ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2022.0394

DEEP LEARNING METHODS FOR THE ACCURATE MODELING AND FORECASTING OF THE INDIAN STOCK MARKET

Godfrey Joseph Saqware[1] and B. Ismail[2]
[1]Department of Statistics, University of Dar Es Salaam, Tanzania
[2]Department of Statistics, Yenepoya University, India

Abstract

The stock markets are among the most volatile market worldwide. The future of these markets is daily affected by political instability and different enacted economic and government policies. Thus, the prediction and forecast of these markets are very important. The Bombay Stock Exchange (BSE) is the oldest stock market in Asia and India. This paper applied deep learning methods to predict the five companies closing prices under BSE. The selected companies based on market capitalization were Reliance Industries Ltd (RELI), TATA Consultancy Services (TCS), HDFC Bank Ltd (HDBK), Infosys Ltd (INFY), and ICICI Bank Ltd (ICBK). Based on Root Mean Square Error (RMSE), the traditional Bidirectional Long Short-Term Model (Bi-LSTM) model predicted well the HDBK closing prices. The Convolution Neural Networks (CNN) outperformed other models in predicting the ICBK, RELI, and INFY. The proposed Hybrid CNN-LSTM model with Bayesian hyperparameter tuning outperformed the CNN and Bi-LSTM models in predicting the TCS close price. Moreover, the hybrid model ranked second in predicting closing prices in all the selected companies. The next 100 days forecast shows high price volatility in the selected companies. In the closing prices forecasts, the hybrid CNN-LSTM model with Bayesian hyperparameter tuning has captured well the trend of the historical data. Additionally, Traders and financial analysts may easily understand the future market trend using the methods. Therefore, the powerful computer and more complex hybrid model may be applied to bring the best performance in terms of accuracy.

Keywords:

Bayesian hyperparameter tuning, Bi-LSTM, Bombay Stock Exchange, CNN, and CNN-LSTM

1. INTRODUCTION

Financial market forecast, such as the stock market, exchange rate, and share value, is a complex topic of study nowadays. The factors such as physical, physiological, rational, irrational, investor sentiment, and market rumors all play great roles in stock markets [1]. Deep learning models have an anonymous application in predicting the time series data. It has a cross-cutting application in many research areas. Recently, deep learning models have become powerful in predicting and forecasting disease outbreaks. The comparative study used deep learning methods to forecast the confirmed and death cases of COVID-19 [2]-[3]. Meteorological variables such as daily temperature have recently drawn considerable attention from researchers to address the limitations of traditional forecasting models. The artificial neural network (ANN), recurrent neural network (RNN), and LSTM are trained and tested by integrating the genetic algorithm (GA). The genetic algorithm (GA) was used to optimize the deep learning network structure of hyperparameter optimization and finally select the best architecture for the network. The findings show that the hybrid model of the LSTM network and GA outperforms other models for 15 days of forecasting in summer [4]. In agriculture, the models such as SARIMA and Holt-Winter's Seasonal method, and LSTM neural network were compared to forecast the Arecanut prices. LSTM neural network model best forecasted the data[5]. Aside from using deep learning models in other fields, stock market data prediction and analysis are critical in today's economy. Financial forecasting, often known as stock market forecasting, is one of the most popular research disciplines. The study used particle swarm optimization to update the parameters of deep learning models and compare them to historical data from the BSE. The results reveal that the BSE patterns are recognized by deep neural network models [6]. The model that uses an ANN optimized by the gray wolf optimization (GWO) technique achieved better prediction accuracy. Furthermore, the result showed that the proposed model outperforms the ANN model in predicting BSE data [7].

The paper is organized as follows. In section two, several related pieces of literature on the stock price prediction were surveyed and reviewed. Section three describes the research materials and methods. In section four, we present the results and discussion for the proposed novel model, which compares its performance with that of traditional deep learning models. Finally, section five discussed concluding remarks and future work.

2. REVIEW OF LITERATURE

Deep Learning (DL) models have recently emerged in the industry, with results far outperform their classic machine learning (ML) counterparts. Despite the increased effort to develop financial time series forecasting models, few review papers are devoted specifically to DL in finance [8]. We observed that recent models combining LSTM with other techniques, such as DNN, have gotten a lot of press. The results of reinforcement learning and different deep learning algorithms were good. We find that the adoption of deep-learning-based financial modeling tools has exploded in recent years. [9]. As an intelligent and optimal stock market prediction model, a hybridization of Adaline Neural Network (ANN) and modified Particle Swarm Optimization (PSO) was constructed. PSO is used to optimize and update the weights of the Adaline representation for the open price of the BSE. Interval measurements, CMS-PSO, and Bayesian-ANN representations compare the proposed model's prediction ability. The findings show that the proposed model outperformed other models[10]. The discussion involving using the Bayesian hyperparameter tuning from the hybrid deep learning models is still important for better forecasting performance[11]. Deep learning methods with hyperparameters tuning affect the performance of the algorithms in predicting the one-day electricity consumption. The best results with the lowest error rate were compared to previous studies on electricity consumption [12]. Deep learning techniques based on neural networks have several hidden layers, making deep neural network computation difficult and complex. The improvement of the deep learning performance requires hyperparameter adjustment. The findings of the experiments reveal that dynamic tweaking of the deep long short-term memory (DLSTM) hyperparameters outperform the original static tuning method [13].

When two models are combined, some collaboration is established between them, which could improve the model's analytical ability. The results show that combining CNN with LSTM, CEEMD, or EMD can improve prediction accuracy and surpass other methods [14].

This paper combined the CNN and LSTM models with the Bayesian hyperparameter tuning. The proposed hybrid model with the Bayesian hyperparameter tuning will be used to predict and forecast the five companies under BSE based on market capitalization.

3. MATERIAL AND METHODS

3.1 DATA SPECIFICATION

This paper considers the stock data of the five companies under BSE from https://investing.com. The analysis solely looked at closing historical stock prices for the five companies chosen based on market capitalization from October 1, 2010, to April 16, 2022. Seventy percent of the data was used for training and thirty percent for testing. The Table.1 below shows the data description for the five selected companies.

