Deep Learning-based Integrated Stacked Model for the Stock Market Prediction

Samit Bhanja, Abhishek Das

Abstract: Recently, the stock market prediction has become one of the essential application areas of time-series forecasting research. The successful prediction of the stock market can be better guided to the investors to maximize their profit and to minimize the risk of investment. The stock market data are very much complex, non-linear and dynamic. Due to this reason, still, it is a challenging task. In recent time, deep learning method has become one of the most popular machine learning methods for time-series forecasting due to their temporal feature extraction capabilities. In this paper, we have proposed a novel Deep Learning-based Integrated Stacked Model (DISM) that integrates both the 1D Convolution neural network and LSTM recurrent neural network to find the spatial and temporal features from the stock market data. Our proposed DISM is applied to forecast the stock market. Here, we have also compared our proposed DISM with the single structured stacked LSTM, and 1D Convolution neural network models, and some other statistical models. We have observed that our proposed DISM produces better results in terms of accuracy and stability.

Keywords : Convolutional Neural Network, Deep Learning, LSTM, Stock Market, Time-series Forecasting.

I. INTRODUCTION

The stock market prediction research always is a prime attraction to the investors as well as the financial institutions. The successful prediction of the stock market is always helping them to make the early decision to buy or sell the shares of the stock market. This early decision helps them to maximize their profit from their investment.

These financial times series data are more complicated compared to the other time-series data because, these data are very much dynamic, and there is a long-term dependency between them. The prediction of these highly fluctuating, and irregular data is a very challenging task. Recently, so many analytical models that have been proposed, viz. threshold autoregressive model [1], Autoregressive Conditional Heteroscedastic (ARCH) model[2], the Autoregressive Integrated Moving Average (ARIMA) model [3], among others. But these analytical models are not that efficient for non-linear time-series analysis.

In the last two decades, several Artificial Neural Network (ANN) models have been proposed, such as Multilayer Perceptron (MLP) neural network [4], Backpropagation neural network model [5], Recurrent Neural Network (RNN) model [6], and many more. The RNN models have offered serious challenges to the statistical methods for the time-series forecasting problems after they provide better prediction accuracy [7]–[9].

In recent times, ANNs get much more attention to the researchers as well as practitioners of various fields. The turning point of getting this attention, when the ImageNet challenge in 2012 was won by the ANN model based on deep convolution network trained on GPUs [10].

In the last few years, there are so many RNN based models [11]–[14] which have been proposed for the time-series forecasting. But the major weakness of the RNN is that it suffers from the vanishing gradient problem and due to this problem RNN is unable to learn the long term dependencies. These difficulties were successfully overcome by the Long Short-Term Memory Networks (LSTMs) [15]. The recent studies [16] show that these shallow LSTMs neural networks are not so efficient to represent the complex features of the highly non-linear and long interval time-series data.

From the existing studies [17]–[20] it has been proved that deep LSTM neural network models with the stacking of multiple layers are more efficient compared to the shallow neural network models for the long interval and non-linear time-series dataset. But these deep LSTM models are not so much efficient to extract the complex temporal and spatial features from multi-variable time-series data.

Recently, the convolution neural network (CNN) is one of the most successful neural network architecture in deep learning models. This CNN achieved remarkable success in classification problems, such as image recognition [10]. From the studies [21], [22], we observed that 1D CNN (Conv1D) based models can also be used for the regression type problems and these models are also produce very good results for the time-series forecasting. A single Conv1D layer can able to capture certain temporal features within an input window and stacked Conv1D layers just work as a series of filters to aggregate the simple temporal features into a more complex shape to handle the complex multi-variable times series dataset [23]. But [22], [24] this stacked Conv1D layers based model is not so much efficient for the forecasting of the time-series data compared to the stacked LSTM models and is also not able to determine the long term dependencies of the time-series data.

In this study, our aim is to develop a new deep learning model that quite capable of extracting the temporal features to identify the patterns of the stock price movement form the historical multi-variable financial time-series (stock market) dataset. From this motivation, we have proposed a Deep Learning-based stacked integrated model (DISM) for forecasting the stock market.

Revised Manuscript Received on October 25, 2019.

***** Correspondence Author

Samit Bhanja, Computer Science, Government General Degree College, Singur, India. Email: samitbhanja@gmail.com

Abhishek Das*, Computer Science & Engineering, Aliah University, Newtown, Kolkata, India. Email: adas@aliah.ac.in

The rest of the paper is organized as follows: Section 2 represents Research Methodology. Section 3 explains the Experiments. Experimental Results and Discussion are explained in Section 4, and in Section 5 Conclusion is drawn.

