International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

An Optimal Deep Learning Model With Effective Feature Learning Mechanism For Stock Market Prediction

D. Syinthiya[1], Dr. K. Sujith[2]
[1]Research Scholar in Computer Science
PG & Research Department of Computer Science
Annai College of Arts and Science
(Affiliated to Bharathidasan University, Tiruchirappalli)
Kovilacheri, Kumbakonam, Thanjavur – 612503
Tamilnadu, India
syinthiyad86@gmail.com
[2]Associate Professor of MCA,
Research Advisor in PG & Research Department of Computer Science,
Annai College of Arts & Science
(Affiliated to Bharathidasan University, Tiruchirappalli),
Kovilacheri, Kumbakonam, Thanjavur - 612503
Tamilnadu, India.
mailme.sujith@gmail.com

Abstract:

The stock market has considered the active research fields today, and forecasting its behaviour is an enormous necessity. Predicting the stock market is complex, necessitating a thorough examination of data patterns. Correct forecasting outcomes can provide significant insight to investors, lowering investment risk. This paper proposes a novel Weight and Bias Tuned Long Short-term Memory (WBTLSTM) with an efficient feature extraction model, Bilateral ReLU-based Two-Dimensional Convolutional Neural Network (BR2DCNN), for stock price prediction (SPP). First, the stock data was collected from the publicly available dataset. Then the missing values imputation and data normalization is performed on the collected dataset. The preprocessed dataset extracts the most relevant features using BR2DCNN. Finally, the future stock events are predicted using the WBTLSTM. The weights and biases are tuned with the help of the Enhanced Butterfly Optimization Algorithm (EBOA). Experimental findings prove that the proposed one achieves superior outcomes compared to the conventional methods regarding some performance metrics.

Keywords: Stock Price Prediction, Preprocessing, Feature Extraction, Normalization, and Deep Learning.

1. INTRODUCTION

Finance marketplaces are among the most interesting technological advances of our time. These financial markets have a massive impact on many industries, including technology and business. The two major procedures are followed by the Investors to make stock market conclusions to increase returns while reducing risk. The evolution of stock market forecasting has acquired prominence among expert analysts and investors [1]. The companies sell stocks on a stock market to extend their businesses and the investors or buyers can buy the companies stocks and sell those stocks to other investors at a predetermined rate. The investors can get the profit by selling those stocks at any time with an additional cost [2]. Stock price fluxes are connected with several variables, including market expectation, macroeconomic circumstances, and trust in the concern's managing and maneuvers. The advent of technology allows the public to access more information in a shorter period [3]. However, producing an accurate prediction remains complex and demanding [4, 5]. The efficient market hypothesis adherents assert that precise predicting of future stock values is impossible. Propositions, however, show how complicated algorithms and sophisticated and optimally developed predictive models enable one to estimate future stock values with great precision [6].

The stock prediction has become more efficient with the emergence of machine learning (ML) models [7]. Many ML techniques, including random forest, support vector machine (SVM), and neural network (NN) approaches, are utilized for improving the classic market estimate strategies [8]. These approaches help deal with stock market data's large dimensionality and non-linearity [9]. All models mentioned above use handmade characteristics extracted from raw data as input. However, the creation of handcrafted features is a time-consuming and domain-specific procedure. Furthermore, as the size of the feature space grows, the training duration of the models increases, and the model outputs become increasingly difficult to comprehend [10].

As a result, recent breakthroughs in ML, such as deep learning (DL) [20] based techniques, have been focused on by numerous researchers in order to anticipate the stock effectively. It can extract essential elements from complex and noisy data and detect latent nonlinear correlations without relying on the human skill or economic assumptions [11]. However, improvements in conventional DL approaches are still required because typical DL approaches choose the hyperparameter to train the network at random [21] [22]. Randomly choosing these hyperparameters increases the computational process's complexity and introduces an overfitting problem. Thus, using an efficient neural network, the proposed system uses a novel optimal tuning-based DL approach to predict the stock and efficiently extract the features from the dataset. The main objectives of the proposed work are explained as follows:

- The missing values imputation and data normalization pre-processing are performed to impute the missing values in the dataset and to normalize the dataset
- Employing BR2DCNN to extract the most relevant feature from the dataset increases the prediction accuracy of the classification system and reduces its overfitting problem.
- Utilizing WBTLSTM to predict the future stock events and the weight and bias are tuned based on EBOA to minimize the loss in SPP and avoid the saturation of the network.