Table.1. Data description of the selected companies

Company	**Market Capitalization (₹Crore)**	**Total observations**
Reliance Industries Ltd (RELI)	1726714.05	3052
TATA Consultancy Services (TCS)	1339688.48	3052
HDFC Bank Ltd (HDBK)	812338.57	3052
Infosys Ltd (INFY)	735611.35	3052
ICICI Bank Ltd (ICBK)	529739.59	3052

3.2 METHODS

The deep learning models have been widely used in image analysis, speech recognition, and Natural Language Processing (NLP). The most used deep learning algorithms are convolutional neural networks (CNN), RNN, LSTM, and self-encoder networks. Thus, two models, CNN and LSTM, are combined to create a Hybrid CNN-LSTM with the integration of the Bayesian hyperparameter tuning to observe the effectiveness. Below are the three models for predicting and forecasting the five major selected companies under BSE.

3.2.1 CNN Model:

The CNN model has tremendous application in Image recognition, Natural Language Processing (NLP), and classification activities[15]. Convolution Neural Networks have been widely regarded as the best neural network architectures for time series classification and forecasting in recent years, and they can perform comparably or even better than other neural network architectures [16]–[19]. The CNN operates under three primary layers: Convolutional Layer, Pooling Layer, and Fully Connected Layer. This enables an understanding of how the Machine Learning Model perceives an image and transforms the idea into time series forecasting. A convolution layer plus a max-pooling layer makes up the CNN, learning global knowledge from 1-D data. The CNN has been shown to have extraction and rearrangement advantages. The convolution layer comprises filter numbers, filter length, and a small collection of neurons. Moreover, the convolution layers link each neuron with its neighbour and compute the dot product of the filter and the input dataset (1-D or 2-D metric). The max polling layer reduces the number of parameters and redundant features. Furthermore, the convergence of neural networks is controlled by a pooling layer. The pooling layer selects the largest value of the field covered by the pooling filter. The weights are sent to the highest-valued filter via the pooling layer [20]. Finally, the flatten layer flattens pooled feature and maps into a column like an image [21], [22]. The Fig.1 shows the 1D CNN model architecture.

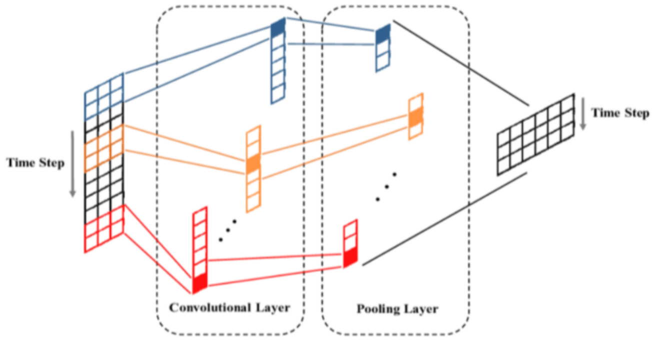

Fig.1. CNN Model Architecture

3.3 LSTM MODEL

The RNN model creates the LSTM model. Unlike standard RNN models, LSTM contains a long-term memory function that prevents the gradient from exploding and vanishing. It deforms the RNN structure by adding memory cells to the hidden layer to regulate the sequential data memory information (Hochreiter & Schmidhuber, 1997). The LSTM model has enormous application in both the classification and forecasting tasks [24]–[26]. The LSTM model sends data to different cells via programmable gates: forget, input, and output. The forget gate is in charge of the memory cell, deciding how much information should be saved or rejected by the system. The input gate controls the historical information and present stimulation and affects how much current input is stored in the cell state. Finally, the LSTM output gate determines which output to provide based on the current internal cell state. The cell state runs throughout the network and, with the use of gates, can add or delete information. The sigmoid function assists in the generation of the numbers 0 and 1. This specifies how much information each component allows or discards. The tanh layer adds a new vector to the state graph [27]. The Fig.2 shows the LSTM model architecture.

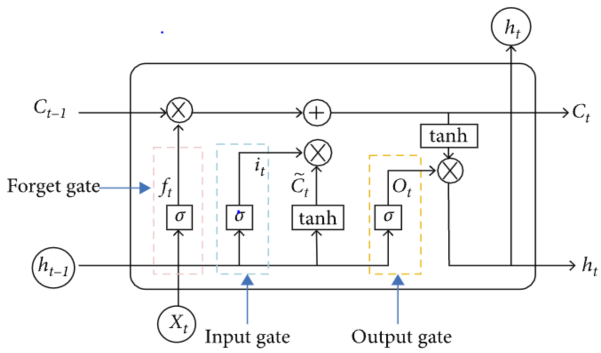

Fig.II: LSTM Model Architecture

The mathematical model derived from the above Fig.is as follows: -

$$f_t = \sigma\left(W_t.[h_{t-1} - x_t] + b_f\right) \quad (1)$$

$$i_t = \sigma\left(W_i.[h_{t-1} - x_t] + b_i\right) \quad (2)$$

$$C_t = \tanh\left(W_c.[h_{t-1} - x_t] + b_c\right) \quad (3)$$

$$C_t = f_t.C_t + i_t.C_t \quad (4)$$

$$O_t = \sigma\left(W_o.[h_{t-1} - x_t] + b_o\right) \quad (5)$$

$$h_t = O_t.tahnC_t \quad (6)$$

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \quad (7)$$

$$\mathrm{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (8)$$

Where the input weights are W_f, W_i, W_c and are input weights, the biased weight are b_f, b_i, b_c and b_o, t and t-1 represents the current and previous time state respectively, b_f represents input; h_t represents output and C_t is cell states W_o.

The bi-LSTM model is a modified augmentation of the LSTM model. For situations involving sequence classification, Bi-LSTM increases model execution. In the training phase of the sequence of inputs, Bi-LSTM employs two LSTMs rather than one. The Bi-LSTM deep learning model can simultaneously access all prior and expected future information [28]. The model consists of the input, forward, backward, activation, and output layers. The Fig.3 shows the Bi-LSTM model architecture.

3.3.1 Hybrid CNN-LSTM Model:

The hybrid CNN-LSTM model was constructed by combining CNN with LSTM to improve forecasting accuracy in the selected companies. This paper incorporates the Bayesian hyperparameter tuning to obtain the best-fit hybrid model for the time series forecasting. The proposed hybrid CNN-LSTM model with Bayesian hyperparameter tuning will predict and forecast the selected companies' closing prices for 100 days.

