Journal of Industrial Information Technology and Application *ISSN(Online): 2586-0852*
Vol.6 No.3

Evaluation of Deep Learning Models for Predicting the Support and Resistance Levels in Stock Market

T. Indhumathy[1], T. Velmurugan[2*]*

[1]Research Scholar, PG and Research Department of Computer Science,
D. G. Vaishnav College, Chennai-600106, Tamil Nadu, India
[2]Associate Professor, PG and Research Department of Computer Science,
D. G. Vaishnav College, Chennai-600106, Tamil Nadu, India

Abstract. Deep Learning evolved from Artificial Intelligence and has been used to solve problems where Machine Learning faces dead ends. The Design and Architecture of the Neural Networks, a Deep Learning paradigm is one of the major factors in deciding how successful the technologies in Deep Learning models are implemented. The architectures have evolved with varying applications and the impact it has on its output. Out of the several architectures, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are taken, modeled for stock exchange to predict the price levels and evaluation on its performance was taken up for study.Two historical datasets containing 5021 of financial data each were taken for the analysis. Out of these 4016 data were taken to train the data and 1005 data were taken as test data. The accuracy and performance analysis were determined by the error metrics and the computation time is sought to determine how well the models have fit the dataset while predicting in stock exchange.

Keywords: Long Short-Term Memory, Gated Recurrent Unit, Support and Resistance Levels, Error Metrics, Hyperparameters.

1. Introduction

In this modern era technologies is much sought after, which ease human tasks and used in various applications which is highly influential on how the world thinks and interprets activities. Machine Learning and Deep learning have set a major break at how technologies are implemented and are made to look like how human perceives

* Corresponding author: tindhu_m3@yahoo.com, velmurugan_dgvc@yahoo.co.in
Received: Jan 22, 2022; Accepted: Feb 11, 2022; Published: Jun 30, 2022

things. Machine Learning methodologies had its own drawbacks and by deploying Deep Learning, most of the tech companies have witnessed a desirable growth and improvement in their product and services which is one of the major factors that is attributed to the successful implementation of technologies.Deep learning was inspired by Artificial Intelligence - a subject viewed as a technical dimension that tries to achieve the capability to perceive, think, act in a similar way humans do and the ability to rationalize, take actions for achieving the desired goals [1]. The foundations of their functionality lies in the mathematical and statistical theories [2] designed as neurons which are simple and connected processors, applied as algorithms and implemented in manmade architectures that were inspired from the human cerebral cortex and other brain regions.

Built upon an idea based on manmade neurons, Neural Networks are known to be the backbone of Deep Learning and approach the problem by deciphering them in a different way.With ever increasing data, neural networks is capable of taking large number of data, train those data, develop a system obtained by training and automatically infer rules that are adept for the problem. These networks are designed to work in layers which consist of weights, bias and the inputs are transformed into outputs through activation functions [3].This paradigm with the technology to learn from the data has surpassed the limitations encountered by machine learning algorithms and has been successfully implemented in processing and classification of images, audio & video [4], speechrecognition and predictive problems [5]. One such successful system implementation was proposed by Fayek et al [6] adhering to Speech Emotion Recognition that was applicable in various deep learning architectures. The authors compared, analyzed the system and concluded that the best results have been achieved. The system could be applicable in other applications pertaining to speech recognition and could provide a platform for devising novel deep learning architectures.

Nevertheless its application varies from Natural Language Processing, Forecasting [7] and Logistical optimization to Robotic applications. A paper by S. A. Hasan and OladimejiFarri [8] gives a brief account of how deep learning is applied to clinical language processing and has also elaborated about the different applications pertaining to clinical data. A brief analysis on the detection of grasping points by robotic systems using deep learning was submitted by Shehan Caldera et al [9]. The authors have discussed the many methods where successful implementations have been achieved and the overall benefits, limitations and a promising development in the future for this application while applying deep leaning approaches in the field of robotics.