The rest portion of the manuscript is outlined in the following manner. Section 2 surveys recent methodologies associated to the SPP. The brief discussion of the proposed method is given in section 3. The outcomes of the proposed and existing schemes for SPP is given in section 4 and the conclusions of the suggested framework is given in section 5.

2. RELATED WORK

This section discusses the recent methodologies for SPP using machine and DL models. The shortcoming of the existing schemes and the solutions offered by the proposed system are also discussed at the end of the section.

Yang Li and Yi Pan [12] presented an SPP system using an ensemble deep-learning model. Firstly, the system collected the data from the New York City dataset and then the collected data was preprocessed by removing the null values and combining the stock data in the dataset. Finally, long short-term memory (LSTM) as well as gated recurrent unit were utilized for SPP. The outcomes showed that the method performed better than the existing state of art schemes. **Kyung Keun Yun *et al.* [13]** suggested a genetic algorithm based on extreme gradient boosting (GA-XGBoost) having three-stage feature learning for SPP. The data was collected from the KOSPI dataset, and then the feature set expansion was done, which contained three stages: technical indicators generation, data preparation and normalization and GA-based optimal feature selection. The selected features were fed into the XGBoost model for SPP. The system achieved an accuracy of 93.28% that was greater than the previous related schemes. **Atharva Shah *et al.* [14]** presented a hybrid approach called convolution neural network (CNN) with LSTM for SPP. The stock market data was initially collected from the Nifty 50 stock market index. Then the collected data was given to the CNN for feature extraction, and finally, the extracted feature set was passed into LSTM for SPP. The results showed that the method attained the mean absolute error (MAE) of 2.54% for the collected stock data of 10 years, which were better than the traditional learning schemes.

Hadi Rezaei *et al.* [15] presented a hybrid deep-learning model for SPP. Initially, the data was collected from publicly available datasets, namely Dow Jones, DAX, S&P500 and Nikkei225 index. The collected data was fed into hybrid classifiers such as Empirical Mode Decomposition (EMD) as well as Complete Ensemble EMD (CEEMD) for SPP. Initially, the data was collected from the Indian stock exchange database. The method attained a root mean squared error of 194.7, 108.56, 163.56 and 14.88 for Nikkei255, DAX, Dow Jones, and S&Ps00 datasets, which was better than the previous schemes. **Srivinay *et al*. [16]** presented a hybrid DL scheme including prediction rule ensembles (PRE) as well as deep neural networks (DNN) for SPP. The collected data's technical indicators were considered to detect the stock price's uptrends. Then different rules were generated using PRE for stock prediction. Finally, DNN was utilized for SPP, in which the system attained an RMSE between 5 to 7%, which was lower than previous related schemes.

The methods mentioned above efficiently predict the future stock price with satisfactory results. However, some works manually extract the features from the dataset. Manual feature extraction takes more time to train the network and decreases the prediction rate. Thus, most of the author uses CNN methods to extract new features from the dataset. However, it cannot capture all the relevant spatiotemporal information because the convolutional kernel moves in a single direction. Thus the proposed system uses Two dimensional CNN to extract all temporal and spatial features from the dataset. In addition, some authors used an ML algorithm to predict the stock, but it needed to be more suitable for a large amount of data. Most of the author uses a DL algorithm to handle a large amount of stock data, which effectively predicts future stock events. However, the biggest challenge in the DL algorithm is tuning its hyperparameter. Because the weights and biases in the DL algorithm are chosen randomly, it increases the computational complexity and maximises the error values. Hence in this proposed work, the parameters are tuned based on EBOA in the LSTM algorithm.

3. PROPOSED METHODOLOGY

This paper proposes a novel WBTLSTM-based stock prediction with efficient feature extraction using BR2DCNN. Initially, the stock market data was collected from publicly available datasets. Then the pre-processing operations, such as missing value imputation and data normalization, are performed on the collected data. Next, feature extraction is performed using BR2DCNN to extract the more relevant features from the pre-processed dataset. Finally, the stock prediction is made based on WBTLSTM. Herein, the weights and biases of the LSTM are optimally chosen by EBOA. The proposed system's flow diagram is illustrated in fig. 1.

