*Journal of Statistics & Management Systems*
ISSN 0972-0510 (Print), ISSN 2169-0014 (Online)
Vol. 26 (2023), No. 1, pp. 87–100
DOI : 10.47974/JSMS-949

# ANN based modeling for stock market prediction

Niranjan Panda *
Twinkle Singh †
Shrabanee Swagatika §
*Department of Computer Science and Engineering*
*Siksha 'O' Anusandhan (Deemed to be Univesity)*
*Bhubaneswar*
*Odisha*
*India*

## Abstract

Analyzing stock market data and using recent developed algorithms for predicting the changes and forecasting the results is a difficult task nowaday. An efficient and faster method for selecting appropriate stock price improves market value as well as helps investors for benefits. In this paper new method implementing state of the art Artificial Neural Networks (ANNs) known as Legendre Neural Network (LENN) and Chebyshev Functional Link based Artificial Neural Network (CHFLANN) has been described and used for the analysis of stock market data. The process has been efficiently implemented in MATLAB language for representing results by evaluating accuracy and F-measure for the processed data. Prior to it the database collected is passed through a series of processes starting from normalization, expansion and then dividing it into train and test datasets to get the processed data used for implementing the learning algorithm for the stock market price evaluation and prediction purpose. In the results second order expansion systems have been used showing accuracy results as well as Mean Square Error (MSE) rates to find best method among proposed schemes. Also, our proposed models have been compared with the statistical model AutoRegressive Integrated Moving Average (ARIMA) and the basic Functional Link based Artificial Neural Network (FLANN) to analyze their performance based on the metrics like Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and R-Squared ($R^2$).

***Subject Classification:*** *68 Computer science.*

***Keywords:*** *Stock market, MATLAB, CHFLANN, LENN, ANN, ARIMA.*

**E-mail:* **niranjanpanda@soa.ac.in (Corresponding Author)**
*† E-mail:* **twinklesingh0101@gmail.com**
*§ E-mail:* **shrabaneeswagatika@soa.ac.in**

## 1. Introduction

Recently, stock market prediction is being paid more attention. The reason can be attributed to the fact that successful prediction of the market trend will help in guiding the investors in a better way [1]. The profits gained by investing and trading in the stock market greatly depend on the predictability. Due to its highly dynamic nature, forecasting the cycle of stock qualities is undoubtedly a difficult process. Therefore, a consistent system that can predict the directions of the stock market will help its users to make informed decisions. It will also be beneficial as the predicted trends will help the regulators of the market in taking corrective measures [1].

The market speculation accepts that the stocks behave arbitrarily and it is difficult to anticipate stock qualities. However, current technical analysis show that most stocks values are reflected in the past records; hence the development patterns are imperative to predict values successfully [2].

Predicting stock prices and forecasting the results by deciding the stock estimation of a monetary trade is our main goal. In the paper, our focus is on implementation of different advanced neural network based algorithms to predict stock prices and finding the error rates by making a comparison between the actual and the predicted result obtained from algorithms. The remaining part of the paper is organized as follows. Section 2 contains a list that represents findings and results from various authors for the related study on the stock market analysis using neural network methods. Section 3 explains the methodology and result analysis of our proposed method. Finally, we have concluded our paper by making a discussion about the future work in Section 4.

## 2. Litereture Review

As we have discussed before stock market analysis is a huge area of research for many researchers. In this section, some of the works carried out by various authors on stock market analysis using neural network methods are discussed and presented.

Niall et. al. [3] used Dow Jones Industrial Average (DJIA) indices from 2 January 1986 to 4 February 2005 as dataset and Artificial Neural Network (ANN) model to predict the stock market and analyzed the result using the metrics Root Mean Square Error (RMSE), DJ Points and Directional Success. Performance comparison is not done to show the novelty of the proposed model in this article. ANN performs outstanding to give a high

Annual Return On Investment (ROI) in comparison to the increase in the DJIA index.