Neural Networks and its components have undergone various modifications that were measured according to their performances, upgradations and variations in deriving the desired outputs [10]. Miikkulainen et al [11] have put forth the notion that establishing architectures and modeling according to the application is a challenging task. They have suggested a model that optimizes deep learning architectures and fitted to cater in the field of object recognition and language modeling, applying in a magazine website by capturing images and proving that the approach can be implemented to get better results in various other applications.Major breakthroughs in Deep Learning are still yet to be achieved and the study of different network architectures [12] allow us to assess the strengths and weaknesses and further facilitate Deep Learning for achieving goals ultimately.

This paper attempts to evaluate two of the neural network architectures and analyze its predictive and computational performance by applying it in two historical datasets from Indian stock exchange. Support and Resistance level in a stock determine the maximum range the price level goes down and up respectively before reverting over a certain period of time. The levels are studied to determine the price points where the investor can choose to buy or sell the shares at maximum advantage. This paper is organized as follows. Section 2 briefs about the architectures and the datasets in which it has been modeled. Section 3 analyses the performance and discusses the results that are obtained. The conclusion of the paper is presented in section 4.

2. Deep Learning Architectures

Neural networks have a plethora of architectures that are built with varying rules pursued by the distinct characteristics of the inputs and the specific output as demanded by the applications. Initially developed from perceptrons, the simplest networks are the Feed Forward Neural Network, Convolutional Neural Network, Multilayer Perceptron, Radial Basis Function Neural Network, RecurrentNeural Network [13]. Eventually the networks evolved to much more advanced architectures [14] like Alexnet, VGG net, Goolge Net, ResNet, Region based CNN, developed for applications that need more flexibility or adaption according to the desired outcome. For applications related to sequential data the architecture based on Recurrent Neural Networks (RNN) [15], have been preferred over than the previous architectures in Deep Learning. But its application has been limited by the “vanishing” gradient problem and to overcome this issue, architectures based on retaining memory had been introduced. Long Short-Term Memory and Gated Recurrent Unit are the two architectures which retained the memory of the previous layers thereby providing a solution to the problem of vanishing gradient. This paper uses the above mentioned two models and predicts the price level in two stock market historical datasets and analyzes its strengths and limitations.

2.1 Long Short-Term Memory Model

The architecture of Long Short-Term Memory (LSTM) is a modification of recurrent neural network to preserve the data for a longer duration was invented by Josef "Sepp" Hochreiter and Jürgen Schmidhuber in 1997. Developed to solve the “vanishing gradient problem”, basically a LSTM unit comprises of a memory cell that supports the layers in the network to retain the information which eventually does not permit loss of information. And the result is more accurate in sequential data or any other application that needs to be more precise. It comprises of three gates, a memory cell and a hidden state. Discarding unwanted data is the responsibility of the forget gate and the data that has to be displayed is determined by the output gate.

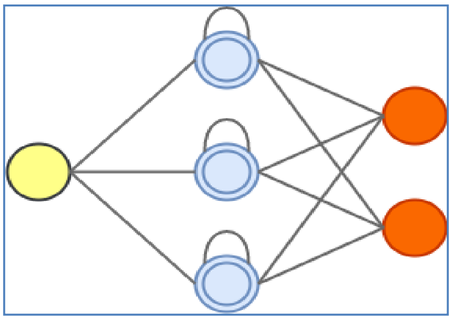

Fig. 1: Structure of LSTM network

In Fig 1 the input gate is depicted as the yellow filled circle, determines what data should be allowed into the cell along with the memory cell which is shown as blue filled circles that contains the activation functions. The data that should enter the next sequence is taken care of by the hidden state and the orange filled circles are the output gates as depicted in Fig 1.

The following transition equations represent the basic architecture of LSTM, where i_t is the input gate, f_t is the forget gate, o_t is the output gate, g_t is the candidate hidden state, h_t is the output hidden state, c_t is the internal memory state and U & W are the weights used for training the gates and t denotes the time.

$$it = \sigma\left(xtU^{i} + ht - 1W^{i}\right) \quad (1)$$

$$ft = \sigma\left(xtU^{f} + ht - 1W^{f}\right) \quad (2)$$

$$ot = \sigma\left(xtU^{o} + ht - 1W^{o}\right) \quad (3)$$

$$gt = \tanh\left(xtU^{g} + ht - 1W^{g}\right) \quad (4)$$

$$ht = \tanh(ct).o \quad (5)$$

$$ct = ct - 1.f + g.i \quad (6)$$

There had been many variants [16], [17] of LSTM since its inception and is used in applications like time series analytics, classification problems, natural language processing, communication, forecasting and prediction.

