	=	commodity channel index
CEC	=	constant error carousel
CRM	=	customer relationship management
Deep-ConvLSTM	=	deep convolutional long short-term memory
DEMA	=	double exponential moving average
EMA	=	exponential moving average
FC-LSTM	=	fully connected LSTM
FR-AR	=	fuzzy random auto-regression
GA	=	genetic algorithm
MACD	=	moving average convergence divergence
MOM	=	momentum
MSE	=	mean squared error
NARX	=	nonlinear autoregressive network with exogenous inputs
NN	=	neural network
OBV	=	on balance volume
Rider-MBO	=	Rider-based monarch butterfly optimization
ROA	=	rider optimization algorithm
ROCR	=	rate of change
RMSE	=	root mean squared error
RSI	=	relative strength index
SAR	=	parabolic SAR
SMA	=	simple moving average
Sparse-FCM	=	Sparse-Fuzzy C-Means
SVM	=	support vector machine
TEMA	=	triple exponential moving average
TRIX	=	triple exponential moving average oscillator
TSF	=	time series forecast