Guresen et. al. [4] collected the dataset for National Association of Securities Dealers Automated Quotations (NASDAQ) indices from 7 October 2008 to 26 June 2009 and techniques like Multilayer Perceptrons (MLP), Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) and Dynamic ANN (DAN) are used to predict the stock market. The performance analysis is done using the matrices Mean Square Error (MSE) and Mean Absolute Deviation (MAD). The result in this article shows that simple MLP seems to be the best and practical ANN architecture but here nothing is said about the prediction in long term trading.

Kara et. al. [5] analyzed the dataset collected for Istanbul Stock Exchange (ISE) National 100 indices from 2 January 1997 to 31 December 2007 using ANN and Support Vector Machine (SVM) and the accuracy of the techniques is computed. In this article, the ANN and SVM models show significant performance in predicting the stock price whereas prediction accuracy of ANN model is higher than SVM model. Also, performance of models may be improved by adjusting the model parameters and by using different input variables.

Niaki and Hoseinzade [6] collected the dataset for US SPDR Standard & Poor's 500 (S&P 500) indices from 1 March 1994 to 30 June 2008 and used the ANN for the ANalysis Of VAriance (ANOVA). The results show that the ANN outperforms the logit model regarding the number of correct forecasts and the buy-and-hold strategy in terms of the obtained profit. But, this proposed ANN is suitable for only certain (bull or bar) market conditions and not for all situations.

Adebiyi et. al. [7] used the NewYork Stock Exchange (NYSE) indices from 17 August 1988 to 25 February 2011 along with the AutoRegressive Integrated Moving Average (ARIMA) and ANN techniques for the stock market prediction. The performance is analyzed using the metric MSE. The study reveals that both ARIMA model and ANN model can achieve good forecast in application.

Patel et. al. [8] collected the National Stock Exchange (NSE) and Bombay Stock Exchange (BSE) Sensex indices from 2003 to 2012 and used the techniques ANN, SVM, Random Forest (RF) and Naive Bayes (NB) to predict the stock market. The metrics like accuracy and F-measure are used to predict the performance of the models. These proposed models focuses on short term prediction and technical indicators are derived based on the period of last 10 days.

Models may not work in case of long term prediction. Also, parameters like stock's quarterly performance, revenue, profit returns, companies' organizational stability etc. is not analyzed.

Sheta et. al. [9] collected the S&P 500 stock market indices from 7 December 2009 to 2 September 2014 and used the ANN, SVM and Multiple Linear Regression (MLR) techniques to analyses the performance in terms of Mean Absolute Error (MAE), RMSE and Correlation Coefficient (R). The result analysis shows that the developed SVM model with RBF kernel model provided good prediction capabilities with respect to the regression and ANN models.

Dash and Dash [10] collected BSE Sensex and S&P 500 indices from 4 January 2010 to 31 December 2014 and tried to propose stock market using their proposed Computational Efficient FLANN (CEFLANN) and Extreme Learning Machine (ELM) models. The authors shown that the proposed models provide superior profit percentage whenever compared to some other known classifiers such as SVM, k-Nearest Neighbors (KNN) and Decision Tree (DT).

Chiang et. al. [11] collected the NASDAQ and S&P 500 stock market indices from 29 January 2004 to 24 June 2011 and analyzed that using Decision Support System (DSS) utilizing Particle Swarm Optimization (PSO) and an ANN. Metrics like Direction Correctness Percentage (DCP), Prediction Accuracy and Simulated Returns are computed to analyze the performance of model proposed.

The proposed model results in a simple network that is allowing a faster system training and greater testing accuracy. PSO helps in faster computation, making the enumeration strategy feasible for real-time application. But, it is highly sensitive to the selection of inputs and parameter settings and limited to the use of technical indicators and patterns as inputs.

Zhong and Enke [12] analyzed the stock market using the models implementing ANN, Principal Component Analysis (PCA), fuzzy Robust PCA (FRPCA) and Kernel-based PCA (KPCA) on the stock S&P 500 stock market indices from 1 June 2003 and 31 May 2013. The MSE and Confusion Matrix are computed and analysis of the result shows that the ANN classifiers combing PCA generate slightly higher risk-adjusted profits than the ones based on combing the ANNs with either FRPCA or KPCA.