2.2 Gated Recurrent Unit Model

Gated Recurrent Unit(GRU) also a modification of recurrent neural network is almost similar to LSTM in architecture except that it has two gates instead of three and has fewer parameters as depicted in Fig 2. There is no output gate but has an update gate and a reset gate along with a current memory state. The amount of information that should flow into memory is controlled by the update gate. The amount of information that should flow out of the memory is controlled and the effect of the previous data that has on the present data is suppressed by the reset gate.

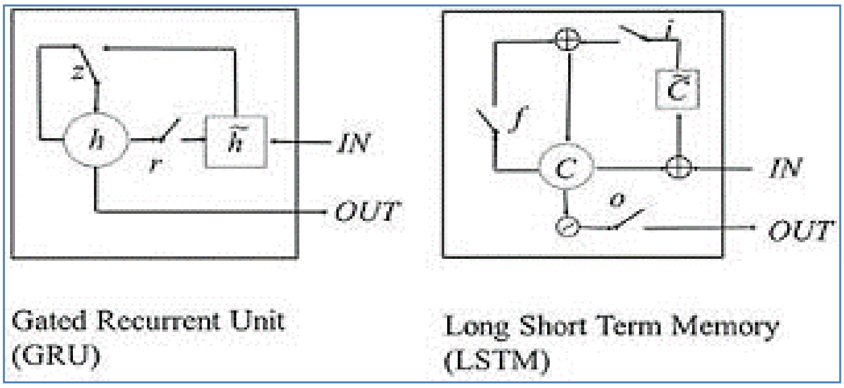

Fig. 2: Architecture of GRU and LSTM

The architecture of GRU`s implementation is denoted by the following equations where r_t is the reset gate, z_t is the update gate, k is the output state, h_t is the hidden state U & W are the weights assigned to train the gates and t denotes time.

$$rt = \sigma\left(xtU^{r} + ht - 1W^{r}\right) \quad (7)$$

$$zt = \sigma\left(xtU^{z} + ht - 1W^{z}\right) \quad (8)$$

$$k = \tanh\left(xtU^{k} + \left(ht - 1.r\right)W^{k}\right) \quad (9)$$

$$ht = \left(1 - z\right).k + z.ht - 1 \quad (10)$$

This architectural modification enables GRU to execute faster, use less memory, applicable for short time sequences and works better for small and sparse datasets. So for specific applications like polyphonic music modeling it shows better results than LSTM [18] and variants of GRU [19] creates a possibility to extend its applications related to bioinformatics, network intrusion, health monitoring [20] and various other fields.

2.3 Description of Dataset

Two datasets that were taken for analysis were obtained from Kaggle which is an online repository for datasets. These datasets are of historical type that contains financial equity stock details dated from 3rd Jan 2000 till 28th Feb 2020. Each of the dataset contains a total of 5021 values from which 80% of the total data was taken for training data and the rest 20% as testing data. The variables Date and Close indicating the date and the price at which the stock was closed were taken for predicting the prices and the rest of the variables that define the datasets are Previous closing Price, Opening price, the highest and the lowest prices of the stock for the day, volume weighted average price, total volume of sales and turnover. The condensed set of samples from both the datasets are shown as Table 1 and Table 2 respectively and the format of these datasets is Excel comma separated value (csv).