3.3.2 Bayesian hyperparameter tuning:

Deep learning has achieved impressive results on many problems. However, the models require a high degree of expertise or experience in tuning the hyperparameters to avoid biases from the manual tuning process. Moreover, it is not practical to try out many hyperparameter configurations in deep learning as in other machine learning scenarios. Evaluating each hyperparameter configuration in deep learning would mean training a deep neural network, which usually takes a long time [29].

Consider a hyperparameter tuning task wherein our objective is to maximize validation set accuracy as:

$$w^* = arg\ max_{w \in W} \ f(w) \quad (9)$$

where $W \subseteq R^D$, $z = f(w) + \varepsilon$, $\varepsilon \sim N\left(0, \sigma^2\right)$, and $f(w)$ is the model's performance on the validation dataset set of hyperparameters w. Let the search bound for different hyperparameters be $[l, \mu]$ – l and μ are D dimensional vectors denote the lower and the upper bounds, respectively. The objective remains to optimize hyperparameters using the whole training data. The process starts by identifying the optimal hyperparameters on a small subset of the training data. We use a standard Bayesian optimization algorithm to quickly identify optimal configurations on subsets of data, given as:

$$w^{*s,b} = arg\ max_{w \in W} \ fs, b(w) \quad (10)$$

Here $fs, b(w)$ denotes the performance of the model trained on a subset b of the validation dataset for a hyperparameter configuration *w*. We use *s* to denote everything related to a subset of data. Repeating Bayesian optimization and averaging the optimal hyperparameters yields a robust estimate of the hyperparameters. The performance of a model trained on small data may exhibit spurious spikes that peak either at very high or low complexity regions. Through averaging, we smooth out the spurious behaviours.

We denote this best hyperparameter from the smaller subset as $\overline{w}^{*s}$. While tuning hyperparameters on the whole dataset, we note that for certain hyperparameters directly controlling the model complexity, the generalization performance would be monotonically changing in the bound $[l, \overline{w}^{*s}]$. If a particular hyperparameter increases the model complexity, then the performance also increases for a larger dataset wherein it decreases otherwise. The Bayesian hyperparameter involves Gaussian process and acquisition functions in its operations.

- **Gaussian process (GP):**

The GP is the technique developed based on the stochastic process and Bayesian learning theory. The mean and covariance function for GP is $\mu(w)$ and $K(w, w')$, respectively [30]. The sample function of the GP is given in Eq.(11).

$$f(w) \sim GP(\mu(w), K(w, w')) \quad (11)$$

Assume the mean function of the GP process is $m(w) = 0$, then the exponential square kernel fully defines GP.

$$k(w_i, w_j) = \exp\left(-\frac{1}{2\phi}\left\|w_i - w_j\right\|^2\right) \quad (12)$$

The smoothness assumption of the function in Eq.(12) is controlled by the length scale parameter ϕ. The w_i and w_j are i^{th} and j^{th} samples, respectively. If w_i and w_j are strongly correlated, then $K(w_i, w_j) \rightarrow 1$, otherwise $K(w_i, w_j) \rightarrow 0$.

Thus, the posterior distribution is obtained through two main steps:

Step 1. Assume a sample of t observations as a training set $D_{1,t} = \{w_n, f_n\}_{n=1}^{t}$, and $f_n = f(w_n)$. If f are drawn from multivariate normal distributions $f \sim N(0, K)$, where K is given by:

$$K = \begin{bmatrix} k(w_1,w_1) & k(w_1,w_2) & . & . & . & k(w_1,w_t) \\ k(w_2,w_1) & k(w_2,w_2) & & & & k(w_2,w_t) \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ k(w_t,w_1) & k(w_t,w_2) & . & . & . & k(w_t,w_t) \end{bmatrix} \quad (13)$$

Step 2. Base on the function *f*, compute $t+1 = f(w_{t+1})$ the new point w_{t+1}, based on the GP, then $D_{1,t}$, and the new value f_{t+1} follows $t+1$ dimensional normal distribution.

$$\begin{pmatrix} f_{1:t} \\ f_{t+1} \end{pmatrix} \sim N\left(0, \begin{bmatrix} K & k \\ k^T & k(w_{t+1}, w_{t+1}) \end{bmatrix}\right) \quad (14)$$

where,

$f_{1:t} = (f_1, f_2, f_3, ..., f_t)^T$ and

$k = [k(w_{t+1}, w_1), k(w_{t+1}, w_2), k(w_{t+1}, w_3), ..., k(w_{t+1}, w_t)]$

If $f_{t+1} \sim N(\mu_{t+1}, \sigma_{t+1}^2)$, then using the property of the Gaussian distribution, μ_{t+1} and $, \sigma_{t+1}^2$ can be given as:

$$\begin{aligned} \mu_{t+1}(w_{t+1}) &= k^T K^{-1} f_{1:t} \\ \sigma_{t+1}^2(w_{t+1}) &= k^T K^{-1} k + k(w_{t+1}, w_{t+1}) \end{aligned} \quad (15)$$

- **Acquisition functions:**

Bayesian optimization employs an efficient strategy that employs a surrogate utility function, which is easy to evaluate. This utility function is usually called the acquisition function. The acquisition function helps us to reach the optimum of the underlying function. The acquisition function translates the epistemic measure offered by the GP to seek the next location to evaluate the function[31]. To obtain posterior distribution of the objective function, the Bayesian optimization uses the acquisition function μ to derive the maximum function f. The acquisition function is assumed to correspond to the objective function's f larger value. Thus, maximizing the acquisition function is the same as maximizing the function f.

$$w^+ = \arg\max_{w \in A} \mu(w \mid D) \quad (16)$$

Probability improvement (PI) and Expected improvement (EI) are the most commonly used acquisition functions. Let the symbols $\psi(.)$ and $\vartheta(.)$ represent cumulative and probability distribution functions of the standard normal distributions, and $w^+ = \arg\max_{w_i \in w_{1:t}} f(w_i)$, w^+ represents the position where the function *f* is maximizing at t sample points.