II. RESEARCH METHODOLOGY

In this section, we describe the components and architecture of the proposed forecasting model, termed as DISM in detail. Here we are predicting the closing price of the Indian stock market based on the last seven days data. This proposed model (DISM) is influenced by the deep stacking network (DSN) [25] architecture.

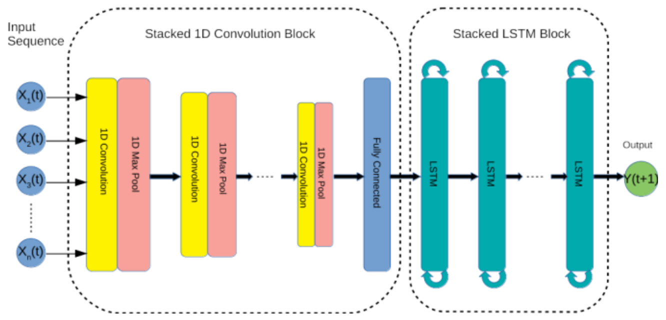


Fig. 1.Proposed Network Model.

The proposed neural network model (DISM) is depicted in Fig. 1. In this model, we integrate two stacked neural network architecture to construct a deep learning-based integrated stacked model (DISM) where the first part of the model is a stacked 1D convolution neural network block followed by stacked LSTM neural network block. The overall architecture of the model is depicted in Fig. 2.

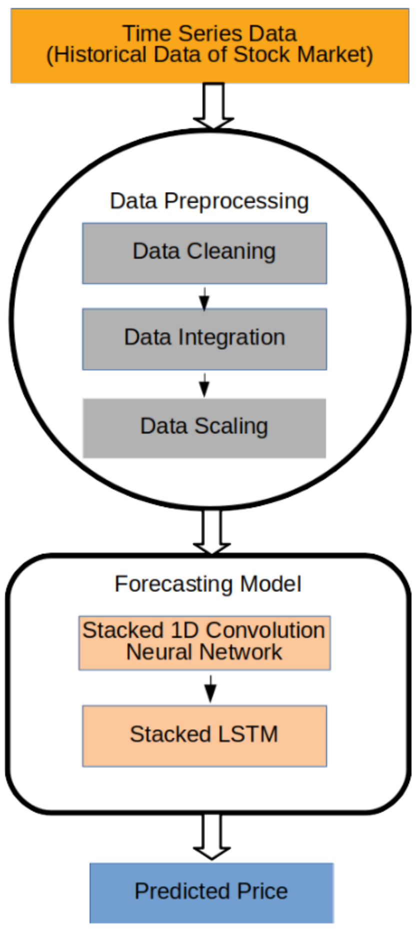


Fig. 2.Architecture of the Proposed Network Model.

A. Stacked 1D CNN Block

1D Convolution neural network (Conv1D) is a special form of Convolution Neural Network (CNN). Conv1D specially used to handle the sequential data. Conv1D consists of a convolution layer followed by a pooling layer. The convolution layer performs the convolution operation on the input data with the kernel, and the output is referred to as the feature map.

The next layer is the pooling layer which uses a pooling function to replace the output of the convolution layer at a certain location by the summary statistics of the nearby outputs.

The architecture of the Stacked 1D Convolution neural network (1D CNN) block depicted in Fig. 3. This stacked 1D CNN block consists of multiple 1D Convolution blocks, which again consist of a 1D convolution layer, a pooling layer, and an activation layer. Each convolution block performs the convolution operation, between input feature matrix *X* with the convolution kernel *W*. These convolution operations, can learn the local features (patterns) within a window of convolution. Then this output is passed to the pooling layer to prevent the overfitting of the learned features.

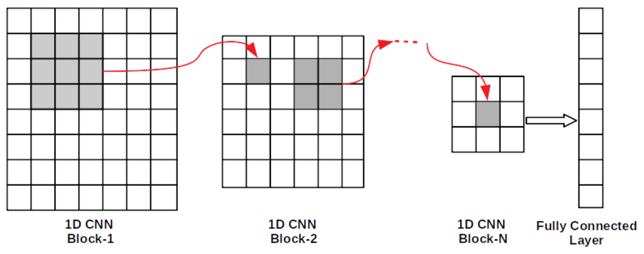


Fig. 3.Architecture of the Stacked 1D CNN Block.

The input signal of a 1D convolution block is the output features signal X of the order $m \times n$ of the previous 1D Convolution block. The convolution kernel W of size $k \times l$ and s is the stride of the input signal X, we get the output signal Y of the order $(m-s)\times(n-s)$. The output signal $y_{i,j}$ is calculated by using the formulae:

$$y_{i,j} = W_{1:k,1:l} \times X_{i:i+k-1,\,j:j+l-1} \tag{1}$$

Where $y_{i,j} \in Y$.