Table 1: Sample Data from dataset1 – AXIS BANK

Date	**Open**	**High**	**Low**	**Last**	**Close**
1/3/2000	26.7	26.7	26.7	26.7	26.7
1/4/2000	27	28.7	26.5	27	26.85
1/5/2000	26	27.75	25.5	26.4	26.3
1/6/2000	25.8	27	25.8	25.9	25.95
1/7/2000	25	26	24.25	25	24.8

Table 2: Sample Data from dataset2 – HDFC BANK

Date	Open	High	Low	Last	Close
1/3/2000	166	170	166	170	170
1/4/2000	182	183.45	171	174	173.8
1/5/2000	170	173.9	165	168	166.95
1/6/2000	168	170	165.3	168.95	168.3
1/7/2000	162.15	171	162.15	170.75	168.35

3. Experimental Results and Performance Analysis

The effectiveness of any neural network architecture depends on how successfully it was modeled and trained. Here LSTM presented in [21] and GRU were modeled to predict the prices that indicate the support and resistance levels through Fibonacci Retracement for two companies that were taken from Indian stock exchange and the level of accuracy attained is taken up for analysis to establish how they have predicted. Major trendlines were considered for assessment of the support and resistance price levels retraced with three of Fibonacci percentages i.e., 23.6, 38.2, 61.8 along with two other percentages 0 and 100 indicating the lowest and highest price levels.

The accuracy is measured by the error metrics and the hyperparameters assess the behavior of a model. Graphs were drawn for illustrative purposes to compare the models accuracy level. Computational time is taken up to determine the level of efficiency achieved by the models so as to judge how well the models have adapted the datasets. For the purpose of training 4016 samples were taken out of 5021 data and the rest 1005 samples for validation purpose. For training the model Adam is set as the optimizer. Google cloud engine was used as a training platform [Machine type: n1-standard-2 (2 vCPUs, 7.5 GB memory), CPU platform: Intel Core i5] and used Windows 7, Keras (Frontend) and Tensorflow (Backend) as the learning environment.

3.1 Error Metrics

For determining the accuracy level of the models the column date was taken as independent variables and the closing price as dependent variables from the dataset. The range of the values for dataset1 for the column closing price lies between 21 to 2050 and 163 to 2566 for dataset2. The error metrics [22] applied were Root Mean Square Error (RSME), Bais, Mean Absolute Error (MAE), Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). The residual obtained is the difference between the actual values and the predicted values. When RSME residuals are taken for analysis it is widely accepted that lower the difference when compared to the lowest range of the dependant variable, higher the level of accuracy.

Table 3: Error Metrics for dataset1- AXIS BANK

Architecture type	**Error Metrics**				
	RMSE	**Bias**	**MAE**	**MSE**	**MAPE**
LSTM	14.01	6.08	10.71	196.46	1.83
GRU	19.70	-6.77	15.25	388.10	2.64

Table 4: Error Metrics for dataset2- HDFC BANK

Architecture type	**Error Metrics**				
	RMSE	**Bias**	**MAE**	**MSE**	**MAPE**
LSTM	53.23	-25.59	28.61	2834.40	1.18
GRU	77.02	35.39	46.74	5932.75	2.72

From Table 3 representing dataset1 it can be seen that the residual values for LSTM is less than the values of GRU though the values for both the models is near the lowest closing price value which implies that the accuracy level is acceptable. To judge the level of accuracy for dataset2, Table 4 displays the residual values obtained from different error metrics. The residual values of LSTM are again less than the values of GRU and the difference between the lowest value of the dependent variable and the residual values is less for both the models. It can be seen that the level of accuracy is high for both LSTM and GRU for dataset1 and dataset2 but in comparison LSTM is more accurate than GRU.

The graphical representation for both the datasets is drawn to analyze the models LSTM and GRU. Through the graphs it can validate the error residuals attained as represented by Table 3 for dataset1 and Table 2 for dataset2. X-axis is plotted with independent variable, in this case the variable date is chosen and the variable close representing the closing price is taken as dependent variable which is plotted along the Y-axis.