Weng et. al. [13] considered the NASDAQ and the DJIA indices from 1 January 2013 to 31 December 2016 for analysis using the ANN, Support Vector Regression, Boosted Regression Tree and Random Forest

Regression models. Metrics like RMSE, MAE, Mean Absolute Percentage Error (MAPE) are used for performance analysis and concluded that ensemble methods outperform single classifiers and predicting the price over multiple time periods provide investors with more information leading to better decision-making. But, no mechanism has been provided by the authors for detecting the model obsolescence.

Hu et. al. [14] considered the S&P 500 and DJIA indices dataset from 1 January 2010 and 16 June 2017. A model implementing Back Propagation Neural Network (BPNN) is used and the Hit Ratio is considered as an attribute to measure its performance. This paper shows that Google trends can help in predicting future financial returns. But, performance comparison is not done to show the novelty of the proposed model.

Naik et. al. [15] suggested analysis of stock market using ANN and Linear Regression models on NSE indices from year 2008 to 2018. The RMSE and MAE metrics are computed and based on that the authors concluded that ANN outperforms compares to SVM by decreasing the error rate in the prediction. On the other hand this work focuses on short term prediction and nothing is said about the prediction in long term trading. Further in [16], various ML and DL algorithms are applied on NSE indices from year 2009 to 2018 to have better accuracy on daily stock prices by the authors.

Zhou et. al. [17] collected the Shanghai Stock Exchange Composite (SSEC), NASDAQ and S&P 500 indices from 3 January 2007 to 30 December 2011 and implemented Empirical Mode Decomposition (EMD), Factorization Machine (FM) and ANN models over it. Metrics such as RMSE, MAE, MAPE, Average Annual Return (AAR), Maximum Drawdown (MD), Sharpe Ratio (SR) are computed for the performance analysis. Authors concluded that this method not only handles the nonlinearity but also the influence among time scales. Irrespective of the short term or long term strategy it only works on the prediction of the index value.

Chopra et. al. [18] used the dataset collected for NSE indices from July 2002 to August 2016 and models like ANN, Levenberg-Marquardt algorithm are proposed to predict the stock market. The performance analysis is done using the matrices Mean Square Error (MSE) and the result in this article shows that the proposed model predicts Indian stock market before and after demonetization most accurately.

## 3. Methodology

This section describes two different state of the art Artificial Neural Network (ANN) methods for the stock analysis named Legendre Neural Network (LENN) and Chebyshev Functional Link based Artificial Neural Network (CHFLANN). The equations representing methods for LENN and CHFLANN are represented below.

### 3.1 *Legendre Neural Network (LENN)*

In LENN, the hidden layers are replaced with various expansion blocks that make the use of Legendre orthogonal functions. These block increases the extent of the input structure by specifying an expansion order, which aids in unraveling the intrinsic nonlinearity among the basic input and its associated output structure. When com-pared to other basic neural networks with hidden layers, this type of network results in less computing complexity and training time [19].

For the generation of higher order Legendre polynomials, the recursive mathematical formula given in equation 1 can be used. This formula is also known as Bonnet's recursion formula [20].

$$(k+1)LP_{k+1}(x) = (2k+1)xP_k(x) - (k)LP_{k-1}(x) \tag{1}$$

where $LP_k$ represents the Legendre polynomial of order $k$, $k$ is the order of expansion and $x$ is the sample input.