- **Probability Improvement (PI)**

The function PI explores near the current optimal value point to find the most likely prevail over the current value. The search continues until the number of iterations of the algorithm reaches the upper limit. The PI of the function is given in Eq.(5.25).

$$PI(w) = P(f(w) \geq f(w^+)) = \psi\left(\frac{\mu(w) - f(w^+)}{\sigma(w)}\right) \quad (17)$$

The PI in the algorithm considers sampling closer to the optimal solution, thus facing some throwbacks. The parameter ε is introduced into the equation to be solved. The new sampling points replace only if the difference between the next sampling point and the current optimal value is not less than ε. Thus Eq.(17) can be re-written as:

$$PI(w) = P(f(w) \geq f(w^+) + \varepsilon) = \psi\left(\frac{\mu(w) - f(w^+) - \varepsilon}{\sigma(w)}\right) \quad (18)$$

- **Expected Improvement (EI)**

The function EI calculates the improvement that point can achieve when exploring the current optimum value. If the current optimal value point is less than the expected value after the algorithm execution, then the current optimal value may be the optimal local solution. Thus, the algorithm will find the optimal value at the other position of the domain. The degree of improvement (I) is the difference between the function value at the sampling point and the current optimum values. The improvement function is 0 if the sampling point is less than the current optimum point.

$$I(w) = \max\{0, f_{t+1}(w) - f(w^+)\} \quad (19)$$

In the optimization strategy, we minimize the current optimum values f as:

$$\begin{aligned} I(w) &= \arg\max E(I(w)) = \\ &\arg\max E\{0, f_{t+1}(w) - f(w^+)\} \end{aligned} \quad (20)$$

When, $f_{t+1}(w) - f(w^+) \geq 0$, the distribution of f_{t+1} obeys the normal distribution with mean and variance $\mu(w)$ and σ_w^2. Thus, the random variable I has a normal distribution with mean $\mu(w) - f(w^+)$ and variance σ_w^2. The probability density function of I is given as:

$$f(I) = \frac{1}{\sigma(w)\sqrt{2\pi}} \exp\left(-\frac{(\mu(w) - f(w^+))^2}{2\sigma^2(w)}\right), I > 0 \quad (21)$$

Using Eq.(21), the EI can be defined as follows:

$$E(I) = \int_{-\infty}^{\infty} If(I)dI = \int_0^{\infty} I \times \frac{1}{\sigma(w)\sqrt{2\pi}} \exp\left(-\frac{(\mu(w) - f(w^+))^2}{2\sigma^2(w)}\right) dI \quad (22)$$

Which can further be simplified as $E(I) = \sigma(w)[Z\psi(Z) + \vartheta(Z)]$, where $Z = \frac{\mu(w) - f(w^+)}{\sigma(w)}$.

The hybrid CNN-LSTM model with Bayesian hyperparameter tuning involves three main steps. The selection of hyperparameter lists, Bayesian optimizers, deep learning models, and model validation. The input of the sequence data precedes the steps. The model automatically selects parameters based on the pre-set criterion. The process undergoes many iterations until the best final model output is achieved. The system automatically opts the either LSTM or Bi-LSTM, depending on which is better. Fig. 4

shows the hybrid CNN-LSTM model architecture with the Bayesian hyperparameter tuning.

3.4 EVALUATION METRICS

The model performance evaluation will be done based Root-Mean-Square-Error (RMSE) and Pearson Rank correlation(r). The RMSE measures the differences between actual and predicted values[32], [33]. The formula for computing RMSE is as given in Eq.(13) below.

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(y_i - \hat{y}_i)^2} \quad (23)$$

$$r = \frac{\sum_{i=1}^{T}(y_i - m_y)(\hat{y}_i - m_{\hat{y}})}{\sqrt{\sum_{i=1}^{T}(y_i - m_y)^2 * \sum_{i=1}^{T}(\hat{y}_i - m_{\hat{y}})^2}} \quad (24)$$

where T is the total number of observations, y_i is the actual value, $\hat{y}_i$ is the predicted value, m_y is the mean of the observed values, and $m_{\hat{y}}$ is the mean of the predicted values.

4. RESULTS AND DISCUSSION

4.1 ANALYSIS OF FIVE SELECTED COMPANIES

The HDBK, ICBK, RELI, TCS, and INFY have mean values of 946.0568, 307.4428, 892.1589, and 1882.6486 846.0435, respectively. The HDBK, ICBK, RELI, TCS, and INFY have standard deviations of 363.5682, 158.7732, 635.8195, 776.5484, and 326.6331, respectively. Higher standard deviations indicate that historical closing prices for the five chosen companies are more volatile. Furthermore, the historical plots in Fig.5-Fig 9 reveal that the closing prices of these major companies under BSE (S&P-100) have an unpredictably volatile pattern. The plots of the closing prices for the five selected companies under the BSE (S&P-100) are shown in Fig.5-Fig 9.