Then the output signal is passed to the pooling layer. In this experiment, we applied *MaxPooling* function on each of the elements of the output signal *Y*, we get the new output features signal *Y'*.

Then the *ReLU* non-linear activation function f is applied on Y' to get the new output features signal Y''.

$$Y'' = f(Y') \tag{2}$$

Where, the *ReLU* activation function as follows:

$$f(z) = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{if } z \le 0 \end{cases} \tag{3}$$

in this way the stacked 1D Convolution block acts as a pre-processing step to extract the sequence of a higher level features.

B. Stacked LSTM Block

Here, we utilize a series of LSTM blocks to deal with this long sequence of high-level features. The main reason for the selection of LSTM recurrent neural networks over RNNs is that RNNs suffer from the vanishing gradient problem during the backpropagation phase. Due to this reason, RNNs are not capable to learn the features from the long sequences with significant delays.

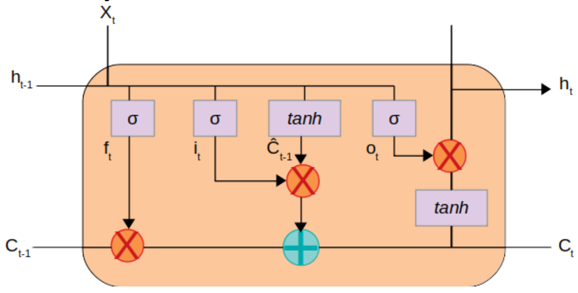


Fig. 4.LSTM Architecture.

There are so many variants of LSTM architectures that have been proposed in recent times. It has been observed that none of the variants of the LSTM architecture is significantly improved over the standard LSTM [26]. So, in this study, we have adopted the standard LSTM architecture. The basic structure of the LSTM architecture is shown in Fig. 4.

The key part of the LSTM architecture is its cell state (memory cell), which consists of three gates viz. forget gate, input gate, and output gate.

The forget gate tells whether the information is to be remembered or not. It removes the information which has no importance or less importance via the multiplication of a filter. The input gate is responsible for adding the information into the cell state. Finally, the output gate selects useful information from the current cell sate to output this information. These operations are performed by the following equations:

$$f_t = \sigma_g(W_f X_t + U_t h_{t-1} + b_f) \tag{4}$$

$$i_t = \sigma_g(W_i X_t + U_i h_{t-1} + b_i) \tag{5}$$

$$o_t = \sigma_g(W_o X_t + U_o h_{t-1} + b_o) \tag{6}$$

$$\tilde{C}_t = \tanh(W_c X_t + U_c h_{t-1} + b_c) \tag{7}$$

The cell sate C_t and the layer output h_t can be calculated for each time interval t by the following equations:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{8}$$

$$h_t = o_t * \tanh(C_t) \tag{9}$$

From the existing studies [27]–[29], it has been observed that deep LSTM architecture with multiple hidden layers can effectively improve the overall performance for the sequential data. Inspired by these facts, we have developed a stacked LSTM block consisting of several LSTM blocks. The architecture of the stacked LSTM block, which is used in our proposed model is shown in Fig. 5. The output of one LSTM block is fed as the input of the next LSTM block.

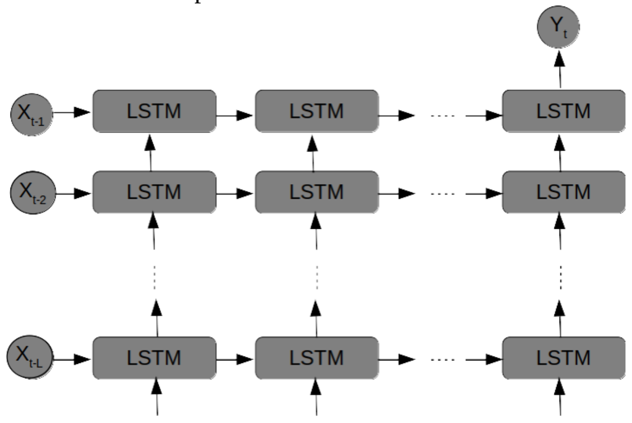


Fig. 5.Architecture of the stacked LSTM Block.

In this way, the DISM can predict the multiple future time step of the stock market based on the historical data of the stock market.

III. EXPERIMENTS

The flow chart of the experiment is depicted in Fig. 6. In this work, all the experiments are done by the Python programming language using Keras deep learning library. This proposed forecasting method consists of the following main steps.

A. Data Set Description

In this work, we have taken the historical time-series data of the Bombay Stock Exchange (BSE) from 1st January 2014 to 31st December 2018 on a daily basis from the website of Yahoo Finance[30]. Here, we consider only five important parameters of the stock market data, viz. opening price, low price, high price, volume, and closing price to predict the closing price of the stock market.

