Improving the Generalization Performance of the Back Propagation Neural Network using Projection based Learning Algorithm

K. Jayanthi
Department of Computer Science
Thiruvalluvar Government Arts College, Rasipuram, Tamilnadu, India
jaysureshlaya@gmail.com

P. Suresh
Department of Computer Science
Salem Sowdeswari College(Autonomous)
Salem, Tamilnadu, India
sur_bh0071@rediffmail.com

K. Velusamy
Department of Computer Science
Sri Vasavi College, Erode
Tamilnadu, India
kvelusamy85@gmail.com

Abstract- **The Projection Based Learning algorithm is used to optimize the connection weights with Back Propagation Neural Network is proposed in this paper. The proposed model referred as PBL-BPNN that employ for predicting the future closing price of the stock related to real time stock market data. The stock price prediction is a demanding research problem in the financial sector that has received a significant amount of attention in machine learning to make a decision about buying or selling the product on the right time to improve the profit of the client. The process of the training and the testing are applied to real time stock market data. In experimental results show that the proposed Projection Based Learning-Back Propagation Neural Network algorithms have been produced better results compared with standard Back Propagation Neural Network in term of statistical accuracy and trading efficiency of the stock price.**

Keyword: Feed Forward Neural Network, Back Propagation Neural Network, Projection Based Learning, Stock Price Prediction, Learning Algorithms

1. INTRODUCTION

Stock market forecasting is one of the most significant problems in economic markets that make the investor worry about their investment in stock market (Bisoi & Dash, 2014). The stock price prediction has a vital role for the investment to brokers and individual investors, and many research works has been carried out by reliable methods for the prediction of the stock price. A stock market is an open market for companies to increase the money, which helps companies to buy or sell their shares. The price of shares depends upon the demand and supplies of shares. The process of buying and selling of shares is called trading and only the listed Companies are allowed to carry out the trading.

The stock market prediction is the process of trying to determine the future stock price of a company from their historical data. The artificial neural network (Sivanandam & Deepa, 2006) is an information processing paradigm that aims to take off the human capability to adapt to altering circumstances, that is encouraged by human brain, which is interconnected with lots of neurons that are transmitted signals among connected neurons.

The time series analysis is a very decisive problems and financial time series analysis data has high complexity, nonlinearity and fail to capture the discontinuities while applying in usual statistical method. Hence, The Artificial Neural Network (Sermpinis, Theofilatos, Karathanasopoulos, Georgopoulos, & Dunis, 2013) is a complex machine learning technique which provides enough learning capacity and is more likely to capture the complex non-linear models. The main objective of this paper is to applied fast learning algorithms for improving the accuracy of the problem such as projection based learning algorithms (Babu, Savitha, & Suresh, 2012) which is capable to overcome the negative aspect of the other learning algorithms like Gradient Decent algorithm (Rumelhart, Hinton, & Williams, 1988), Kalman Filter(Watanabe & Tzafestas, 1990) and Extended Kalman Filter(Singhal & Wu, 1988).