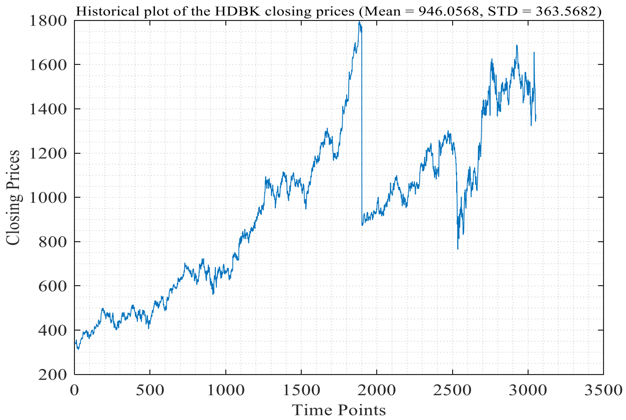

Fig.5. HDBK historical plot of the closing prices

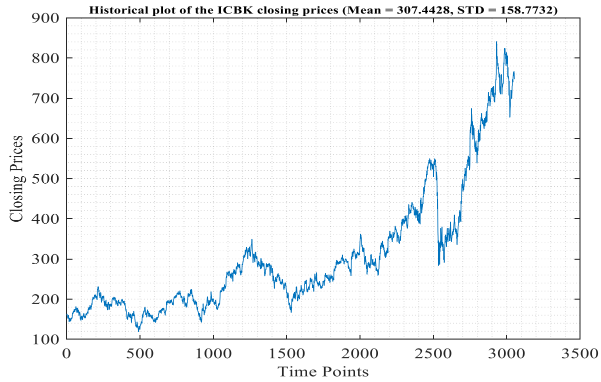

Fig.6. ICBK historical plot of the closing prices

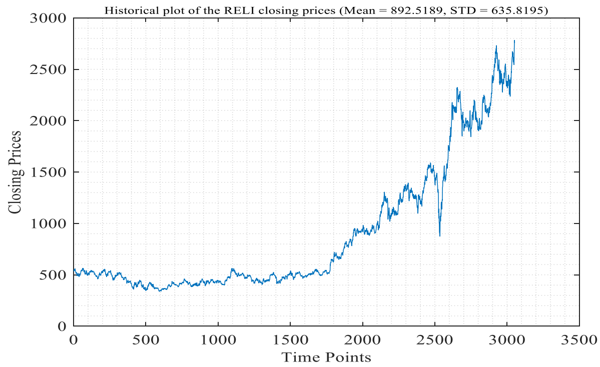

Fig.7. RELI historical plot of the closing prices

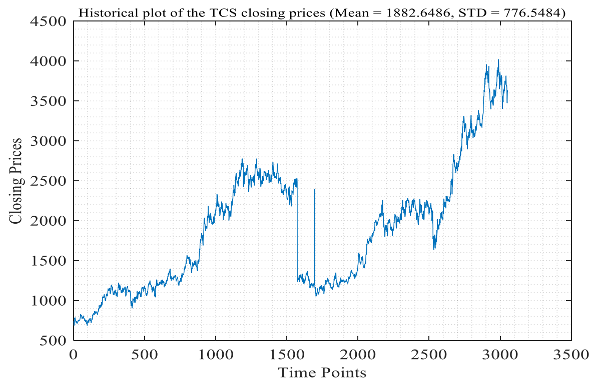

Fig.8. TCS historical plot of the closing prices

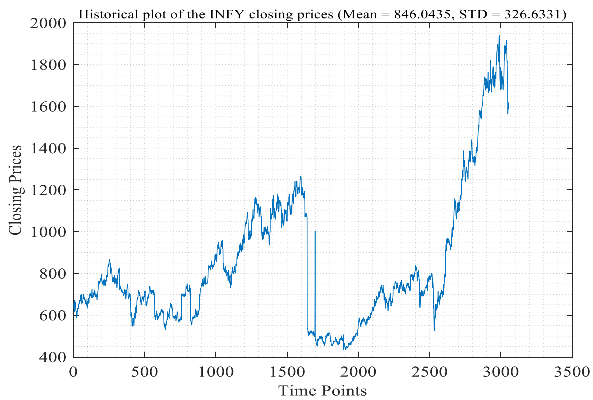

Fig.9. INFY historical plot of the closing prices

4.2 PERFORMANCE ANALYSIS OF THE MODELS

The degree of rank autocorrection between observed and predicted values is extremely high (>95%). This demonstrates that the closing prices and their predicted values for the five companies are highly correlated. The minimum and maximum RMSE are 5 and 92 for testing sets and, 25 and 477 for the training sets. Based on RMSE, the CNN model has performed well in predicting the ICBK, RELI, and INFY. The previous study to estimate future NIFTY index values showed that the CNN-based multivariate forecasting model is the most effective and accurate [34]. For the HDBK company, the Bi-LSTM model performed well in predicting the closing price. Previous studies pointed out that precise parameter adjustment is also required for LSTM and Bi-LSTM models. Thus, when the same parameters are used in both models, the Bi-LSTM model produces a lower RMSE than the LSTM model in predicting individuals and ventures for stock market forecasting[28], [35], [36]. The proposed hybrid CNN-LSTM with hyperparameter tuning performed well in predicting TCS's closing prices. The hybrid CNN-LSTM model with Bayesian hyperparameter tuning ranked second in predicting closing price testing set prediction for all five selected companies. Because the model takes a long time to converge, a high-performance machine with at least a single Graphics Processing Unit (GPU) is required. The previous study was done on the TCS using a deep neural network Conv1D-LSTM that combines layers of two different techniques CNN and LSTM. The performance of the hybrid Conv1D-LSTM model was found to be better than CNN and LSTM for stock price prediction [37]. Furthermore, the hybrid CNN-LSTM model has provided reliable stock price forecasting with the highest prediction accuracy in the different studies on financial time series data [38], [39]. These prediction sets confirm that no single model fits the prediction of every stock market.

4.3 THE FORECASTING COMPARISON OF THE BEST INDIVIDUAL AND HYBRID MODELS

The forecast comparison between the best single model and the proposed hybrid model is made. In Fig.10(a) and Fig.10(b), the forecasting comparison for the HDBK for both the Bi-LSTM and hybrid models looks different. The hybrid model shows a decrease and a price increase, while the best model Bi-LSTM, shows an increase in price in the 100 days forecasting closing prices. In Fig.11(a) and Fig.11(b), the ICBK 100 days forecasts for CNN and the hybrid CNN-LSTM models with hyperparameter tuning don't look the same. The hybrid model forecast looks much better than the CNN model. In Fig.12(a) and Fig.12(b), the 100 days forecast of the RELI for the hybrid CNN-LSTM model looks much better compared to the CNN model. In Fig.13(a) and Fig.13(b), the 100 days forecasts for the TCS hybrid CNN-LSTM look much better in tracking the previous historical data trend than the CNN model. In this case, the hybrid model performs better in predicting the training set. Fig.14(a) and Fig.14(b) 100 days forecasts for both the hybrid and the CNN model don't show convincing results in predicting the INFY closing prices. Based on the evaluation of the 100 days forecast of the selected five companies, the hybrid CNN-LSTM model with the Bayesian hyperparameter tuning has shown at least better results. The hybrid model failed to produce a better result only for INFY closing prices. The Bayesian hyperparameter tuning plays a more significant role in selecting the best parameters of the models, thus helping to keep the model forecasting ability consistent.

