A Jaya Algorithm Trained Neural Network for Stock Market Prediction

Puspanjali Mohapatra, Reeva Mishra, Tapas Kumar Patra

Abstract: ***This paper demonstrates how the two types of FLANN models (Functional link artificial neural network models) i.e. Chebyshev-FLANN (CFLANN) and Trigonometric-FLANN (TFLANN) are trained using Jaya algorithm to predict the Stock Market Indices. The intention of the current paper is putting forward a contrast between popular training algorithms such as Back Propagation (BP) and Jaya algorithm. The BP and Jaya algorithm trained FLANN models are examined for predicting stock indices for a day and a week ahead. The stock indices BSE500, DJIA and NASDAQ with few technical indicators are taken as inputs in this experimental time series data. The study confirms the superiority of Jaya algorithm trained FLANN models to the traditional BP trained FLANN models. The Mean Square Error (MSE) and Mean Absolute Percentage Error (MAPE) are used for performance evaluation. The simulation study is done using python3 in Anaconda environment.***

Index Terms: Stock Market Prediction, BP, Jaya algorithm, MAPE, MSE.

I. INTRODUCTION

Stock market investment is always considered as an easy and lucrative opportunity to overcome the financial burdens and fundamental living standards of the global society. People become highly optimistic. Often they become ardent expectant of making profit in a shorter duration through stock market investments. Today, more than three billion people of the global society are under privileged, deprived as they are below the poverty line. Now a days the competitions amongst people in business & trading have become very tough because of rapid growth in population of the world population. In this crucial moment investors are trying their best to find out more suitable means of earning money fast with an added focus on stock market investment.

Stock market data is considered as the financial time series data. It is very volatile, noisy, chaotic, non-linear and high dimensional. It gets influenced due to the variation of the global socio-economic- political factors including wars and terrorist attacks. The domain of stock market prediction has gained popularity because of the increasing amount of money being invested in the stock markets. The potential of the stock market is huge. The extraction of patterns and trends from financial time series assists the common men, businessmen and investors in taking a wise decision about the right time to buy, hold or sell stocks.

Thus, statistical analysts have worked on various statistical as well as intelligent soft computing methods. There are many statistical approaches like Auto-regressive Integrated Moving Average (ARIMA), Auto-regressive Moving Average (ARMA) [1], Generalized Auto-regressive Conditional Heteroscedasticity (GARCH) [1], Auto-regressive Conditional Heteroscedasticity (ARCH), and Box-Jenkins approach. The different soft computing and evolutionary approaches are Artificial Neural Networks (ANN), neuro-fuzzy approach. Different optimization techniques are implemented like Differential Evolution (DE) [2], Ant Colony Optimization (ACO) [3] and Particle Swarm Optimization (PSO) [4]. The literature survey shows that different types of ANN's like Radial Basis Function Network (RBFN) [5], Multi-Layer Perceptron (MLP) [6], Local Linear Wavelet Neural Network (LLWNN) [7], FLANN, Time Delay Neural Network (TDNN) [8], Evolutionary Neuro-Fuzzy Neural Network (NN) and various neuro-fuzzy hybrid models have been used.

This paper is prepared in follow sequence: Basic working of FLANN and Back propagation training algorithm are dealt in Section-II. The basics of Jaya algorithm and it's evaluation steps is described in Section III. Further, the application of Jaya algorithm optimization technique is applied to FLANN model in Section-IV. The simulation study is elaborated in section V. The training and testing results are discussed in section VI followed by a brief analysis in section VII. Section-VIII contains the conclusions.

II.BASIC WORKING- PRINCIPLE OF FUNCTIONAL LINK ARTIFICIAL NEURAL NETWORK AND BACK PROPAGATION

A. FLANN Model

The FLANN has less computational complexity as compared to the Multi-layer Perceptron (MLP). MLP has hidden layers which makes it more complex and reduces its convergence rate. So, to reduce the computational burden, we selected the FLANN model. This model, being a single layer ANN, has the ability to form complex decision regions by creating non-linear decision boundaries. In this proposed FLANN model, each input is fed into a functional expansion unit that produces an enhanced representation of the original pattern. In our proposed model, there are six inputs from the time series data and the seventh input is the mean of rest six inputs.