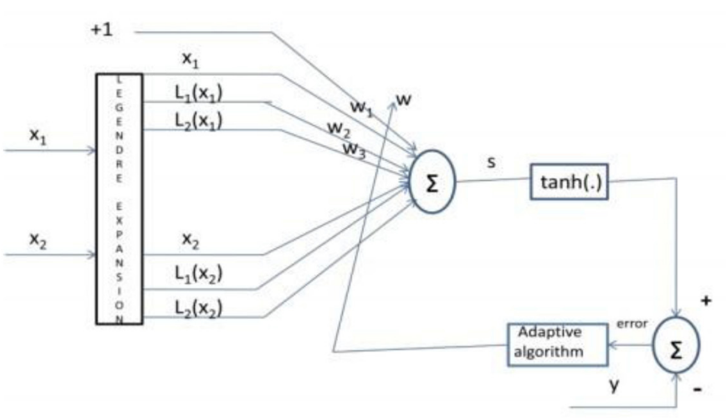


**Figure 1**

**The expansion and adaptive algorithm position for ANN**

### 3.2 *Chebyshev Functional Link based Artificial Neural Network (CHFLANN)*

In CHFLANN, as similar to LENN the hidden layers are replaced with various expansion blocks. But, instead of Legendre orthogonal functions they use Chebyshev orthogonal functions. This single layer neural network shows a better result in terms of computing complexity and training time to other neural networks [21].

The equation 2 gives the recursive formula to generate the higher order Chebyshev polynomials.

$$CP_{k+1}(x) = 2xCP_k(x) - CP_{k-1}(x) \tag{2}$$

where $CP_k$ represents the Chebyshev polynomial of order $k$, $k$ is the order of expansion and $x$ is the sample input.

The learning component in the artificial neural network consisting of a tanh() or log sigmoid function, which is passed with a weighted sum of the extended input vector computed using the formula in equation 3 and equation 4 respectively for LENN and CHFLANN.

$$V = Wt_i LP_i(x) \tag{3}$$

$$V = Wt_i CP_i(x) \tag{4}$$

where $Wt_i$ is the weights of length $l$ generated randomly and $l$ is the length of weights or dimension of pattern after extension using the expansion block and computed as in equation 5.

$$l = k \times n + 1 \tag{5}$$

where $n$ is the dimension of pattern before extension or original dimension of the input pattern.

### 3.3 *Model Design*

In our study, we have collected a dataset of National Stock Exchange (NSE) India and its description is given in the Table 1. The collected dataset consists of 8 attributes: date, time, four stock values (open, high, low, close) and the volume of the stock. Along with all these attributes we have added a class level attribute that has either a value zero or one. The zero presents a negatively open stock value i.e. expects a fall in stock price, where as one indicates a positively open stock value i.e. expects a rise in stock price.

We have verified our model considering the dataset mentioned as a standard dataset. After preprocessing the data in the dataset (dealing with

**Table 1**

**Stock Market Dataset Details for the Proposed Work**

| | |
|---|---|
| Number of rows in original dataset | 768 |
| Number of attributes + a class level attribute | 8 + 1 = 9 |
| Number of rows in training dataset | 512 |
| Number of rows in testing dataset | 256 |

the missing entries), normalization is done for the attribute values to bring them into a specific range. Then the entire dataset is divided into two parts in a ratio two each to one. The first part is called a train set and the second part is called a test set respectively. Then the order of expansion and activation function for the LENN model is chosen and the LENN model is trained using a back-propagation algorithm. The function tanh() used in the learning process is given below in equation 6.

$$Wt_{i,c} = Wt_{i,p} + LR \times E_p(1 - O_p)^2 \times LP_i(x) \tag{6}$$

where $Wt_{i,c}$ is the current weight, $Wt_{i,p}$ is the previous weight, $LR$ is the learning rate, $E_p$ is the error and $O_p$ is the result or output at previous instance.