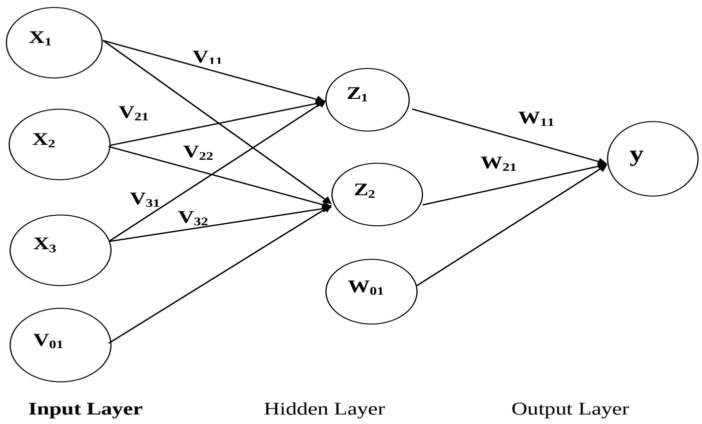

Figure 1: Architecture of Artificial Neural Networks

The time series analysis is a very decisive problems and financial time series analysis data has high complexity, nonlinearity and fail to capture the discontinuities while applying in usual statistical method. Hence, The Artificial Neural Network (Sermpinis, et al., 2013) is a complex machine learning technique which provides enough learning capacity and is more likely to capture the complex non-linear models. The main objective of this paper is to applied fast learning algorithms for improving the accuracy of the problem such as projection based learning algorithms (Babu, et al., 2012) which is capable to overcome the negative aspect of the other learning algorithms like Gradient Decent algorithm (Rumelhart, et al., 1988), Kalman Filter(Watanabe & Tzafestas, 1990) and Extended Kalman Filter(Singhal & Wu, 1988).

In order to overcome the drawback of the BPNN , hybridization of the Cuckoo Search (CS) and Levenberg-Marquardt algorithm to train neural network for XOR data sets. The proposed CSLM (Nawi, Khan, & Rehman, 2013) method reduces the error and improve the performance by escaping from local minima. In nonlinear dynamical systems(Saptoro, 2012), the Extended Kalman Filter based BPNN is improved between 2.45 to 21.48 % in term of training and testing is improved 8.35 and 29.15 %. The proposed Wavelet De-noising-based Back Propagation (WDBP) neural network (Wang, Wang, Zhang, & Guo, 2011) was applied for forecasting the stock prices to improve the accuracy. The neural network initial weights for tuning with LM algorithm by using genetic algorithm(Asadi, Hadavandi, Mehmanpazir, & Nakhostin, 2012) which is applied for predicting stock price. The proposed hybrid PSO–BP algorithm (Zhang, Zhang, Lok, & Lyu, 2007) improves the convergent speed and accuracy than Adaptive Particle swarm optimization algorithm (APSOA) and BPNN. The Kalman filtering (Watanabe & Tzafestas, 1990) compare to standard Back Propagation Neural Network approach has been estimate the connection weights of a network with a linear or nonlinear inspection. The Marquardt algorithm (Hagan & Menhaj, 1994) for nonlinear least squares is proposed with back propagation algorithm for training feed forward neural networks. Recently, many research paper applied for projection based learning algorithms for updates network weights that finds the optimal output based on its energy function that is defined by the hinge loss error for classification and time series analysis data. The Meta-cognitive Radial Basis Function Network (McRBFN) and its Projection Based Learning (PBL) algorithm (Babu, et al., 2012) (G Sateesh Babu & Sundaram Suresh, 2013) (Giduthuri Sateesh Babu & Sundaram Suresh, 2013) has applied to classification problems for fine-tuning the its network weights. The modified Meta-Cognitive Radial Basis Function Network (McRBFN+) and its Projection Based Learning (PBL) (Subramanian, Suresh, & Cheng, 2015) algorithm is applied for classification problems.

According to the literature review, the following points are mentioned below,

- Several researcher has been implemented various approach for predicting future price of the stock market. The appropriate learning algorithm in neural network for improving accuracy makes fast convergence with least iteration and reduces the computing time.
- Learning algorithm helps to achieve better generalization of the proposed approaches in neural network.

- In stock price prediction, reducing the error between actual prices of the stock and predicting price of the stock.

From the literature, the Projection Based Learning algorithm helps to decrease the computational effort, locates the most favorable network weights and improves the accuracy of the networks. Hence, the proposed projection based learning for Back Propagation Neural Networks is to improve the accuracy of the networks and finding optimal weights which is applied for prediction the next day closing price of the stock. The presented approach is applied for predicting stock price for well known stock datasets including State Bank of India, HCL and Infosys.