Fig. 3:Result of LSTM model for dataset1-Axis Bank

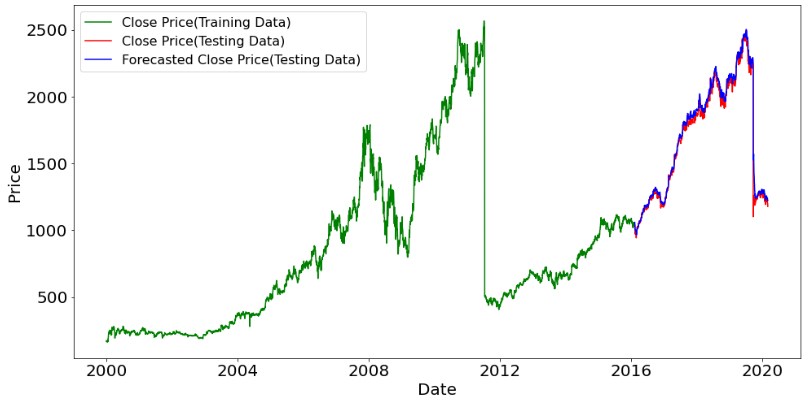

Fig. 4:Result of LSTM model for dataset2 - HDFC Bank

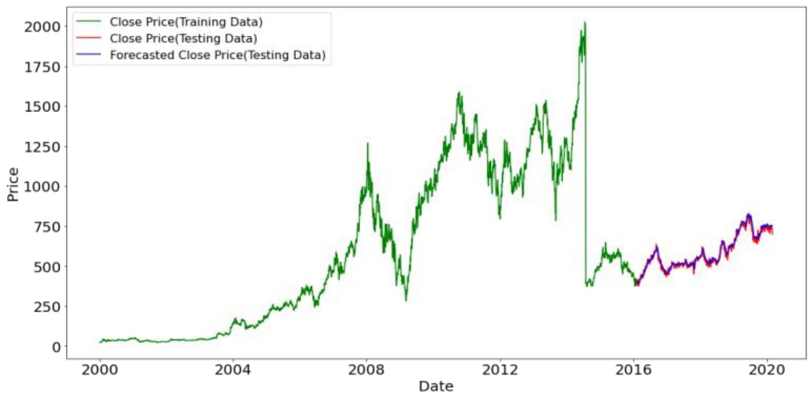

Fig. 5:Result of GRU model for dataset1-Axis Bank

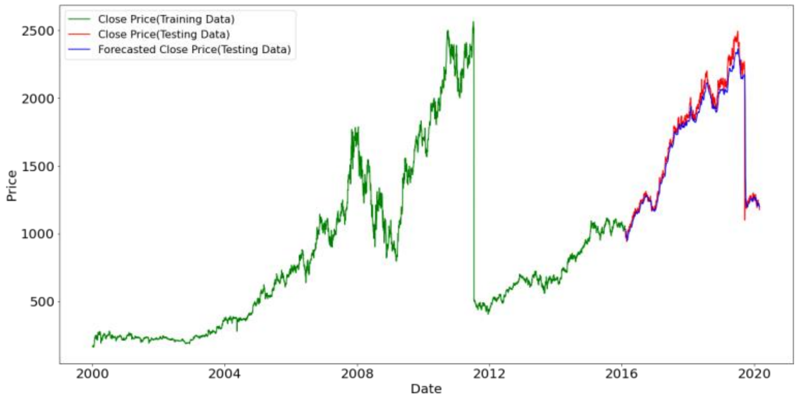

Fig. 6:Result of GRU model for dataset2-HDFC Bank

The green lines represent the 4016 instances which are the training data and the testing data with 1005 instances are represented as red lines. The blue lines indicate the values as forecasted by the models otherwise known as unseen data. The testing data and the unseen data for dataset1 and dataset2 implemented by LSTM are very close and the difference between them is negligible as seen from Fig 3 and Fig 4 which clearly indicates the level of accuracy achieved. The accuracy level of GRU for dataset1 and dataset2 is depicted in Fig 5 and Fig 6 respectively indicating the difference between testing data and the unseen data is under the acceptable range but in comparison LSTM proves to be more efficient than GRU.

3.2 Hyperparameters

The process of learning during training the data are set through hyperparameters. The batch size, number of neurons used, number of hidden layers and epoch were set for the models to assess the learning process in this paper. Hyperparameters [23] are deemed to be important due to the fact that the process of training must be properly tuned considering its effect it has on the performance to get the best possible result and recently adaption to the method of training has been taken for further study so that the

users can reduce their unnecessary effort and time [19].