Revised Manuscript Received on 07 August 2018.

Puspanjali Mohapatra, Department of Computer Science and Engineering, International Institute of Information Technology (IIIT), Bhubaneswar (Odisha), India. E-mail: puspanjali@iiit-bh.ac.in

Reeva Mishra, Department of Computer Science and Engineering, International Institute of Information Technology (IIIT), Bhubaneswar (Odisha), India. E-mail: reevamishra208@gmail.com

Tapas Kumar Patra, Department of Electronics and Instrumentation Engineering, College of Engineering and Technology (CET), Bhubaneswar (Odisha), India. E-mail: tkpatra@cet.edu.in

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

The function used in this unit may be any orthogonal function like Trigonometric, Chebyshev, Legendre or Laguerre. It has been observed that the Trigonometric and Chebyshev outperform the other two models [9]. Each input is assigned with a randomly selected weight.

Then, the initially selected random weights [w1, w2, w3, w4….w20, w21] are up-dated using the BP learning algorithm. The proposed TFLANN and CFLANN are as follows:

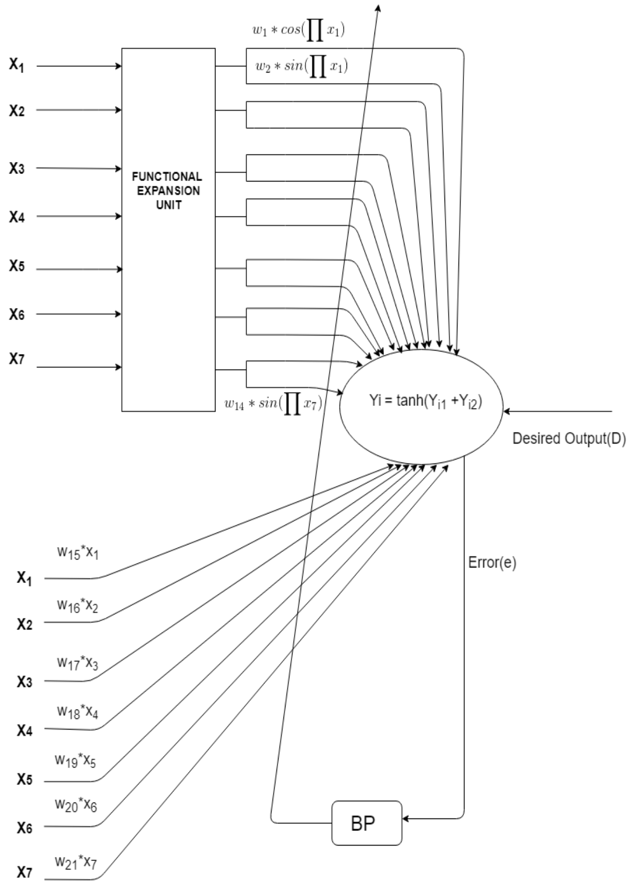

Fig 1: BP Trained TFLANN

The expansion of inputs is done by the Fourier Expansion because it provides a compact representation of the function in the mean square sense [10]. Each input is expanded to have two components as follows:

$$t_1 = cos(\pi x) \quad (5)$$

$$t_2 = sin(\pi x) \quad (6)$$

$$t_1 = x \quad (7)$$

$$t_2 = 2x^2 - 1 \quad (8)$$

$$t3 = 4x^3 - 3x \quad (9)$$

$$t_4 = 8x^4 - 8x^2 + 1 \quad (10)$$

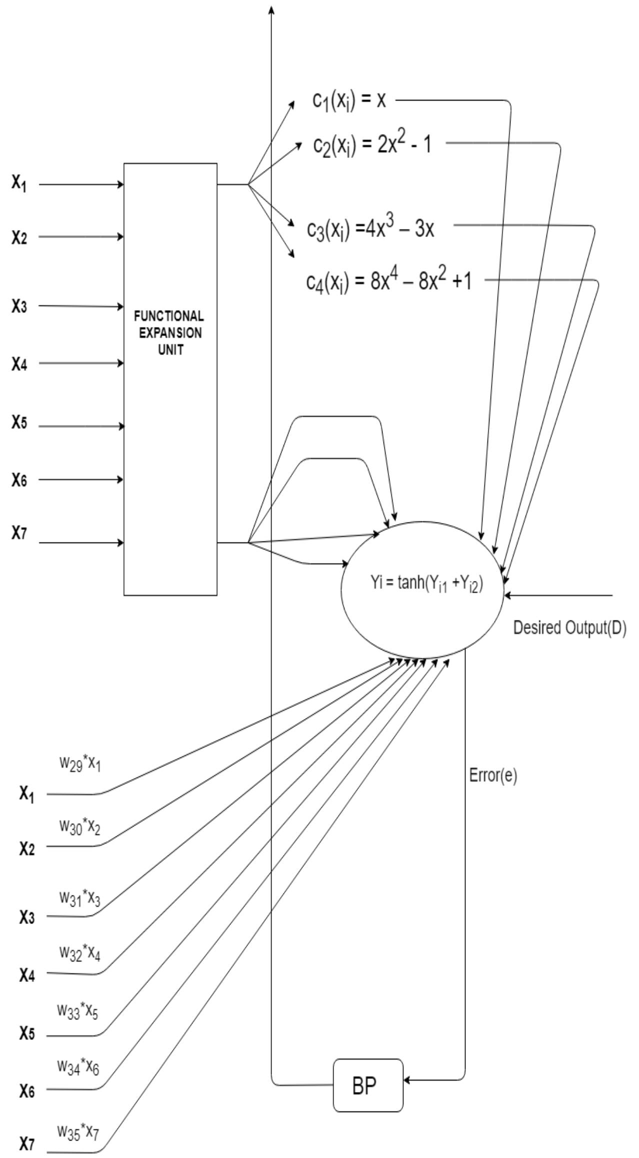

Fig 2: BP trained CFLANN

B. Evaluation

$Y_1 = [w_1 * \cos(\pi x_1) + w_2 * \sin(\pi x_1) + \ldots + w_{13} * \cos(\pi x_7) + w_{14} * \sin(\pi x_7)]$ (1)

$$Y_2 = [w_{15}x_1 + w_{16}x_2 + \ldots + w_{20}x_7 + w_{21}x_7] \quad (2)$$

Finally, the output 'Y' is represented as the sum of Y_1 an Y_2. The output is passed through an activation function, which in this case is a tan-hyperbolic function.

$$Y = tanh\,(Y_1 + Y_2) \quad (3)$$

The final output is then compared with our Desired Output (D) to produce an error.

$$Error(e) = Y - D \quad (4)$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

III. BASICS OF JAYA ALGORITHM

According to R. Venkata Rao [12], Jaya algorithm has always been a very powerful optimization algorithm that solves constrained as well as unconstrained problems [11]. The main concept behind this algorithm is the solution for a given problem should avoid the worst solution and move towards the best solution.

Unlike other algorithms, the Jaya algorithm requires a very few common parameters. It doesn't require algorithm-specific control parameters. [12]

Population-based heuristic algorithms are mainly of two types:

Evolutionary Algorithm (EA): Examples of popular evolutionary algorithms are Genetic Algorithm (GA), Evolution Strategy (ES), Differential Evolution (DE) [14], etc.

Swarm Intelligence (SI) based algorithm: Examples of SI based algorithms are Particle Swarm Optimization (PSO) [13] [16], Ant Colony Optimization (ACO), Ant Bee Colony (ABC), Fire Fly (FF) [12], etc.

Besides, Some other algorithms work on different principles of the nature like Harmony Search (HS), Whale Optimization Algorithm (WOA), etc. In case of HS algorithm, the EA and SI-based algorithms have their own algorithm-specific controlling parameters like number of generations, population size, and harmony memory considering rule (hmcr) and pitch adjusting rate (par). If the tuning of these algorithm-specific parameters are improper, they may result in increased computational error or may give poor results. For this reason, R. Venkata Rao had suggested Teaching Learning Based Optimization (TLBO) that has two common controlling parameters i.e. the number of generations and population size. Thus, this optimization technique has gained much popularity among the re-searchers. [12]

However, the TLBO algorithms possess two phases: one is Teaching Phase and the other is Learning Phase. Thus, Jaya optimization technique was suggested by R. Venkata Rao that has only one phase and is much easier to apply [15].