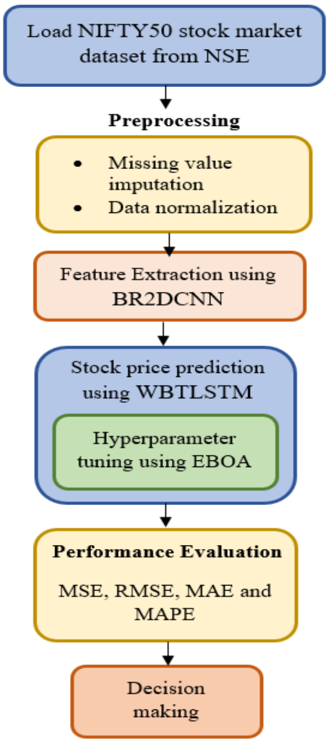


Figure 1: Proposed system's flow diagram

3.1 Date Preprocessing

Initially, the stock market data was collected from publicly available datasets. Then preprocessing is applied to the collected dataset to minimize the error and increase the prediction rate. Herein, the missing values imputation and data normalization preprocessing is applied to the collected dataset to generate accurate and reliable data while reducing the time necessary to analyze raw data. Initially, the missing values in the data set are replaced based on the mean imputation. It is an approach that replaces a missing value on a variable with the mean of the available values. Following that, the obtained dataset is normalized. Scaling data from its initial range to ensure every value falls between 0 and 1 is referred to as normalization. The primary goal of the data normalization procedure is to provide high-quality data that can be fed into a prediction system. It increases gradient descent speed and accuracy. By applying a linear modification to the initial data, min-max normalization is often employed for scaling data between specific ranges. It can be stated as follows:

$$Norm\left(\breve{C}_{DS}\right)=\frac{\breve{C}_{DS}-\left(\breve{C}_{DS}\right)_{\min}}{\left(\breve{C}_{DS}\right)_{\max}-\left(\breve{C}_{DS}\right)_{\min}} \quad (1)$$

Where, $\breve{C}_{DS}$ refers to the input collected stock data, $\left(\breve{C}_{DS}\right)_{\min}$ and $\left(\breve{C}_{DS}\right)_{\max}$ signifies minimum and maximum data in the input dataset.

3.2 Feature Extraction

The feature extraction process extracts the more informative features from the collected dataset by avoiding the irrelevant features. It can fine-tune extensive databases to achieve greater accuracy and robustness. In this proposed work, the feature extraction is done with the help of a BR2DCNN to extract the most relevant features from the dataset and to increase the prediction rate. The 2DCNN extracts the features with the kernel size of (3 × 3), and the kernel moves in two directions to generate a feature map. The network includes three layers to perform its extraction process: convolution, pooling, and fully connected layer. The sigmoid and tanh functions are utilized in the 2DCNN activation function, resulting in the gradient vanishing problem [19]. Rectified linear unit (ReLU) eliminates the issue of vanishing gradients, one of the primary factors behind 2DCNN's current rebirth. One of the most significant drawbacks of ReLU is the existence of dead neurons. The original input is forced to contend with the constant term 0, resulting in nonlinear transformation abilities, with specific neurons staying untrained during the training process. The suggested system employs the Bilateral ReLU activation function to avoid the issue of neuron death during training. As a result, this update in 2DCNN has been dubbed BR2DNN. The following is an explanation of the BR2DCNN process:

The kernel of convolution examines the input features regularly. The input preprocessed dataset is first fed into the 2D convolutional kernel, with each cell representing a bias vector as well as a weight coefficient. Then the input features are combined and multiplied by the matrix elements in the arena of reception. It is mathematically expressed as follows:

$$\overline{FM}''_m = \psi^*\left(B_m + \sum_{i=0}^{IN_m-1} \overline{\omega}_m * \overline{Y}_m\right) \tag{2}$$

Where, $\overline{FM}''_m$ represents the feature map, $*$ refers to the 2D convolution, IN_m indicates the total numnber of instances from the preprocessed dataset, $\overline{\omega}$ refers to the weight of the filter, $\overline{Y}$ proffers the preprocessed dataset, B_m signifies the bias value. Moreover, the term ψ^* denotes the to Bilateral ReLU activation function, which avoids the problem of gradient saturation in conventional neural network. It is mathematically expressed as follows:

$$\psi^*\left(\overline{Y}\right) = \min\left(\overline{T}''_{\max}, \max\left(\overline{T}''_{\min}, \overline{Y}\right)\right) \tag{3}$$