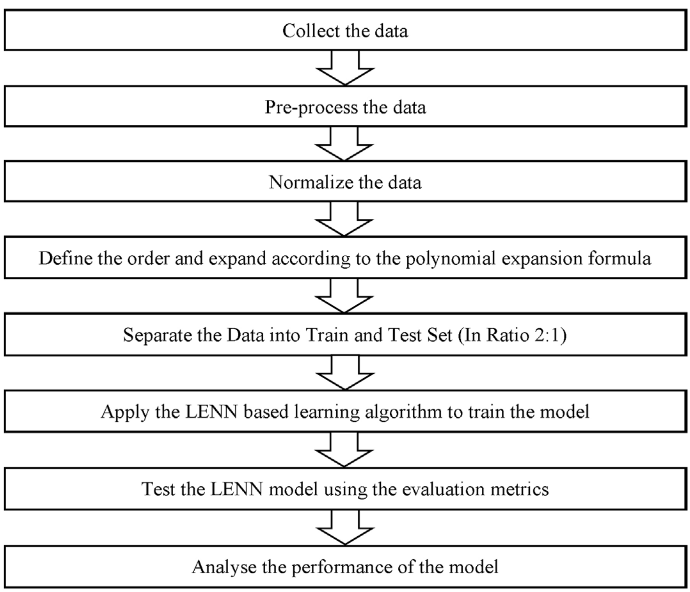


**Figure 2**

**Framework for stock market prediction using LENN**

**Table 2**

**Confusion matrix**

| Actual data | Classified Data | Results |
|---|---|---|
| 1 (Positive) | 1 (Positive) | True Positive |
| 1 (Positive) | 0 (Negative) | False Negative |
| 0 (Negative) | 0 (Negative) | True Negative |
| 0 (Negative) | 1 (Positive) | False Positive |

Accuracy and f-measure of the model can be computed by using the formula given in equation 7 and equation 8.

$$\text{Accuracy} = \text{sensitivity} \times \left(\frac{\text{no of positive values}}{\text{total no of values}}\right) + \text{specificity} \times \left(\frac{\text{no of negative values}}{\text{total no of values}}\right) \quad (7)$$

$$\text{F}-\text{ measure} = \left(\frac{2\times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}\right) \quad (8)$$

where, the terms total no of values, sensitivity, specificity, precision and recall can be computed using the formulas given in equation 9 to equation 13.

$$\text{total no of values} = \text{no of positive values} + \text{no of negative values} \quad (9)$$

$$\text{Specificity} = \frac{\text{no of true positive values}}{\text{no of positive values}} \quad (10)$$

$$\text{Sensitivity} = \frac{\text{no of true negative values}}{\text{no of negative values}} \quad (11)$$

$$\text{Precision} = \frac{\text{no of true positive values}}{\text{no of true positive values+no of false positive values}} \quad (12)$$

$$\text{Recall} = \frac{\text{no of true positive values}}{\text{no of false negetive values}} \quad (13)$$

The performance of the LENN and the CHFLANN method on stock market prediction is tested and represented in Table 3. From this table it is clearly visible that LENN behaves more accurately and also possesses a higher f-measure in comparison to CHEFLANN. Also the order 2 polynomials behave in a more efficient way than the order 3 polynomials in LENN as well as CHFLANN.

**Table 3**

**LENN and CHFLANN classification results**

| Algorithm name | Order of polynomial used in the algorithm | Training set | | Testing set | |
|---|---|---|---|---|---|
| | | Accuracy | F-measure | Accuracy | F-measure |
| LENN | 2 | 0.783412 | 0.90571 | 0.830803 | 0.999211 |
| CHFLANN | | 0.777623 | 0.90324 | 0.824713 | 0.979923 |
| LENN | 3 | 0.779543 | 0.89728 | 0.816476 | 0.989745 |
| CHFLANN | | 0.761776 | 0.88581 | 0.810523 | 0.950563 |

The Mean Square Error (MSE) for the second order LENN and CHFLANN algorithm is computed on the test dataset and presented in the figure 3 and figure 4 respectively.

Here, also the 2$^{nd}$ order LENN performs better with less MSE in comparison to CHEFLANN.

We have also compared the LENN and CHFLANN models to the basic Functional Link Artificial Neural Network (FLANN) model and the statistical analysis model Auto Regressive Integrated Moving Average (ARIMA) to test their predictive capabilities. The Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and R-squared ($R^2$) are computed using the formulas given equation 14 to equation 16 respectively and used as the markers to measure the novelty of the models. We have used the ARIMA (p, d, q) where the p, d, q values are specified explicitly in our model as 1, 0 and 1 respectively.