Table. 5:Hyperparameters set for dataset1-AXIS BANK

Architecture type	Epoch	Batch size	Neurons	Hidden Layers
LSTM	2	1	50	2
GRU	2	1	50	4

Table. 6:Hyperparameters set for dataset2- HDFC BANK

Architecture type	Epoch	Batch size	Neurons	Hidden Layers
LSTM	3	1	50	2
GRU	3	1	50	4

To attain the desired level of accuracy for dataset1 as can be referred from Table 5 the hyperparameters epoch, batch size and the number of neurons is at minimum and same for both LSTM and GRU differing only in the number of hidden layers where GRU needed two more layers to predict. Table 6 lists the hyperparameters set for dataset2 and the value of epoch was raised to one more when compared to the set epoch for dataset1, for both the models. Batch size, number of neurons and number of hidden layers used for dataset2 is the same set values as dataset1 for both the models and differs only in the number of the hidden layers. Considering the values set for the training process it can be inferred that for both the models, not much of an effort was taken for a database with 5021 instances.

3.3 Computation time

In Artificial neural networks the computational complexity is assessed by the parameter weight, time taken to train the dataset, the size and length of the input. In case of neural networks considering the number of operations required for a forward and backward passes, is one of the likely methods to assess the time taken. For Deep Learning methods the total training time is taken up so that a fair judgment on how the models perform may be considered. On the above mentioned basis, in this paper the running time is represented in seconds to show how the models had trained and predicted from the test data.

Table 7: THE COMPUTATION TIME FOR THE DATASETS

Computation time	LSTM (Time in Sec.)	GRU (Time in Sec.)
Dataset1 AXIS BANK	76	168
Dataset2 HDFC BANK	80	146

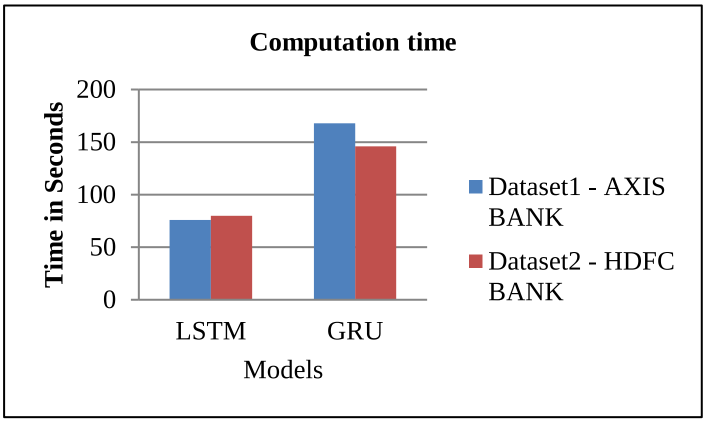

Fig 7: Comparison of the computation time of the models

Table 7 displays the time taken for training for both the models. The values from Table 7 are depicted as bar diagram in Fig 7. The vertical bars represent the two models where the blue bar shows the running time for dataset1 and the red bar shows the running time for dataset2. The time in seconds is plotted at Y-axis and the time taken by GRU is higher for both the datasets when compared to the training time taken by LSTM.

4. Conclusion

Deep Learning has a major impact in modern technology and is considered as a cutting edge solution provider in most of the applications where machine learning meets its limitations. Predicting stock prices is a risky business and the Neural Network paradigms of Deep Learning have aided in reducing the errors while forecasting. Two enhanced architectures of Neural Networks learning, LSTM and GRU, were modeled to predict the support and resistance levels for two stocks from Indian Stock Exchange containing 5021 data each, to determine the exit or the entry price point. The structure of the model was set through four hyper parameters and is taken up to analyze the training process. Epoch, Batch size and Neurons were set to the same values for both the models expect for Hidden layers in which the model GRU needed one more layer than LSTM. With this structure the models are assessed to determine the level of prediction achieved and the training time taken to achieve the desired rate of accuracy. Five types of error metrics were used to ascertain how accurate the predictions were and based on the residual values derived from the error metrics, it is inferred that both the models have achieved a high level of accuracy. But when compared LSTM achieved more accuracy than GRU and the computation time taken by LSTM is much lower than the time taken by GRU. The differences in performance can be attributed to one of the hyperparameter i.e., hidden layers and the size of the dataset. Though both of the architecture has been modeled well, it can be concluded that LSTM is a better fit than GRU both in terms of accuracy and computation time.This research work predicted the support and resistance levelsand the work can be extended further by implementing various multiple technical factors for predicting the stock prices.