The proposed learning algorithm performs outstandingly well for predicting the future price of the stock in the next day in term of statistical efficiency and trading efficiency. It give the impression that its adaptability and flexibility allows it to outperform in our forecasting problems which match up to with the more familiar algorithms is back propagation neural network.

2. LEARNING ALGORITHMS

The process of learning (Baruník & Malinska, 2016) is defined as the adjustment of weights using a learning algorithm. The main goal of the learning process is to minimize the sum of the prediction errors for all training examples. The most used learning algorithms such as gradient based learning (Bisoi & Dash, 2014) algorithms, Kalman Filter Learning (Watanabe & Tzafestas, 1990) Algorithm and Extended Kalman Filter (Singhal & Wu, 1988) algorithms are the following drawbacks has mentioned in the literature including Exhibit slow convergence rate, and fall down into local minima. Hence, to keep away from aforementioned, two defects are solved by the proposed projection based learning (PBL) algorithms.

The activation functions control the amplitude of the output and consequently determine the final desired signals. In this paper used hyperbolic tangent sigmoid function for both input and hidden layer of the neural networks. The neural network with learning rate (Jammazi & Aloui, 2012) is control the step side of the weight. The learning process is very slow while using least learning rate in the network and the learning process is very much change in the error function while using high learning rate.

3. BACK PROPAGATION NEURAL NETWORK

The Back Propagation Neural Network (Rajasekaran & Pai, 2003) (BPNN) is a well known multi-layer feed forward training model that has been proposed by Rumelhart and McClelland (Rumelhart, et al., 1988) and the figure 1 depicts the architecture of back propagation neural network.

They are effective in many applications of fields such as pattern recognition, risk evaluation and self-adaptive control, time series analysis, etc., and BPNN has four parts as below,

a. Initialization
b. Feed Forward
c. Back Propagation Error
d. Updating

3.1. Initialization

The normalization is done to ensure that the inputs to the BPNN, the input values are range between 0 and 1 and It is important to note that (Sheela & Deepa, 2013) the values for scaling used within 0.1 to 0.9 for feed forward network. The normalization of datasets is done by equation (1)

$$X_i^{'} = \left(\frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \right) \left(X_{\max}^{'} - X_{\min}^{'} \right) + X_{\min}^{'} \tag{1}$$

Where $X_{\max}^{'}, X_{\min}^{'}$ be the minimum and maximum target value and X_i , $X_{\max}, X_{\min}$ are actual input data.

Weight is information by the neural net to work out a problem which is the most part in neural network that initialization is some small values by randomly

- Input layer neurons are connected to the hidden neurons by means of their weights that are represented by $[V_{ij}]$. The V is weight values of ith input neurons and j^{th} hidden neurons.
- Hidden layer neurons are connected to the output layer neurons with their weight that is represented by $[W_{jk}]$
- Bias improves the performance of the neural network; its value is 1. Bias value for the hidden neurons is represented by $[V_{0j}]$ and for the output neurons represented by $[W_{0k}]$

3.2. Feed Forward

An input values are $X = (x_1, x_2, ..., x_n)$ transmits to each of the hidden units $Z = (z_1, z_2, ..., z_k)$ after multiples with weight and those values are transferred to hidden unit. Each hidden units calculate activation function and it sends its value to the each output unit. The output unit calculates the activation function to form the response of the net.