Table.2. Model results for the five companies

Variable	Model	Training data		Test data		Test data RMSE Rank
		Rank Correlation	RMSE	Rank Correlation	RMSE	
HDBK	CNN	0.99729	19.2107	0.93125	56.8316	3
	Bi-LSTM	0.9976	23.8864	0.98085	32.3485	1
	Hybrid	0.98988	35.8421	0.96813	51.6033	2
ICBK	CNN	0.99736	5.359	0.99135	23.9731	1
	Bi-LSTM	0.9967	6.1169	0.99225	95.0516	3
	Hybrid	0.9957	33.7858	0.97412	35.7656	2
RELI	CNN	0.99527	13.4816	0.98451	136.9663	1
	Bi-LSTM	0.99295	18.9966	0.97827	476.3477	3
	Hybrid	0.98937	16.0743	0.96464	342.4285	2
TCS	CNN	0.9958	50.0106	0.99304	238.1239	2
	Bi-LSTM	0.99468	51.625	0.99387	533.5963	3
	Hybrid	0.99321	50.2231	0.9909	89.8019	1
INFY	CNN	0.99435	19.2371	0.9898	35.7305	1
	Bi-LSTM	0.99226	22.6395	0.99389	243.1368	3
	Hybrid	0.9925	22.168	0.99227	66.8471	2