Where, $\overline{T}''_{\min}$ and $\overline{T}''_{\max}$ signifies the constant input values $\left(\overline{Y}\right)$ for Bilateral ReLU with $\overline{T}''_{\min} < 0$ and $\overline{T}''_{\max} > 1$, respectively. The output feature map of the convolutional layer will be transmitted to the max-pooling layer for reducing extracted feature maps' size into lower dimensions and avoiding the overfitting issues in the network. It employs the highest value in the convolution filter with a defined size before subsampling. The pooling area is chosen in the same way that the stages of the convolution kernel's scanning feature map are, using the pooling size, step size, as well as filling parameters. Following the max-pooling operation, the acquired feature map is given into the flattening layer, which turns the pooling layer's input matrix into a vector. Finally, the fully connected layer gets the feature vector and is given as input to the classifier to predict the stock.

3.3 Stock Prediction

Stock prediction aims at predicting the future worth of an organization's financial stocks. WBTLSTM are employed in this proposed work to estimate upcoming market events. The LSTM is a kind of recurrent neural network (RNN). It maintains the strong prediction performance of an RNN network for time series data. It establishes the input gate, forgetting gate, and output gate simultaeously to deal with disappearing gradients and gradient explosion by reciting and forgetting past knowledge. It comprises distinct memory blocks acknowledged as cells. The two states namely cell and the hidden states are forwarded to the following cell. Although the LSTM achieves better performance in future stock prediction issues, the major difficulty in LSTM is fine-tuning its weights and bias because random weights and bias initialization increase the training time and increase the overfitting/underfitting issues. So the proposed system uses EBOA to optimally select the weight and bias of the LSTM to avoid overfitting issues. This improvization in conventional LSTM is termed as WBTLSTM. The WBTLSTM' stpes are explained as follows:

Step 1: Weights and Bias optimization

The weights as well as biases of the LSTM network are initially optimized utilizing the Enhanced Butterfly Optimisation Algorithm (EBOA). The BOA is a metaheuristic swarm-based algorithm. Butterflies use their smelling sense to discover food from faraway locations and to distinguish between distinct smells in a specific region. Butterflies foraging is the core strategy of the BOA optimization algorithm, which uses their sense of smell to locate food. Despite its improved performance, BOA still faces numerous challenges. For example, the traditional BOA assumes the population randomly, increasing population diversity.

Furthermore, while addressing highly complicated problems, the basic BOA is prone to falling into the local optimum, and the convergence speed decreases dramatically in the late iteration period. As a result, the proposed method employs a chaotic Tent map to boost population diversity and randomization. Furthermore, the suggested system employs Levy flight to maintain a healthy balance between exploration and exploitation, enhancing its global search capabilities and effectively preventing it from falling into a local optimum. These two incorporations in conventional BOA are termed EBOA. These are explained as follows:

The fragrance is represented in this EBOA as a function of the significant concentration of stimulation of the butterflies, which is mathematically determined by:

$$\overline{RR}_p = \delta_c \times \overline{SM}^{ex} \tag{4}$$

Where, $\overline{RR}_p$ indicates the amount of fragrance produced by p^{-th} butterfly, $\overline{SM}$ denotes the stimulus' strength, δ_c refers to the sensory modality, and $ex-$signifies called power exponent. After explaining the essential variables, the population of the butterflies are initialized using the Tent chaotic map for increasing the population's diversity. A tent

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

chaotic map is a two-dimensional linear map, and the mathematical model of tent chaotic map for population initialization is expressed as follows:

$$\overline{E}_{l+1} = \begin{cases} 2\,\overline{E}_l\,, & 0 \le \overline{E}_l \le 0.5 \\ 2\left(1-\overline{E}_l\right), & 0.5 < \overline{E}_l \le 1 \end{cases} \quad ()$$

Where, $\overline{E}_l$ refers to the current population of l^{-th} individual and $\overline{E}_0$ indicates the random number between o and 1. Then compute the fitness of the individual minimizing the mean squared error. This is expressed as follows:

$$Fit\_Func\left(\overline{E}_l\right) = \min\left(MSE\right) \quad (5)$$

$$MSE = \frac{1}{i_N}\sum_{o=1}^{i_N}\left(A''_{vl} - P''_{vl}\right) \quad (6)$$