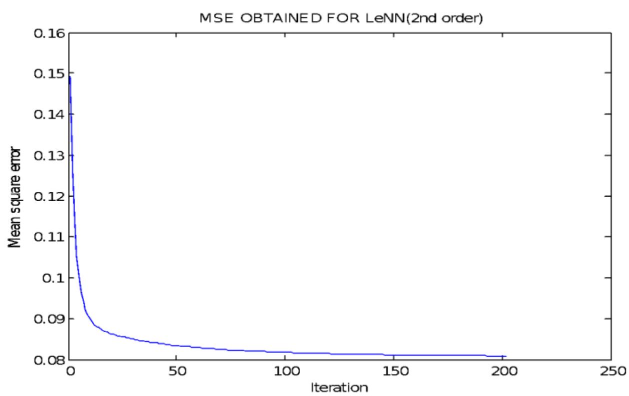


**Figure 3**

**MSE obtained for 2nd order LENN**

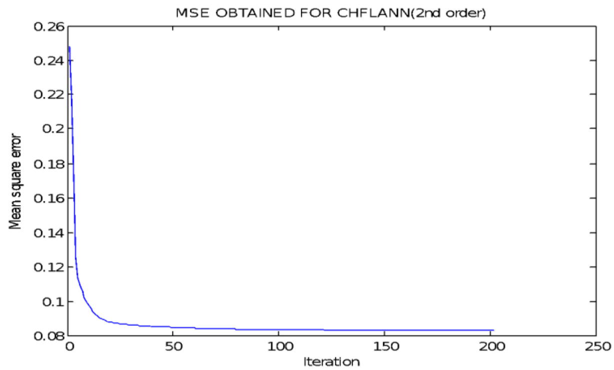


**Figure 4**

**MSE obtained for 2nd order CHFLANN**

$$MSE = \sqrt{\sum_{i=1}^{n} (Y_{PRED_i} - Y_{OBS_i})^2 / n} \tag{14}$$

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{Y_{OBS} - Y_{PRED}}{Y_{OBS}} \right| \tag{15}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (Y_{OBS_i} - \bar{Y})^2}{\sum_{i=1}^{n} (Y_{OBS_i} - Y_{PRED_i})^2} \tag{16}$$

Where $Y_{PRED}$ represents the predicted values, $Y_{OBS}$ represents the observed values and n is the sample size.

The comparison result is summarized in Table 4.

**Table 4**

**Performance matrix**

| Model | RMSE | MAPE | $R^2$ |
|---|---|---|---|
| **ARIMA(1,0,1)** | 1.054711 | 1.326603 | 0.02952 |
| **FLANN** | 0.371667 | 0.16865 | 0.923531 |
| **CHFLANN** | 0.290956 | 0.15863 | 0.944336 |
| **LENN** | 0.283837 | 0.146787 | 0.952917 |

## 4. Conclusion and Future Work

LENN and CHFLANN two types of advanced artificial neural network were discussed in detail in this paper and evaluated on the basis of their accuracy and f-measure for classification and prediction of stock market. In our discussion, we concentrated only on the LENN and CHFLANN and compared their performance, while other type of neural networks may be considered and compared with these neural network performances in future studies. Comparison and evaluation of the LENN and CHFLANN shows that LENN performs better than CHFLANN and second order polynomials in both the cases show better performance than other order polynomials. Ultimately, it can be concluded that LENN that uses the Legendre polynomial of 2$^{nd}$ order performs outstanding than all others. The content of this paper offers the basic idea of classification and prediction stock market using LENN and CHFLANN.

## References

[1] Wang, X. (2021). Rigorous Modeling of Solubility of Acid in Supercritical Carbon Dioxide Using Connectionist approach: Comparison between ANN and density based modeling. Energy Sources, Part A: Recovery, Utilization and Environmental Effects, 1-14.