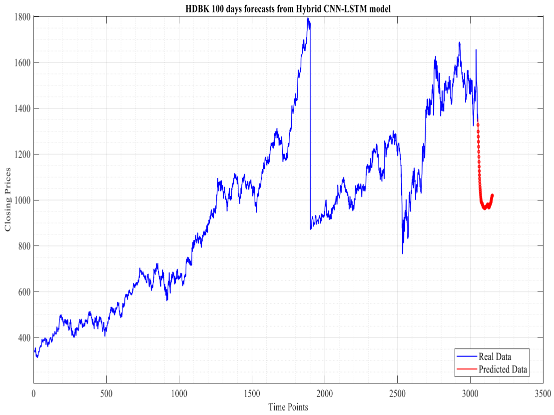

Fig.10(a). HDBK Hybrid CNN-LSTM model forecasts

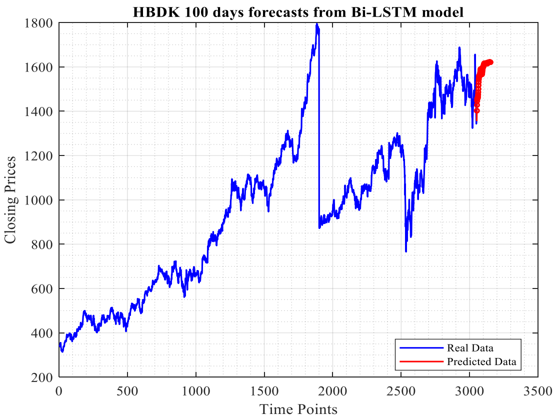

Fig.10(b). HDBK Bi-LSTM model forecasts

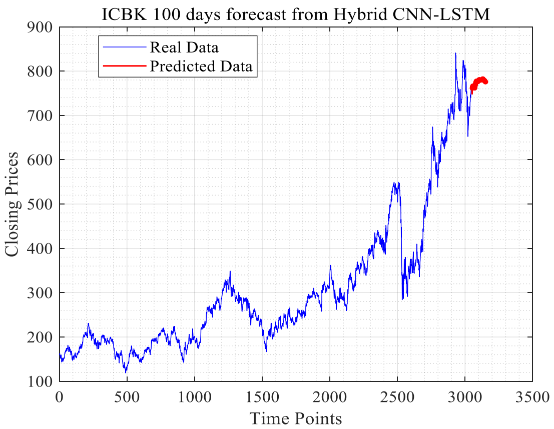

Fig.11(a). ICBK Hybrid CNN-LSTM forecasts

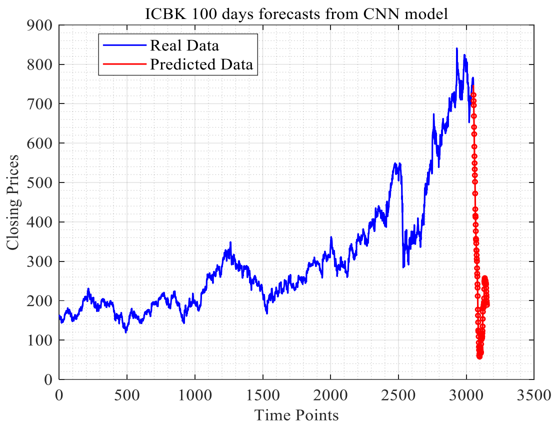

Fig.11(b). ICBK CNN model forecasts

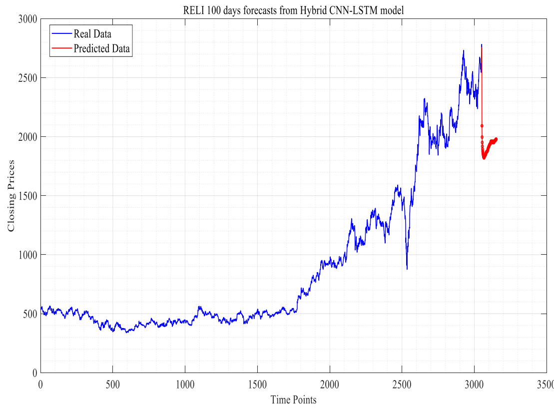

Fig.12(a). RELI Hybrid CNN-LSTM model forecasts

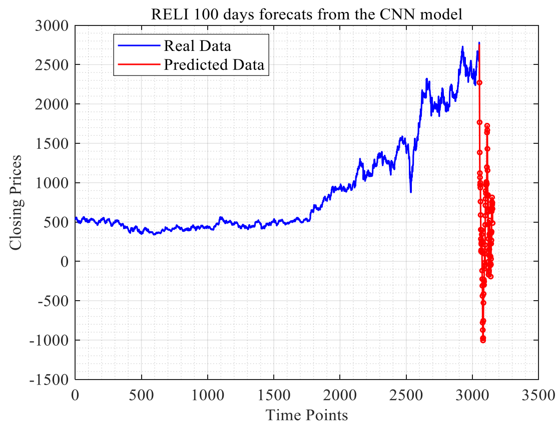

Fig.12(b). RELI CNN model forecasts

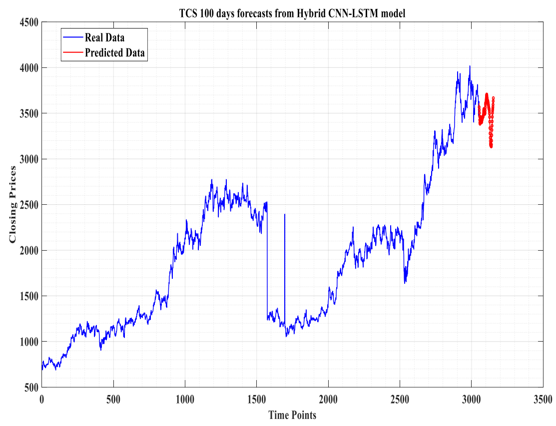

Fig.13(a). TCS Hybrid CNN-LSTM model forecasts

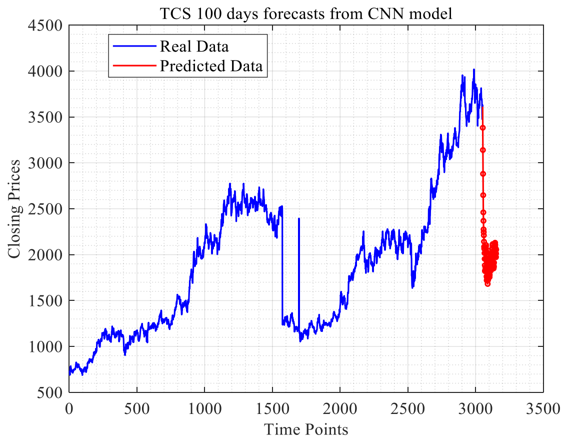

Fig.13(b). TCS CNN model forecasts

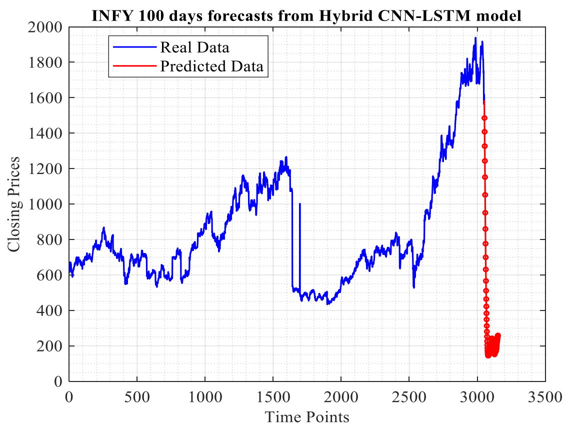

Fig.14(a). INFY Hybrid CNN-LSTM model forecasts

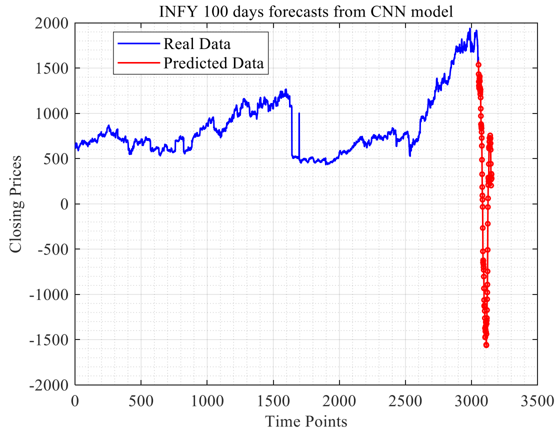

Fig.14(b). INFY CNN model forecasts

5. CONCLUSION

It will always be tough to predict the stock market. Higher volatility was seen in the 100-day forecast of selected BSE companies based on market capitalization. The particular stock market influences the models' performance in consideration. Each of the five BSE-selected companies has demonstrated that it is forecasted using a different model. The CNN model looked to perform better in the validation sets for the three companies, but future forecasts were dismal. The hybrid model appears the second in each experimental case. Despite the complexity of the hybrid model with Bayesian hyperparameter tuning, it has more consistent forecast results than individual models. Furthermore, the hybrid CNN-LSTM model with hyperparameter tuning may be used to forecast prices in different stock markets. Bayesian hyperparameter tuning necessitates a high-speed computer to get the fastest output convergence and improved outcomes.

REFERENCES

[1] A. Ghosh, S. Bose, G. Maji, N. Debnath and S. Sen, “Stock Price Prediction using LSTM on Indian Share Market”, *Proceedings of International Conference on Neural Computing*, Vol. 63, pp. 101-110, 2019.

[2] H. Abbasimehr, R. Paki and A. Bahrini, “A Novel Approach based on Combining Deep Learning Models with Statistical Methods for COVID-19 Time Series Forecasting”, *Neural Computing and Applications*, Vol. 45, pp. 1-15, 2021.

[3] N.F. Omran, S.F. Abd-El Ghany, H. Saleh, A.A. Ali, A. Gumaei and M. Al-Rakhami, “Applying Deep Learning Methods on Time-Series Data for Forecasting Covid-19 in Egypt, Kuwait, and Saudi Arabia”, *Complexity*, Vol. 2021, pp. 1-12, 2021.

[4] T. Thi Kieu Tran, T. Lee, J.Y. Shin, J.S. Kim and M. Kamruzzaman, "Deep Learning-based Maximum Temperature Forecasting Assisted with Meta-Learning for Hyperparameter Optimization", *Atmosphere*, Vol. 11, No. 5, pp. 487-498, 2020.

[5] K.M. Sabu and T.M. Kumar, "Predictive Analytics in Agriculture: Forecasting prices of Arecanuts in Kerala", *Procedia Computer Science*, Vol. 171, pp. 699-708, 2020.

[6] A. Perwej, K.P. Yadav, V. Sood and Y. Perwej, "An Evolutionary Approach to Bombay Stock Exchange Prediction with Deep Learning Technique", *IOSR Journal on Business Management*, Vol. 20, No. 12, pp. 63-79, 2018.

[7] S. Sahoo and M.N. Mohanty, "Stock Market Price Prediction Employing Artificial Neural Network Optimized by Gray Wolf Optimization", *New Paradigm in Decision Science and Management*, pp. 77-87, 2020.

[8] O.B. Sezer, M.U. Gudelek and A.M. Ozbayoglu, "Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005-2019", *Applied Soft Computing*, Vol. 90, pp. 106181-106189, 2020.