IV. JAYA OPTIMIZED FLANN MODEL

In the Jaya optimized FLANN model, the random weights are updated by the Jaya algorithm. In the above FLANN model, there are 7 inputs (i.e. candidates) each corresponding to 3 components (or design variables).

In the first iteration, the objective function's value is calculated, from which the best and worst values are selected. Based on these values, the corresponding design variables are updated and thus, the value of the objective function is also found out. Now, the best among the new and old values of the objective function is selected and their corresponding design variables are used for evaluation in the e next iteration.

The basic working of Jaya can be demonstrated by thee flowchart given below:

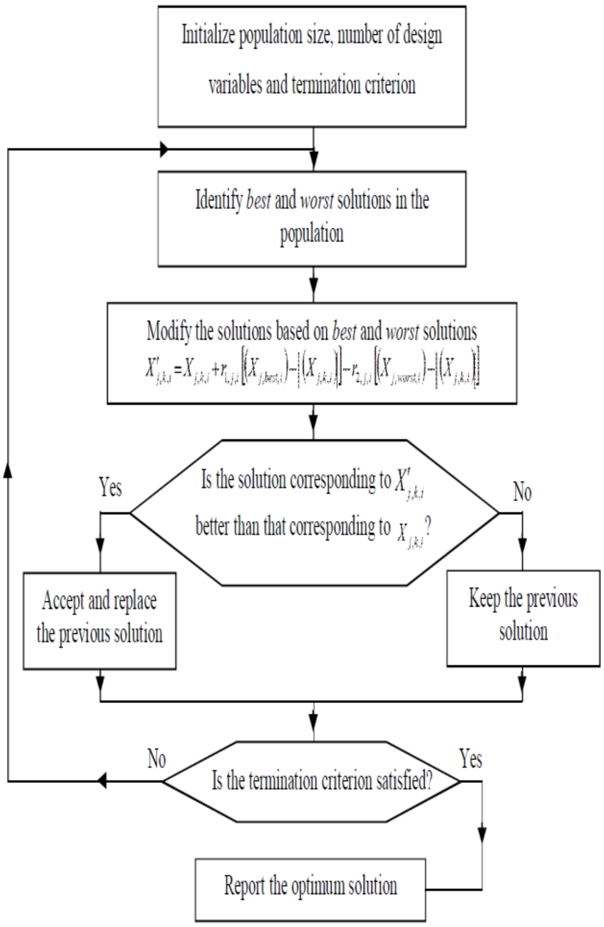

Fig 3: Flowchart of Jaya Model

Here, $X'_{j,k,i}$ *is the value of* j^{th} *variable for the* k^{th} *candidate during* i^{th} *iteration.*

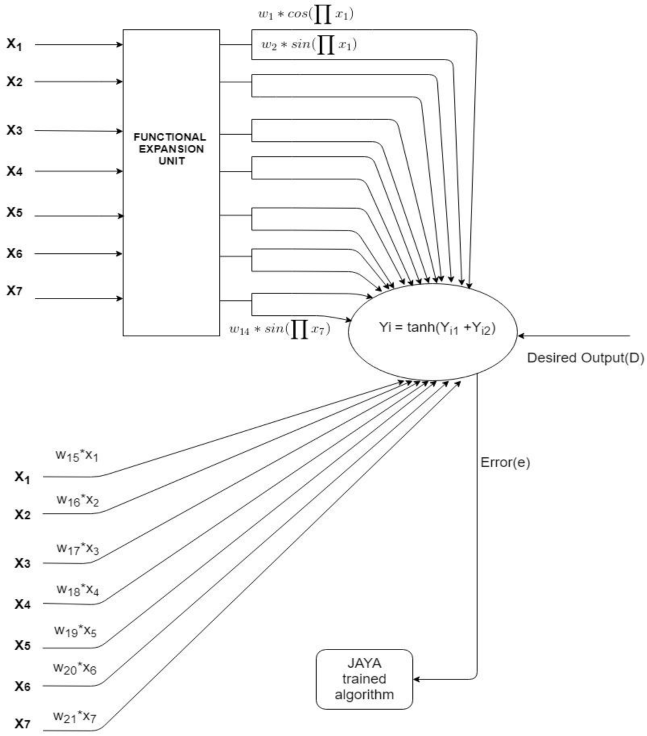

Fig 4: Jaya Algorithm Trained TFLANN Model

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

V. SIMULATION STUDY

The daily closing price of various stock market data i.e. BSE500, DJIA and NASDAQ, obtained from www.yahoo.finance.com and www.bseindia.com are considered as the experimental data in this experiment. All the inputs are normalized within the range [0, 1] using the following formula:

$$X_{norm} = \frac{X_{orig} - X_{min}}{X_{max} - X_{min}}$$

Where X_{norm} is normalized input value, X_{orig} is current closing price, X_{min} is the daily minimum price of stock data and X_{max} is daily maximum price of stock data.

The datasets used for the simulation is depicted as follows:

Table 1: Stock Data Sets Used

STOCK DATA SETS	TOTAL SAMPLE	DATA RANGE	TRAINING SAMPLE	TESTING SAMPLE
BSE500	19th Feb 2003 19th Feb 2018	3700	1300	250
DJIA	19th Feb 2003 19th Feb 2018	2500	1300	250
NASDAQ	19th Feb 2003 19th Feb 2018	3200	1300	250

The technical indicators used in the process are as given below:

Table 2: Technical Indicators and Their Formulae

TECHNICAL INDICATORS	FORMULA
Simple Moving Average(SMA)	$\frac{1}{N}\sum_{i=1}^{N} x_i$ N = No. of Days. x_i = today's price
Price Rate of Change (PROC)	$\frac{\text{Today's close} - \text{Close X-periods ago}}{\text{Close X-periods ago}} * 100$

VI. RESULTS DISCUSSION

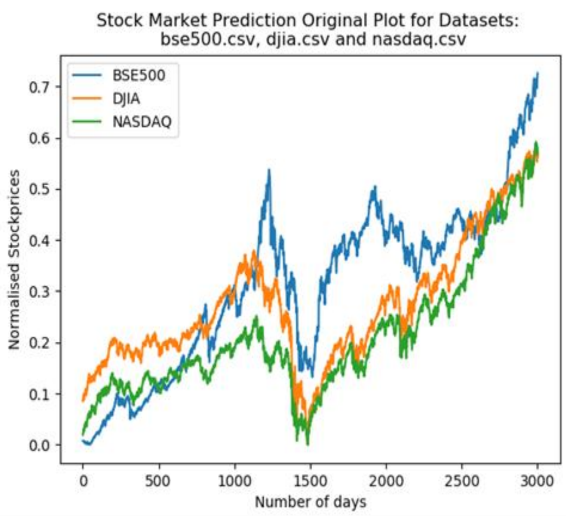

Fig 5: Original datasets: BSE500, DJIA and NASDAQ

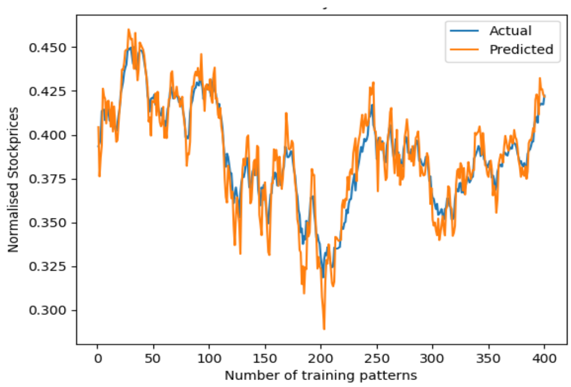

Fig 6: Actual vs Predicted of BP trained CFLANN during testing and training respectively for 1 week ahead