[2] Tambe, S., Pawar, A., & Yadav, S. K. (2021). Deep fake videos identification using ANN and LSTM. *Journal of Discrete Mathematical Sciences and Cryptography*, 24(8), 2353-2364.

[3] O'Connor, Niall and Michael G. Madden. "A neural network approach to predicting stock exchange movements using external factors." *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, London, 2005.

[4] Guresen, Erkam, Gulgun Kayakutlu and Tugrul U. Daim. "Using artificial neural network models in stock market index prediction." *Expert Systems with Applications* 38.8 (2011): 10389-10397.

[5] Kara, Yakup, Melek Acar Boyacioglu and Ömer Kaan Baykan. "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange." *Expert Systems with Applications* 38.5 (2011): 5311-5319.

[6] Niaki, Seyed Taghi Akhavan and Saeid Hoseinzade. "Forecasting S&P 500 index using artificial neural networks and design of experiments." *Journal of Industrial Engineering International* 9.1 (2013): 1-9.

[7] Adebiyi, Ayodele Ariyo, Aderemi Oluyinka Adewumi and Charles Korede Ayo. "Comparison of ARIMA and artificial neural networks models for stock price prediction." *Journal of Applied Mathematics* 2014 (2014).

[8] Patel, Jigar, et. al. "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques." *Expert Systems with Applications* 42.1 (2015): 259-268.

[9] Sheta, Alaa F., Sara Elsir M. Ahmed and Hossam Faris. "A comparison between regression, artificial neural networks and support vector machines for predicting stock market index." *Soft Computing* 7.8 (2015): 2.

[10] Dash, Rajashree and Pradipta Kishore Dash. "A hybrid stock trading framework integrating technical analysis with machine learning techniques." *The Journal of Finance and Data Science* 2.1 (2016): 42-57.

[11] Chiang, Wen-Chyuan, et. al. "An adaptive stock index trading decision support system." *Expert Systems with Applications* 59 (2016): 195-207.

[12] Zhong, Xiao and David Enke. "Forecasting daily stock market return using dimensionality reduction." *Expert Systems with Applications* 67 (2017): 126-139.

[13] Weng, Bin, et. al. "Predicting short-term stock prices using ensemble methods and online data sources." *Expert Systems with Applications* 112 (2018): 258-273.

[14] Hu, Hongping, et. al. "Predicting the direction of stock markets using optimized neural networks with Google Trends." *Neurocomputing* 285 (2018): 188-195.

[15] Naik, Nagaraj and Biju R. Mohan. "Optimal feature selection of technical indicator and stock prediction using machine learning technique." *International Conference on Emerging Technologies in Computer Engineering*. Springer, Singapore, 2019.

[16] Naik, Nagaraj and Biju R. Mohan. "Stock price movements classification using machine and deep learning techniques-the case

study of indian stock market." *International Conference on Engineering Applications of Neural Networks*. Springer, Cham, 2019.

[17] Zhou, Feng, et. al. "EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction." *Expert Systems with Applications* 115 (2019): 136-151.

[18] Chopra, S., D. Yadav and A. N. Chopra. "Artificial neural networks based indian stock market price prediction: before and after demonetization." *J Swarm Intel Evol Comput* 8.174 (2019): 2.

[19] Zhang, Y., & Wu, L., Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. *Expert Systems with Applications*, 36(5), 8849-8854, (2009)..

[20] Nikfarjam, A., Emadzadeh, E., & Muthaiyah, S., Text mining approaches for stock market prediction. In 2010 The 2nd international conference on computer and automation engineering (ICCAE) (Vol. 4, pp. 256-260). IEEE, (2010, February).

[21] Yang, H., Chan, L., & King, I., Support vector machine regression for volatile stock market prediction. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 391-396). Springer, Berlin, Heidelberg.. (2002, August).

[22] Jeswal, S. K., & Chakraverty, S. (2020). Connectionist based models for solving Diophantine equation. *Journal of Interdisciplinary Mathematics*, 23(4), 825-841.