[9] Z. Hu, Y. Zhao and M. Khushi, "A Survey of Forex and Stock Price Prediction using Deep Learning", *Applied System Innovation*, Vol. 4, No. 1, pp. 1-9, 2021.

[10] M.R. Senapati, S. Das and S. Mishra, "A Novel Model for Stock Price Prediction using Hybrid Neural Network", *Journal of the Institution of Engineers (India): Series B*, Vol. 99, No. 6, pp. 555-563, 2018.

[11] W. Kong, "Effect of Automatic Hyperparameter Tuning for Residential Load Forecasting via Deep Learning", *Proceedings of Australasian Universities Conference on Power Engineering*, pp. 1-6, 2017.

[12] S. Ayvaz and O. Arslan, "Forecasting Electricity Consumption using Deep Learning Methods with Hyperparameter Tuning", *Proceedings of International Conference on Signal Processing and Communications Applications*, pp. 1-4, 2020.

[13] N. Bakhashwain and A. Sagheer, "Online Tuning of Hyperparameters in Deep LSTM for Time Series Applications", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 1, pp. 212-220, 2021.

[14] H. Rezaei, H. Faaljou and G. Mansourfar, "Stock Price Prediction using Deep Learning and Frequency Decomposition", *Expert Systems with Applications*, Vol. 169, pp. 1-18, 2021.

[15] J. Zhao, X. Mao and L. Chen, "Speech Emotion Recognition using Deep 1D & 2D CNN LSTM Networks", *Biomedical Signal Processing and Control*, Vol. 47, pp. 312-323, 2019.

[16] N. Hatami, Y. Gavet and J. Debayle, "Classification of Time-Series Images using Deep Convolutional Neural Networks", *Proceedings of International Conference on Machine Vision*, pp. 1-5, 2018.

[17] A. Le Guennec, S. Malinowski and R. Tavenard, "Data Augmentation for Time Series Classification using Convolutional Neural Networks", *Proceedings of International Conference on Signal Processing*, pp. 1-9, 2016.

[18] S.H.I. Xingjian, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong and W. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting", *Advances in Neural Information Processing Systems*, pp. 802-810, 2015.

[19] B. Zhao, H. Lu, S. Chen, J. Liu and D. Wu, "Convolutional Neural Networks for Time Series Classification", *Journal of Systems Engineering and Electronics*, Vol. 28, No. 1, pp. 162-169, 2017.

[20] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural Language Processing (Almost) from Scratch", *Journal of Machine Learning Research*, Vol. 12, No. 1, pp. 2493-2537, 2011.

[21] A. Gulli and S. Pal, "*Deep Learning with Keras*", Packt Publishing Ltd, 2017.

[22] J. Jin, A. Dundar and E. Culurciello, "Flattened Convolutional Neural Networks for Feedforward Acceleration", *Proceedings of International Conference on Signal Processing*, pp. 1-12, 2014.

[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, 1997.

[24] S. Bodapati, H. Bandarupally, and M. Trupthi, "COVID-19 Time Series Forecasting of Daily Cases, Deaths Caused and Recovered Cases using Long Short Term Memory Networks", *Proceedings of International Conference on Computing Communication and Automation*, pp. 525-530, 2020.

[25] F. Karim, S. Majumdar, H. Darabi and S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification", *IEEE Access*, Vol. 6, pp. 1662-1669, 2017.

[26] X. Song, "Time-Series Well Performance Prediction based on Long Short-Term Memory (LSTM) Neural Network Model", *Journal of Petroleum Science and Engineering*, Vol. 186, pp. 106682-106689, 2020.

[27] A. Felix and Fred Cummins, "Learning to Forget: Continual Prediction with LSTM", *Neural Computation*, Vol. 12, No. 10, pp. 2451-2471, 2000.

[28] M.A.I. Sunny, M.M.S. Maswood and A.G. Alharbi, "Deep Learning-Based Stock Price Prediction using LSTM and Bi-Directional LSTM Model", *Proceedings of International Conference on Novel Intelligent and Leading Emerging Sciences*, pp. 87-92, 2020.

[29] J. Wang, J. Xu and X. Wang, "Combination of Hyperband and Bayesian Optimization for Hyperparameter Optimization in Deep Learning", *Proceedings of International Conference on Machine Learning*, pp. 1-13, 2018.

[30] C.K. Williams and C.E. Rasmussen, "*Gaussian Processes for Machine Learning*", MIT Press, 2006.

[31] J. Snoek, "Scalable Bayesian Optimization using Deep Neural Networks", *Proceedings of International Conference on Machine Learning*, pp. 2171-2180, 2015.

[32] T. Chai and R.R. Draxler, "Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)? Arguments against Avoiding RMSE in the Literature", *Geoscientific Model Development*, Vol. 7, No. 3, pp. 1247-1250, 2014.

[33] C.J. Willmott and K. Matsuura, "Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance", *Climate Research*, Vol. 30, No. 1, pp. 79-82, 2005.

[34] S. Mehtab and J. Sen, "Stock Price Prediction using Convolutional Neural Networks on a Multivariate

Timeseries", *Proceedings of International Conference on Neural Networks*, pp. 1-13, 2020.

[35] K.A. Althelaya, E.S.M. El-Alfy and S. Mohammed, "Evaluation of Bidirectional LSTM for Short-and Long-Term Stock Market Prediction", *Proceedings of International Conference on Information and Communication Systems*, pp. 151-156, 2018.

[36] R. Chandra, S. Goyal and R. Gupta, "Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction", *IEEE Access*, Vol. 9, pp. 83105-83123, 2021.

[37] S. Jain, R. Gupta and A.A. Moghe, "Stock Price Prediction on Daily Stock Data using Deep Neural Networks", *Proceedings of International Conference on Advanced Computation and Telecommunication*, pp. 1-13, 2018.

[38] A. Kelotra and P. Pandey, "Stock Market Prediction using Optimized Deep-Convlstm Model", *Big Data*, Vol. 8, No. 1, pp. 5-24, 2020.

[39] W. Lu, J. Li, Y. Li, A. Sun and J. Wang, "A CNN-LSTM-Based Model to Forecast Stock Prices", *Complexity*, Vol. 2020, pp. 1-14, 2020.