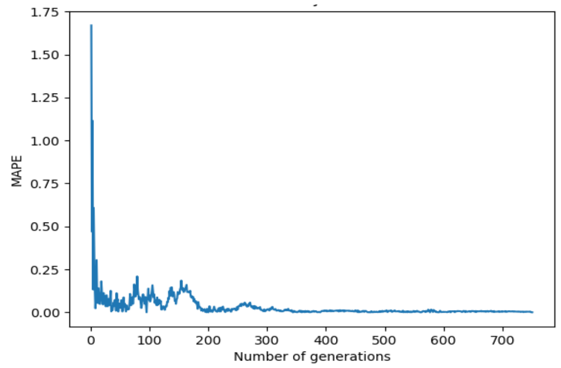

Fig 7: MSE of BP trained CFLANN during training for 1 week

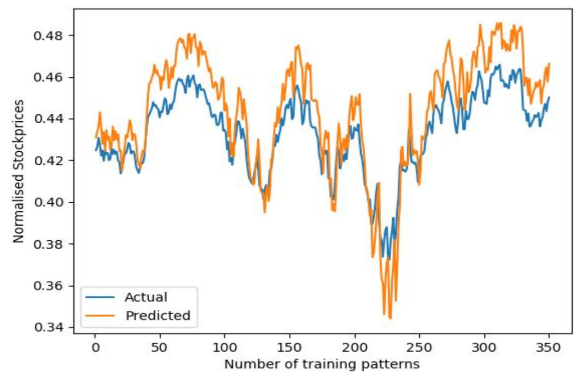

Fig 8: Actual vs Predicted of BP trained TFLANN during testing for 1 week ahead

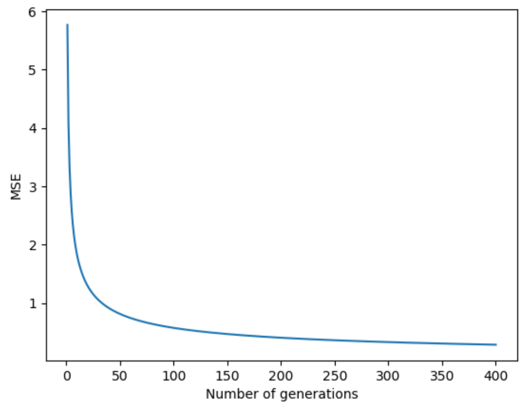

Fig 9: MSE of BP trained TFLANN during training for 1week ahead

Table3: MSE and MAPE During Training and Testing for BP Trained CFLANN and TFLANN

STOCK DATA	FLANN MODEL	PREDICTION	MSE (train)	MAPE in % (train)	MSE (test)	MAPE in % (test)
BSE500	TFLANN	1-day	0.7277	1.956	0.036	1.642
		1-week	0.8524	6.77	0.045	4.9
	CFLANN	1-day	0.2885	1.02	0.006	0.415
		1-week	0.2925	1.75	0.39	0.424
DJIA	TFLANN	1-day	0.579	5.06	0.11	1.71
		1-week	0.9824	4.27	0.486	2.7
	CFLANN	1-day	0.2433	2.35	0.0143	1.09
		1-week	0.2484	2.09	0.115	1.31
NASDAQ	TFLANN	1-day	0.6171	9.33	0.025	3.15
		1-week	0.8326	11.13	0.082	3.96
	CFLANN	1-day	0.2737	3.37	0.047	1.78
		1-week	0.2778	6.9	0.061	2.59

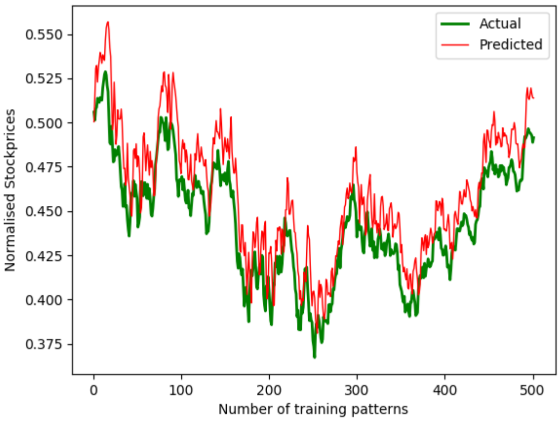

Fig 10: Actual vs Predicted of Jaya Trained CFLANN During Testing for 1 Week Ahead