X = {x(1), x(2), ... , x(t)}, where *x(t)* is the most recent value. Our goal is to forecast the future value *x(t + 1)* at timestamp *(t+1)* from previous *n* number of values.

B. Data Preprocessing

The success of any forecasting model heavily depends on the data preprocessing. Most of the time-series data are complex, heterogeneous and noisy. So, data preprocessing is very much essential before the feature extraction from the time series data. Data preprocessing consists of three steps, viz. data cleaning, data integration, and normalization.

At first, we have performed the data cleaning to fill-up the missing data and to remove the outliers present in the raw data set to deal with the incomplete (missing of some of the attribute values), noisy (errors or outliers) data respectively.

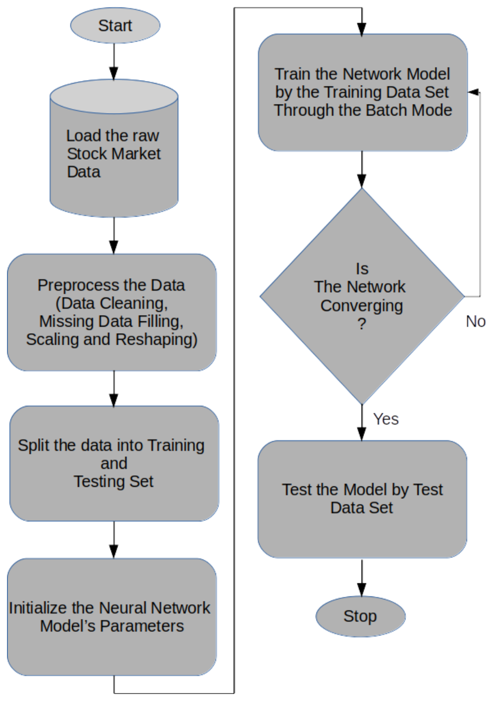

Start
Load the raw Stock Market Data
Preprocess the Data (Data Cleaning, Missing Data Filling, Scaling and Reshaping)
Split the data into Training and Testing Set
Initialize the Neural Network Model's Parameters
Train the Network Model by the Training Data Set Through the Batch Mode
Is The Network Converging ?
No
Yes
Test the Model by Test Data Set
Stop

Fig. 6.Flow Chart of the Experimental Process.

Second, data can be collected from the various sources in various formats and to feed these data into the model, we have adopted the proper data integration method.

Finally, we have applied the data normalization technique [31], [32] to scale the data values into the range of [0, 1]. Since the times series data like stock market data can vary over a large range. So, to produce high-quality data that can be fed into any learning algorithm and to speed up the learning process, data normalization is done.

Before starting the training process, we split the entire data set into three parts. The first part contains 70% of data, which is used for the training purpose. The second part contains 15% of data and it is used for validation purposes. The remaining 15% of the data is used for testing purposes.

C. Training Methodology

We have trained our DISM by the training data set to extract the features. We have fed the seven days data as the input window to the network to predict the closing price of the 8th day. After performing the parameter tuning, we have set the number of training epochs to 100, the window stride length is 1 day, and the input batch size is 20.

In this experiment, we have used backpropagation with stochastic gradient descent (SGD) optimizer to learn the network parameters. During the training phase, overfitting may occur. If we observe the learning curve that describes the error rate against the number of epochs, the point in the curve where the test error starts in increasing but still the training error is decreasing then we can say that the overfitting has occurred. To deal with the overfitting problem, here we have used the k-fold cross-validation to identify the stopping condition of the training process.

In Table I, we have summarized the experimental settings that have used in this work for different models.

Table- I: Experimental Settings.

Model	**LSTM Block**	**1D CNN Block**	**Hidden Units**	**Epochs**
DISM (1-1-32)	1	1	32	100
DISM (1-1-64)	1	1	64	100
DISM (2-2-32)	2	2	32	100
DISM (2-2-64)	2	2	64	100
DISM (2-3-32)	2	3	32	100
DISM (2-3-64)	2	3	64	100
DISM (3-2-32)	3	2	32	100
DISM (3-2-64)	3	2	64	100
DISM (3-3-32)	3	3	32	100
DISM (3-3-64)	3	3	64	100
Model	**Block**		**Hidden Units**	**Epochs**
Stacked LSTM(2-32)	2		32	100
Stacked LSTM(2-64)	2		64	100
Stacked LSTM(3-32)	3		32	100
Stacked LSTM(3-64)	3		64	100
Model	**Block**			**Epochs**
Stacked 1DCNN(2)	2			100
Stacked 1DCNN(3)	3			100