Where, A''_{vl} and P''_{vl} indicates the actual and predicted values and $i_N -$ refers to the number of samples in the training dataset. After that, update the position of the butterfly globally and locally. During a global search, the butterfly takes a step towards the best position based on the fitness value of the objective function. It is expressed as follows:

$$\overline{E}_l^{\tau+1} = \overline{E}_l^{\tau} + \left(R_{na}{}^2 \times G\_B^* - \overline{E}_l^{\tau}\right) \times \overline{RR}_p \times \mathrm{LF_S}(\eta) \quad (7)$$

Where, $\overline{E}_l^{\tau}$ refers to the l^{-th} individual position at τ^{-th} iteration, G_B^* indicates the global best solutions in the current iteration amongst all, R_{na} denotes the arbitrary number between 0 to 1 that generates a specific degree of randomness in the search horizon, and $\mathrm{LF_S}(\eta)$ refers to the levy flight strategy to balance the global search and to avoid the local optimal issue. Levy flight is a random walk where the steps are defined by their lengths, which have a particular probability distribution, and it is expressed as follows:

$$\mathrm{LF_S}(\eta) \sim |\eta|^{-1-\alpha} \quad (8)$$

$$\eta = \frac{u_j}{\left|v_j\right|^{1/\alpha}} \quad (9)$$

Where, $\alpha\left(0 < \alpha \le 2\right)$ indicates an index, η indicates the step length, u_j and v_j are drawn from normal distributions. Then the local search can be represented as:

$$\overline{E}_l^{\tau+1} = \overline{E}_l^{\tau} + \left(R_{na}{}^2 \times \overline{E}_m^{\tau} - \overline{E}_n^{\tau}\right) \times \overline{RR}_p \quad (10)$$

Where $\overline{E}_m^{\tau}$ and $\overline{E}_n^{\tau}$ represents the positions of any two butterflies from the same swarm. The iteration process is terminated if the optimal weights and biases are established. Otherwise, the iteration process is repeated until the best optimal solutions are obtained.

Step 2: After optimally chosen weights and bias, the mathematical formulation of various operations performed in LSTM is given below:

$$\hat{f}\hat{g}_{\varphi} = \psi^*\left(W_{\hat{f}\hat{g}}^{ow} \cdot \left[\hat{h}_{\varphi-1}, \overline{EF}_{\varphi}\right] + B_{\hat{f}\hat{g}}^{ob}\right) \quad (11)$$

$$\hat{i}\hat{g}_{\varphi} = \psi^*\left(W_{\hat{i}\hat{g}}^{ow} \cdot \left[\hat{h}_{\varphi-1}, \overline{EF}_{\varphi}\right] + B_{\hat{i}\hat{g}}^{ob}\right) \quad (12)$$

$$\hat{o}\hat{g}_{\varphi} = \psi^*\left(W_{\hat{o}\hat{g}}^{ow} \cdot \left[\hat{h}_{\varphi-1}, \overline{EF}_{\varphi}\right] + B_{\hat{o}\hat{g}}^{ob}\right) \quad (13)$$

Where, $\hat{f}\hat{g}, \hat{i}\hat{g}, and\ \hat{o}\hat{g}$ refers to the forget, input and output gates, respectively, $\hat{h}_{\varphi-1}$ indicates the prior hidden layer units at time step φ, $\overline{EF}$ denotes the extracted feature set, $W_{\hat{f}\hat{g}}^{ow}, W_{\hat{i}\hat{g}}^{ow}, and\ W_{\hat{o}\hat{g}}^{ow}$ and $B_{\hat{f}\hat{g}}^{ob}, B_{\hat{i}\hat{g}}^{ob}, and\ B_{\hat{o}\hat{g}}^{ob}$ represents the optimal weights and bias for the forget gate, input gate, and output gate, which is selected based on the EBOA, and ψ^* signifies the Bilateral ReLU activation function, which is calculated by suing equation (3).