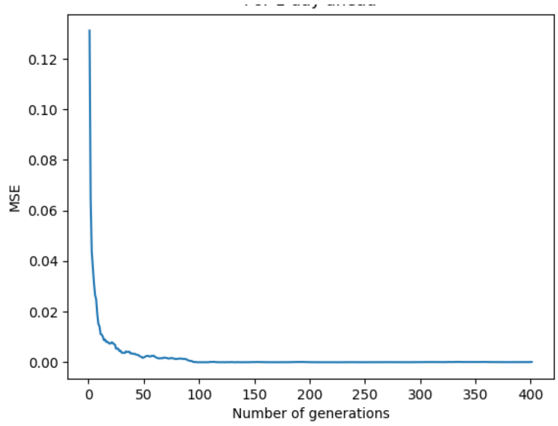

Fig 11: MSE of Jaya Trained CFLANN During Training for 1 Week Ahead

VII. ANALYSIS

The TFLANN and CFLANN models were trained with Back Propagation and Jaya successfully. A remarkable difference was observed in the values of MSE in Jaya as compared to BP. It almost got reduced by 90%.

VIII. CONCLUSION

The prediction of an accurate stock market is a very difficult task. However, our currently proposed FLANN model upon trained with BP gave good results as per the recorded MSE and MAPE during training and testing, in prediction for one-day ahead and one-week ahead.

The Jaya optimized FLANN has proved its superiority over BP in terms of MSE. Further, the performance of Jaya algorithm should be compared with DE, PSO and Harmony Search algorithms.

REFERENCES

1. Tang, Him, K. C. Chun, and Lei Xu. "Finite mixture of ARMA-GARCH model for stock price prediction." *Proceedings of the Third International Workshop on Computational Intelligence in Economics and Finance (CIEF'2003), North Carolina, USA*. 2003.
2. Qin, A. Kai, Vicky Ling Huang, and Ponnuthurai N. Suganthan. "Differential evolution algorithm with strategy adaptation for global numerical optimization." *IEEE transactions on Evolutionary Computation* 13.2 (2009): 398-417.
3. Dorigo, Marco, and Mauro Birattari. "Ant colony optimization." *Encyclopedia of machine learning*. Springer, Boston, MA, 2011. 36-39.
4. Kennedy, James. "Particle swarm optimization." *Encyclopedia of machine learning*. Springer US, 2011. 760-766.
5. Jang, J-SR, and C-T. Sun. "Functional equivalence between radial basis function networks and fuzzy inference systems." *IEEE transactions on Neural Networks* 4.1 (1993): 156-159.
6. 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.
7. Chen, Yuehui, Xiaohui Dong, and Yaou Zhao. "Stock index modeling using EDA based local linear wavelet neural network." *Neural Networks and Brain, 2005. ICNN&B'05. International Conference on*. Vol. 3. IEEE, 2005.
8. Kim, Hyun-jung, and Kyung-shik Shin. "A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets." *Applied Soft Computing*7.2 (2007): 569-576.
9. Mohapatra C. and Ron S., "Flann based model to predict stock price movements of stock indices," Ph.D. dissertation, 2007.
10. Bebarta D. K., Biswal B., and Dash P. K., "Polynomial based functional link artificial recurrent neural network adaptive system for predicting indian stocks," International Journal of Computational Intelligence Systems, vol. 8, no. 6, pp. 1004{1016, 2015.
11. Angeline P. J., \Evolutionary optimization versus particle swarm optimization: Phi-losophy and performance di erences," in International Conference on Evolutionary Programming. Springer, 1998, pp. 601{610.
12. Jothi G., Inbarani H. H., Azar A. T., and Devi K. R. ,"Rough set theory with jaya optimization for acute lymphoblastic leukemia classification," Neural Computing and Applications, pp. 1{20, 2018.
13. Mohapatra, Puspanjali, Alok Raj, and Tapas Kumar Patra. "Indian stock market prediction using differential evolutionary neural network model." *International Journal of Electronics Communication and Computer Technology (IJECCT) Volume*2 (2012).
14. Rao R., "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," International Journal of Industrial En-gineering Computations, vol. 7, no. 1, pp. 19{34, 2016.
15. Rao R., More K., Taler J., and Oclon P.,"Dimensional optimization of a micro-channel heat sink using jaya algorithm," Applied Thermal Engineering, vol. 103, pp. 572{582, 2016.
16. Benala T. R., Chinnababu K., Mall R., and Dehuri S., "A particle swarm optimized functional link artificial neural network (pso- ann) in software cost estimation," in Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA). Springer, 2013, pp. 59-66.