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we have described the experimental results for the training and testing of all the neural network models (Stacked LSTM, Stacked 1D CNN, and DISM). All the neural network models are trained and tested by the Python programming language using Keras deep learning library. Here, we have used Mean Squared Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as the performance evaluation metric and these errors are calculated by using the following formula:

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(o_j - \hat{o}_j)^2 \qquad (10)$$

$$MAE = \frac{1}{n}\sum_{j=1}^{n}(|\, o_j - \hat{o}_j \,|) \qquad (11)$$

$$MAPE = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{|o_j - \hat{o}_j|}{o_j}\right)*100 \qquad (12)$$

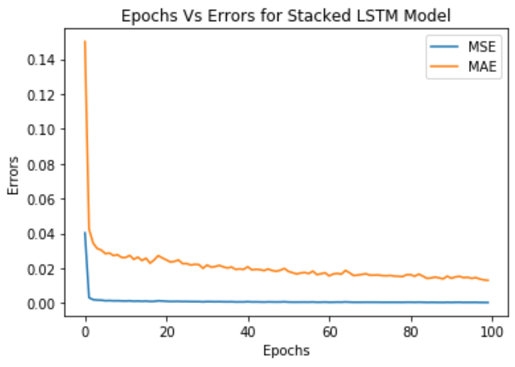
Where n is the number of observation, o_j is the actual value and $\hat{o}_j$ is the predicted value.

In Table-II, we have represented MSE and MAE training errors of our proposed DISM with the different number of epochs. We have graphically represented the MSE and MAE training errors of the different neural network models in Fig. 7. We have compared the performance of DISM with all other neural network models in terms of MSE and MAE during the training phase in Fig. 8. In Fig. 9, we have graphically represented the actual and the predicted closing price values of the BSE dataset during the testing phase of the different neural networks models. In Table-III, we have shown the testing errors of the different prediction models with different experimental settings.

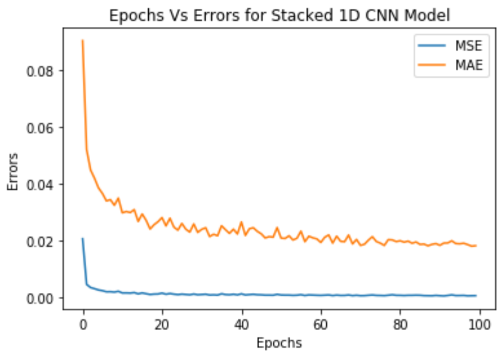
From Table-II, it is quite clear that all versions of DISM have the lowest training errors for 100 number epochs. Among them, DISM (2-2-64) has the lowest training errors. We can observe that, Stacked LSTM Model and DISM have greater convergence speed with respect to the number of epochs compared to the Stacked1D CNN Model from Fig. 7 and Fig. 8. From Table-III, we can observe that all the neural network models outperform the ARIMA and SVM models. Among all the neural network models, our proposed DISM with different parameter settings has the lowest forecasting errors. Additionally DISM(2-2-64) exhibits better performance among all other versions of DISM. It is also quite clear from Fig. 9, DISM outperforms all other models in terms of predicting the closing price of the BSE. It may also be noted that the Stacked LSTM Model produces a very good result in terms of prediction of the closing price and forecasting errors, whereas the Stacked1D CNN Model produces poor performance.

Table- II: Training errors of different DISM models with different number of epochs.

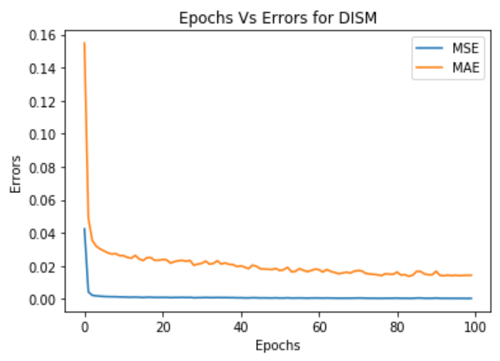
Model	Epochs	MSE	MAE	Model	Epochs	MSE	MAE
DISM (1-1-32)	50	0.00275	0.0297	DISM (2-3-32)	50	0.00209	0.0289
DISM (1-1-32)	100	0.00271	0.0292	DISM (2-3-32)	100	0.00203	0.0263
DISM (1-1-32)	150	0.00273	0.0295	DISM (2-3-32)	150	0.00208	0.0273
DISM (1-1-64)	50	0.00258	0.0268	DISM (2-3-64)	50	0.00197	0.0253
DISM (1-1-64)	100	0.00253	0.0263	DISM (2-3-64)	100	0.00193	0.0249
DISM (1-1-64)	150	0.00257	0.0265	DISM (2-3-64)	150	0.00195	0.0279
DISM (2-2-32)	50	0.00231	0.0297	DISM (3-3-32)	50	0.00213	0.0281
DISM (2-2-32)	100	0.00201	0.0257	DISM (3-3-32)	100	0.00211	0.0297
DISM (2-2-32)	150	0.00214	0.0261	DISM (3-3-32)	150	0.00213	0.301
DISM (2-2-64)	50	0.00267	0.0283	DISM (3-3-64)	50	0.00210	0.0275
DISM (2-2-64)	**100**	**0.00163**	**0.0187**	DISM (3-3-64)	100	0.00199	0.284
DISM (2-2-64)	150	0.00195	0.0241	DISM (3-3-64)	150	0.00205	0.0287