Step 3: Compute the cell state $\left(\hat{c}\hat{g}\right)$ using the following equation.

$$\hat{c}\hat{g}_{\varphi} = \left(W_{\hat{c}\hat{g}}^{ow} \cdot \left[\hat{h}_{\varphi-1}, \overline{EF}_{\varphi}\right] + B_{\hat{c}\hat{g}}^{ob}\right) \quad (14)$$

Then, update the current cell state' information $\left(\hat{S}_{\varphi}\right)$. It is accomplished by pointwise multiplying the forget gate output and the current cell state. If $\left(\hat{S}_{\varphi}\right)$ is 0, The multiplication will also result in zero, indicating that the initial value has been

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

completely dropped, otherwise, $\left(\hat{S}_{\varphi}\right)$ is 1, it is expressed as follows:

$$\hat{S}_{\varphi} = \hat{f}\hat{g} \otimes \hat{c}\hat{g}_{\varphi-1} + \hat{i}\hat{g} \otimes \hat{f}\hat{g} \tag{15}$$

Step 4: At last, the hidden state is calculated using equation (16) and the outcomes of the WBTLSTM predict the future stock events.

$$\hat{h}_{\varphi-1} = \hat{o}\hat{g}_{\varphi} * \psi^{*}\left(\hat{S}_{\varphi}\right) \tag{16}$$

4. RESULTS AND DISCUSSION

This section shows the performance of the proposed novel weights and bias-tuned LSTM-based SPP with efficient feature extraction using BR2DCNN. The system was built using the Python Keras working platform and the TensorFlow library backend. The program code was emulated utilizing a GPU with 12.72 GB RAM and a 107.77 GB Google Cloud disc. The dataset used for the proposed work is explained first, and then the outcomes of the proposed and existing methods are analyzed and the conclusion is given.

4.1 Dataset Descriptions

The system collects the data NIFTY 50 stock price data from National Stock Exchange of India (NSE) which includes the day level stock data (pricing and trading) from December 10, 2011, to December 10, 2021. The dataset includes 50 stock prices per day, including the closing, opening, highest, lowest, and volume-weighted average price. To train and validate its performance, the system gathered stock price data from the Kaggle dataset repository. To achieve the best performance results, we split the dataset using different training and testing splitting ratios (60-80) for three distinct window sizes (30, 60, and 90).

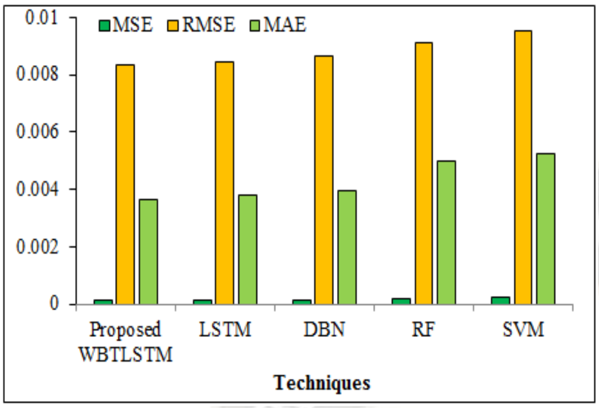
4.2 Performance Evaluation

Here, the outcomes of the proposed WBTLSTM are investigated against the conventional LSTM, Deep belief Network (DBN), Random Forest (RF), and Support Vector Machine (SVM). The performance evaluation is done concerning the Mean Squared Error (MSE), Root MSE (RMSE), MAE, and Mean Absolute Percentage Error (MAPE). Table 1 shows the performance of the proposed model with optimum window size (OWS) and optimum splitting ratios (OSR).

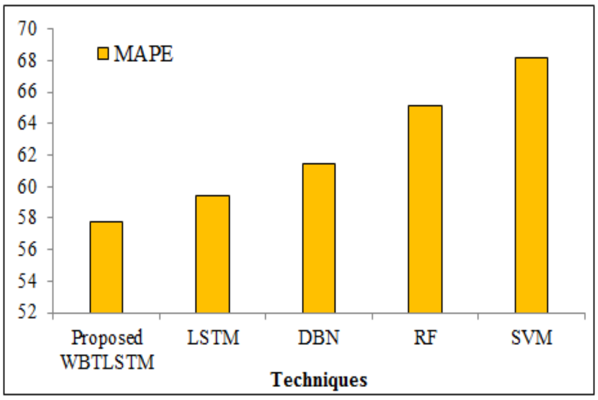
Table 1: Results analysis of the proposed model