References

[1] T. J. Sejnowski, "The unreasonable effectiveness of deep learning in artificial intelligence", Proceedings of the National Academy of Sciences, 2020.

[2] R. Vidal, J. Bruna, R. Giryes and S. Soatto, "Mathematics of deep learning", arXiv preprint arXiv:1712.04741, 2017.

[3] M. A. Nielsen, "Neural networks and deep learning", San Francisco, CA: Determination press, Vol. 25, 2015.

[4] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", nature 521, No. 7553, pp. 436-444, 2015.

[5] T. Velmurugan and T. Indhumathy, "Comparative study on Prediction of Support and Resistance Levels with k-Nearest Neighbor and Long Short-Term Memory Methods", Journal of Information and Computational Science, ISSN: 1548-7741, Vol. 10, Issue 09, pp. 243-252, 2020.

[6] H. M. Fayek, M. Lech and L. Cavedon, "Evaluating deep learning architectures for Speech Emotion Recognition", Neural Networks, Vol. 92, pp. 60-68, 2017.

[7] T. Indhumathy and T. Velmurugan, "Performance of *k*-Nearest Neighbor and Gated Recurrent Unit in Stock Market", Journal of the Maharaja Sayajirao University of Baroda, Vol. 54, No.2 (V), pp. 122-127, 2020.

[8] S. A. Hasan and OladimejiFarri, "Clinical natural language processing with deep learning", In Data Science for Healthcare,Springer, Cham,pp. 147-171, 2019.

[9] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection", Multimodal Technologies and Interaction, Vol. 2, No. 3,p. 57, 2018.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview", Neural networks, Vol. 61, pp 85-117, 2015.

[11] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy and B. Hodjat,"Evolving deep neural networks", In Artificial intelligence in the age of neural networks and brain computing, Academic Press, pp. 293-312, 2019.

[12] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu and F. E. Alsaadi, "A survey of deep neural network architectures and their applications", Neurocomputing, Vol. 234, pp. 11-26, 2017.

[13] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.L. Shyu, S.C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications", ACM Computing Surveys (CSUR), Vol. 51, No. 5,pp. 1-36, 2018.

[14] R. Shanmugamani, "Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras", Packt Publishing Ltd, 2018.

[15] S. Grossberg, “Recurrent neural networks”, Scholarpedia, Vol. 8, No. 2, p. 1888, 2013.
[16] Y. Wang, M. Huang, X. Zhu and Li Zhao, “Attention-based LSTM for aspect-level sentiment classification”, In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 606-615, 2016.
[17] A. PulverandSiweiLyu, “LSTM with working memory”, International Joint Conference on Neural Networks (IJCNN), pp. 845-851, 2017.
[18] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling”, arXiv preprint arXiv:1412.3555, 2014.
[19] G. B. Zhou, J. Wu, C.L. Zhang and Z. H. Zhou, “Minimal gated unit for recurrent neural networks”, International Journal of Automation and Computing, Vol. 13, No. 3, pp. 226-234, 2016.
[20] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen and J. Wang, “Machine health monitoring using local feature-based gated recurrent unit networks”, IEEE Transactions on Industrial Electronics, Vol. 65, No. 2, pp.1539-1548, 2017.
[21] T. Indhumathy and T. Velmurugan, “Predicting Support and Resistance Indicators for Stock Market with Fibonacci Sequence in Long Short-Term Memory”, Journal of Computer Science, Vol. 16, Issue 10, pp. 1428-1438, 2020.
[22] A. Botchkarev, “Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology”, arXiv preprint arXiv:1809.03006, 2018.
[23] J. Brownlee, “How to grid search hyperparameters for deep learning models in python with keras”, Retrieved April, Vol. 20, p. 14, 2018.
[24] L. Mai, A. Koliousis, G, Li, A. O. Brabete, and P. Pietzuch, “Taming Hyper-parameters in Deep Learning Systems”, ACM SIGOPS Operating Systems Review, Vol. 53, No. 1, pp. 52-58, 2019.