(a)

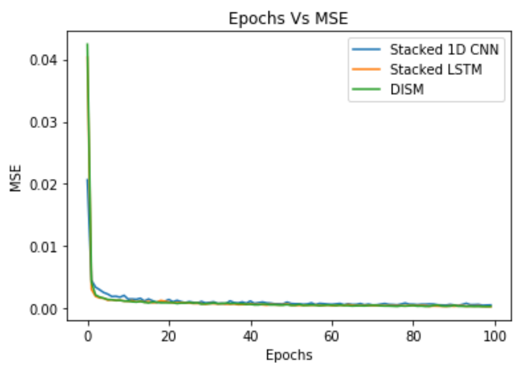


(b)

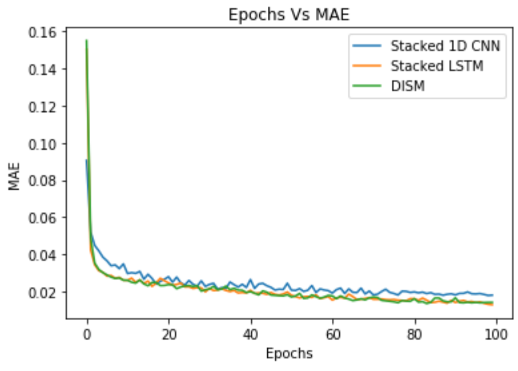


(c)

Fig. 7.MSE and MAE Training errors of different models for the BSE data set. (a)Stacked LSTM Model(3-32) b) Stacked 1D CNN(3) Model and (c) DISM(2-2-64).

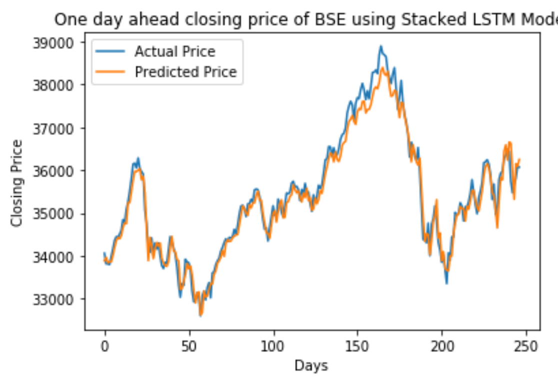


(a)

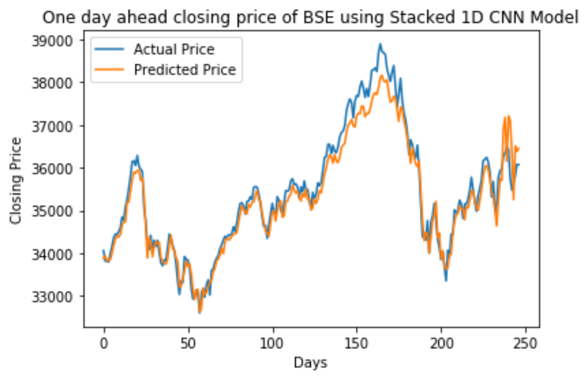


(b)

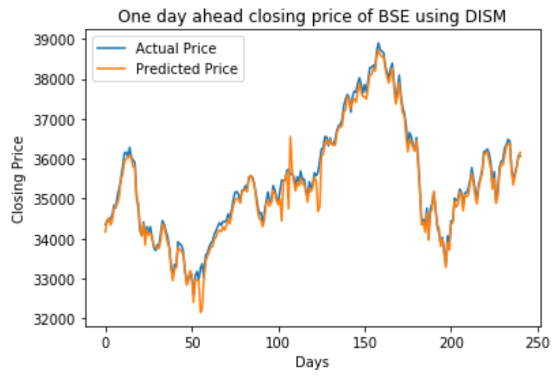
Fig. 8.Comparison of MSE and MAE training errors of DISM (2-2-64), Stacked LSTM (3-32) and Stacked 1D CNN (3) Models for the BSE dataset. (a) MSE and (b) MAE.



(a)



(b)



(c)

Fig. 9.Comparison of actual and predicted stock indices of BSE of different Neural Network Models. (a) Stacked LSTM (3-32) Model, (b) Stacked 1D CNN(3) Model and (c) DISM(2-2-64)

Table- III: Forecasting errors of different models.