Proposed Model		MSE	RMSE	MAE	MAPE
Window 30	Split 60	0.0005048	0.0226883	0.0049917	117.057
	Split 65	0.0000077	0.0042416	0.0028831	70.137
	Split 70	0.0000116	0.0046778	0.0032589	45.336
	Split 75	0.0000131	0.0048221	0.0033176	36.21
	Split 80	0.0000146	0.0049901	0.0033737	30.647
Window 60	Split 60	0.0004852	0.0222524	0.0045814	114.666
	Split 65	0.0000076	0.0042299	0.0028786	66.597
	Split 70	0.0000111	0.0046256	0.0031959	43.233
	Split 75	0.0000132	0.0048479	0.0033433	35.213
	Split 80	0.0000181	0.0053316	0.0036402	29.616
Window 90	Split 60	0.0005232	0.0230885	0.0058211	109.724
	Split 65	0.0000111	0.0046105	0.0031556	64.291
	Split 70	0.0000108	0.0045939	0.0031621	41.878
	Split 75	0.0000141	0.0049366	0.0033832	34.261
	Split 80	0.0000172	0.0052466	0.0035496	28.036

Table 1 shows the results of the proposed WBTLSTM with the existing schemes regarding MSE, RMSE, MAE, MSLE, and MAPE metrics. In this table, for OWS of 30 and OSR of 60, the proposed one has an MAE, RMSE, MAE, and MAPE of 0.0005048, 0.0226883, 0.0049917, and 117.057. For OWS of 30 and OSR of 65, the proposed one has an MAE, RMSE, MAE, and MAPE of 0.0000077, 0.0042416, 0.0028831, and 70.137. For OWS of 30 and OSR of 70, the proposed one has an MAE, RMSE, MAE, and MAPE of 0.0000116, 0.0046778, 0.0032589, and 45.336. For OWS of 30 and OSR of 75, the proposed one has an MAE, RMSE, MAE, and MAPE of 0.0000131, 0.0048221, 0.0033176, and 36.21, and for OWS of 30 and OSR of 80, the proposed one has an MAE, RMSE, MAE, and MAPE of 0.0000146, 0.0049901, 0.0033737, and 30.647, which showed that the proposed one has a small error value for the OWS 30 and the splitting ratio of 65 to 80. Similarly, the proposed one has minimal error values for the remaining OWS of 60 and 90 and the OSR of 65 to 80. If the system has low error values, then the system is regarded as a sound system. From the table, it is concluded that the proposed one is superior in error minimization for predicting stock prices. Next, the average efficiency of the proposed WBTLSTM is weighted against the conventional methods, such as LSTM, DBN, RF, and SVM, based on the same error metrics shown in Figure 2.



(a)



(b)

Figure 2: Efficiency analysis of the proposed one with conventional methods based on (a) MES, RMSE, and MAE and (b) MAPE

Figure 2 demonstrates the efficiency of the proposed one with the conventional methods in terms of MSE, RMSE, MAE, and MAPE metrics. First, consider the MSE metric; the proposed one has a lower MSE of 0.0001109, but the existing LSTM, DBN, RF, and SVM have a higher MSE of 0.0001234, 0.0001457, 0.0001956, and 0.0002365. Similarly, the proposed one achieves the RMSE, MAE, and MAPE of 0.0083456, 0.0036357, and 57.793, which are lower than the previous models. Thus, the overall results showed that the proposed system outperforms previous related schemes for SPP. Because the proposed work applies preprocessing methods on the dataset and the proposed system uses BR2DCNN to effectively extract the features from the preprocessed dataset, that enhances the prediction performance of the classifier without any computation burden and overfitting issues. Additionally, the parameter tuning in LSTM results in a minimized loss in SPP and avoids the gradient saturation problem compared to existing algorithms.

5. CONCLUSION

This paper suggests a novel WBTLSTM-based SPP model with an efficient feature extraction scheme, BR2DCNN. The proposed system mainly comprises three phases: preprocessing, feature extraction, and stock prediction. The data from India's NSE with ten years of historical stock price data (NIFTY 50 index) is utilized to analyze the effectiveness of the proposed system. Initially the system's results are discussed concerning the MSE, MAE, RMSE, and MAPE metrics by varying different OWS and OSRs. The outcomes proved that the proposed method attains the lowest MSE, RMSE, MAE, and MAPE of 0.0000076, 0.0042299, 0.0028786, and 28.036. Next, the performance of the proposed WBTLSTM is investigated against the conventional LSTM, DBN, RF, and SVM based on the same metrics. Herein also, the proposed one has a minimal error rate than the conventional methods. Thus, the outcomes indicated that the proposed one performs better than the conventional methods. In the future, we plan to increase the proposed work's prediction rate by introducing efficient feature selection techniques, and predictions will be made using an advanced deep-learning approach.