Model	MSE	MAE	MAPE
SVM	0.08901574071	0.132401872310	10.23561020126
ARIMA	004185231812	0.075678201497	5.856253410591
Stacked LSTM(2-32)	0.003125630210	0.014234612045	1.610125432401
Stacked LSTM(2-64)	0.003102315801	0.014210452157	1.609246910272
Stacked LSTM(3-32)	0.002990051564	0.013453220382	1.600362156084
Stacked LSTM(3-64)	0.003012543516	0.013466452012	1.608328645901

Stacked 1DCNN(2)	0.025231409801	0.034901294186	4.164482949716
Stacked 1DCNN(3)	0.025028326793	0.034850853752	4.160576280137
DISM (1-1-32)	0.001593012452	0.013572435012	0.315284695032
DISM (1-1-64)	0.001579531470	0.013928597301	0.314670135917
DISM (2-2-32)	0.001551204591	0.011925861024	0.311562754039
DISM (2-2-64)	**0.001526468174**	**0.010830290456**	**0.304806144709**
DISM (2-3-32)	0.0015928530121	0.0120451301	0.310859470347
DISM (2-3-64)	0.0015910235676	0.0120158943	0.310762784381
DISM (3-3-32)	0.0016152379240	0.0124025419	0.313587394428
DISM (3-3-64)	0.0016159513708	0.0124972504	0.313601836201

V. CONCLUSION

In this paper, we have proposed a Deep Learning-based Integrated Stacked Model (DISM) for forecasting the stock market indices. We also compared our model with the Support Vector Machine (SVM), the AutoRegressive Integrated Moving Average (ARIMA), the Stacked LSTM model, and the Stacked 1D CNN model.

We can easily conclude that our proposed model outperforms all other models for the forecasting of the multi-variable and non-linear time-series dataset. Although all the neural network models produce very good prediction results and low error rate compares to other statistical models. We can also conclude that spatial and temporal feature extraction is one of the most important aspects of the stock market price prediction.

We can also extend our experiment in different directions in proposing novel neural network architecture. In this paper, we have applied our proposed model to predict the Indian stock market indices. In the future, we would like to apply our proposed model on the different linear and non-linear time-series data. We will also want to improve our model for better accuracy.

ACKNOWLEDGMENT

The Authors will like to thank Hewlett Packard (HP) for their generous grant of setting up of a Machine Learning Laboratory which includes a HP Workstation Z4 G4 Pike along with two units of 24' Display devices to the Principal Investigator for conducting the research.

REFERENCES

1. H. Tong, Non-linear time series: a dynamical system approach. Oxford University Press, 1990.
2. R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," Econom. J. Econom. Soc., pp. 987–1007, 1982.
3. G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," J. Am. Stat. Assoc., vol. 65, no. 332, pp. 1509–1526, 1970.
4. M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," Atmos. Environ., vol. 32, no. 14–15, pp. 2627–2636, 1998.
5. R. Law, "Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting," Tour. Manag., vol. 21, no. 4, pp. 331–340, 2000.
6. M. Hüsken and P. Stagge, "Recurrent neural networks for time series classification," Neurocomputing, vol. 50, pp. 223–235, 2003.
7. G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," Int. J. Forecast., vol. 14, no. 1, pp. 35–62, 1998.
8. J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," IEEE Trans. neural networks, vol. 5, no. 2, pp. 240–254, 1994.
9. S.-L. Ho, M. Xie, and T. N. Goh, "A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction," Comput. Ind. Eng., vol. 42, no. 2–4, pp. 371–375, 2002.
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.
11. T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," IEEE Trans. Energy Convers., vol. 21, no. 1, pp. 273–284, 2006.
12. M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, 1997.
13. M. Assaad, R. Boné, and H. Cardot, "A new boosting algorithm for improved time-series forecasting with recurrent neural networks," Inf. Fusion, vol. 9, no. 1, pp. 41–55, 2008.
14. A. K. Mishra and V. R. Desai, "Drought forecasting using feed-forward recursive neural network," Ecol. Modell., vol. 198, no. 1–2, pp. 127–138, 2006.
15. Y. Bengio, P. Simard, P. Frasconi, and others, "Learning long-term dependencies with gradient descent is difficult," IEEE Trans. neural networks, vol. 5, no. 2, pp. 157–166, 1994.
16. M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," Pattern Recognit. Lett., vol. 42, pp. 11–24, 2014.
17. A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," Neurocomputing, vol. 323, pp. 203–213, 2019.
18. Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," IET Intell. Transp. Syst., vol. 11, no. 2, pp. 68–75, 2017.
19. R. Xiong, E. P. Nichols, and Y. Shen, "Deep learning stock volatility with google domestic trends," arXiv Prepr. arXiv1512.04916, 2015.