REFERENCES

1. Kumar, D., Sarangi, P. K., & Verma, R. (2022). A systematic review of stock market prediction using machine learning and statistical techniques. Materials Today: Proceedings, 49, 3187-3191.
2. Dey, P., Hossain, E., Hossain, M. I., Chowdhury, M. A., Alam, M. S., Hossain, M. S., & Andersson, K. (2021). Comparative analysis of recurrent neural networks in stock price prediction for different frequency domains. Algorithms, 14(8), 251.
3. Nghiem, H., Muric, G., Morstatter, F., & Ferrara, E. (2021). Detecting cryptocurrency pump-and-dump frauds using market and social signals. Expert Systems with Applications, 182, 115284.
4. Thakkar, A., & Chaudhari, K. (2021). A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. Expert Systems with Applications, 177, 114800.
5. Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. Applied System Innovation, 4(1), 9.
6. Wu, J. M. T., Li, Z., Srivastava, G., Tasi, M. H., & Lin, J. C. W. (2021). A graph-based convolutional neural network stock price prediction with leading indicators. Software: Practice and Experience, 51(3), 628-644.
7. Sharaf, M., Hemdan, E. E. D., El-Sayed, A., & El-Bahnasawy, N. A. (2021). StockPred: a framework for stock Price prediction. Multimedia Tools and Applications, 80, 17923-17954.

8. Leippold, M., Wang, Q., & Zhou, W. (2022). Machine learning in the Chinese stock market. Journal of Financial Economics, 145(2), 64-82.
9. Ma, C., Zhang, J., Liu, J., Ji, L., & Gao, F. (2021). A parallel multi-module deep reinforcement learning algorithm for stock trading. Neurocomputing, 449, 290-302.
10. Gunduz, H. (2021). An efficient stock market prediction model using hybrid feature reduction method based on variational autoencoders and recursive feature elimination. Financial innovation, 7(1), 28.
11. Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. Expert Systems with Applications, 116659.
12. Li, Y., & Pan, Y. (2022). A novel ensemble deep learning model for stock prediction based on stock prices and news. International Journal of Data Science and Analytics, 1-11.
13. Yun, K. K., Yoon, S. W., & Won, D. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. Expert Systems with Applications, 186, 115716.
14. Shah, A., Gor, M., Sagar, M., & Shah, M. (2022). A stock market trading framework based on deep learning architectures. Multimedia Tools and Applications, 81(10), 14153-14171.
15. Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. Expert Systems with Applications, 169, 114332.
16. Yıldırım, M., Okay, F. Y., & Özdemir, S. (2021). Big data analytics for default prediction using graph theory. Expert Systems with Applications, 176, 114840.
17. Yun, K. K., Yoon, S. W., & Won, D. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. Expert Systems with Applications, 186, 115716.
18. Jankova, Z., Jana, D. K., & Dostal, P. (2021). Investment decision support based on interval type-2 fuzzy expert system. Engineering Economics, 32(2), 118-129.
19. Poornappriya, T. S., and R. Gopinath. "Enhancing Breast Cancer Diagnosis: A Neural Network-Based Clustering Approach For Segmentation." Webology (ISSN: 1735-188X) 18.5 (2021).
20. Poornappriya, T. S., and R. Gopinath. "Rice plant disease identification using artificial intelligence approaches." International Journal of Electrical Engineering and Technology 11.10 (2022): 392-402.
21. R Gopinath, TS Poornappriya. (2020). An Analysis of Human Resource Development Practices in Small Scale Startups. International Journal of Advanced Research in Engineering and Technology (IJARET), 11(11), 2475-2483.
22. M Durairaj T S Poornappriya. (2019). High relevancy low redundancy vague set based feature selection method for telecom dataset. Journal of Intelligent & Fuzzy Systems, 37(5), 6743-6760