20. Z. Cui, R. Ke, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," arXiv Prepr. arXiv1801.02143, 2018.
21. R. Mittelman, "Time-series modeling with undecimated fully convolutional neural networks," arXiv Prepr. arXiv1508.00317, 2015.
22. A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," arXiv Prepr. arXiv1703.04691, 2017.
23. L. Zhong, L. Hu, and H. Zhou, "Deep learning based multi-temporal crop classification," Remote Sens. Environ., vol. 221, pp. 430–443, Feb. 2019.
24. M. Patel, A. Patel, D. Ghosh, and others, "Precipitation Nowcasting: Leveraging bidirectional LSTM and 1D CNN," arXiv Prepr. arXiv1810.10485, 2018.
25. M. Das and S. K. Ghosh, "Deep-STEP: A deep learning approach for spatiotemporal prediction of remote sensing data," IEEE Geosci. Remote Sens. Lett., vol. 13, no. 12, pp. 1984–1988, 2016.
26. K. Greff, R. K. Srivastava, J. Koutn\'\ik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," IEEE Trans. neural networks Learn. Syst., vol. 28, no. 10, pp. 2222–2232, 2017.
27. A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in 2013 IEEE workshop on automatic speech recognition and understanding, 2013, pp. 273–278.
28. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, p. 436, 2015.
29. J. S. Bayer, "Learning Sequence Representations," Technische Universität München, 2015.
30. "BSE SENSEX - Yahoo Finance," 2019. [Online]. Available: https://in.finance.yahoo.com/quote/%5EBSESN/history?p=%5EBSESN. [Accessed: 10-Jul-2019].
31. S. Bhanja and A. Das, "Impact of Data Normalization on Deep Neural Network for Time Series Forecasting," arXiv Prepr. arXiv1812.05519, 2018.
32. S. C. Nayak, B. B. Misra, and H. S. Behera, "Impact of data normalization on stock index forecasting," Int. J. Comput. Inf. Syst. Ind. Manag. Appl., vol. 6, no. 2014, pp. 257–269, 2014.

AUTHORS PROFILE

Mr. Samit Bhanja is a PhD scholar with the Dept. of Computer Sc. & Engineering at Aliah University, Kolkata. His research area in Machine Learning and Deep learning. He is currently working as an Assistant Professor, Dept. of Computer Science, Government General Degree College, Hooghly. He has also worked as an Assistant Professor in the department of Computer Science & Engineering, Seacom Engineering College, Howrah and in the department of computer science, Scottish Church College, Kolkata. He has a MTech in Computer Science and Engineering from Maulana Abul Kalam Azad University of Technology (formerly West Bengal University of Technology). He is also the Associate Member of The Institution of Engineers (INDIA).

Dr. Abhishek Das is currently working as the Head and Associate Professor in the Dept. of Computer Sc. & Engineering at Aliah University, Kolkata. His Research Area is in Medical Image Processing, Computer Vision and IoT. He has received a Post-Doctoral Research position from University of West Scotland, UK by a Fellowship received from the European Commission. He has also worked as an Assistant Professor in the Dept. of Information Technology, Tripura University (A Central University), India. He has worked previously as a Reader in Computer Sc. & Engineering, Indian Institute of Space Science & Technology (IIST) and Lecturer in Information Technology, Bengal Engineering and Science University, Shibpur (now IIEST Shibpur) His PhD is from the Dept. of Computer Sc. & Engineering, Jadavpur University, India. His M.S. in Electrical & Computer Engineering from Kansas State University, USA and B.Tech in Computer Science & Engineering from Kalyani University, India. He has worked as a Business System Analyst in Blue Cross Blue Shield New York, USA & as a Quality Analyst in Vensoft Inc. Phoenix, USA and also as a Consultant Project Manager in the Software Industry. He is also a Visiting Scientist at Indian Statistical Institute, Kolkata. He has also worked in Technical Educational Administration as Regional Officer and Asst. Director (on deputation) at AICTE(under MHRD, Govt. of India). He is a NCERT National Scholar, New Delhi and a Tilford Dow Scholar, USA. His Biography has been published in the 28th edition of National Dean's List, USA. He is a Member of IEEE and a Honorary Senior Member of IACSIT Singapore. He has several publications in International and National Journals and peer-reviewed Conferences. He is also a TPC Committee Member of IEEE International conferences in South-East Asia and Editor of the Journal of Image Processing and Pattern Recognition progress(A UGC CARE Journal) and Special Issues Journal on Image Processing & Computer Vision of IGI Global Publishers, USA (SCI and Scopus Indexed).