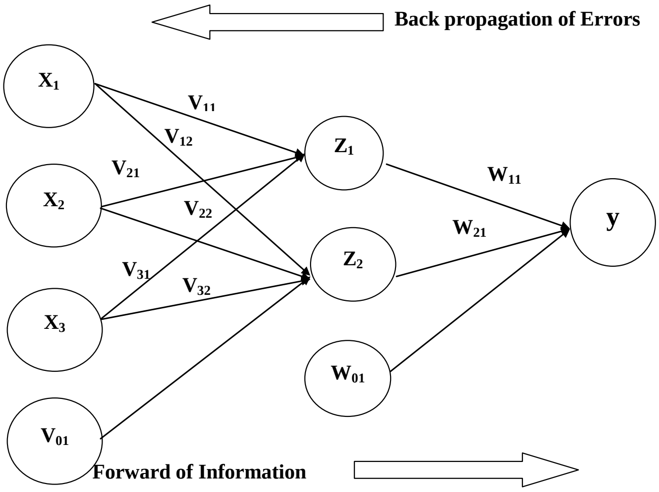

Figure 2 : Architecture of BPNN

Calculate the hidden neuron by using

$$z_{-inj} = V_{oj} + \sum_{i=1}^{n} x_i v_{ij} \tag{2}$$

Calculate output neurons by using

$$y_{-ink} = W_{ok} + \sum_{j=1}^{p} z_i w_{jk} \tag{3}$$

3.3. Back propagation of Errors

Each output unit compares its computed actual activation results with its target.

$$\delta_k = (t_k - y_k) f(y_{-ink}) \tag{4}$$

Error information is calculated by using following formula based on Equation 4.

$$\delta_j = \delta_{-inj} f(z_{-inj}) \tag{5}$$

3.4. Weight Updating using Projection Based Learning

In conventional Back Propagation Neural Networks, weight updating process done by gradient decent algorithms which is learning rate is slow and fall into local optima. Hence, this paper proposed Projection Based Learning algorithms to overwrite the shortcoming of the conventional Back Propagation Neural Networks. An objective of proposed Projection Based Learning algorithm (Babu, et al., 2012; G Sateesh Babu & Sundaram Suresh, 2013) is to minimize the error of the networks and finds the optimal network weights.

The error function is defined for i^{th} sample as follows,

$$J_i = \sum_{j=1}^{n} (y_j^i - \sum_{k=1}^{K} w_{kj} Z_k^i)^2 \tag{6}$$

The whole error function of the model is calculated for t samples as follows,

$$\mathrm{J(W)} = \frac{1}{2}\sum_{i=1}^{t} \mathrm{J_i} = \frac{1}{2}\sum_{i=1}^{t} \sum_{j=1}^{n} \left(y_j^i - \sum_{k=1}^{K} w_{kj} Z_k^i\right)^2 \tag{7}$$

Where, h_k^i an output of hidden neuron with k node and i^{th} is the training samples. The optimal weight represented $W^* \in \Re^{K\times n}$ are expected total error reaches its minimum,

$$W^* \arg \min_{\Re^{K\times n}} J(W) \tag{8}$$

The optimal W^* is the equivalent to the minimum of the error function. That is $(J(W^*))$ obtained by equates the first order partial derivative of $J(W)$ with respect to the weight to zero.

$$\frac{\partial J(W)}{\partial w_{pj}} = 0, p = 1, \dots . K: \ j = 1, \dots ., n \tag{9}$$

Equating the first partial derivative to zero and reorganize as follows,

$$\sum_{k=1}^{K} \sum_{i=1}^{t} h_k^i h_p^i = \sum_{i=1}^{t} h_p^i y_j^i \tag{10}$$

The above equation correspond to in matrix form as follows,

$$AW = B \tag{11}$$

The projection matrix $A \in \Re^{K\times k}$ is given by

$$a_{kp} \sum_{i=1}^{t} h_k^i h_p^i, p = 1, \dots . K: \ k = 1, \dots ., p \tag{12}$$

And the output matrix $B \in \Re^{K\times k}$ is

$$b_{pj} \sum_{i=1}^{t} h_p^i h_j^i, p = 1, \dots . K: \ j = 1, \dots ., n \tag{13}$$

Equation (11) gives the set of $K \times n$ linear equations with $K \times n$ unknown weights W. If $\left(\partial^2 J/\partial W_{pj}{}^2\right) > 0$. The output weights is given in term of the second derivative of the error function as follows,

$$(J) \frac{\partial^2 J(W)}{\partial W_{pj}{}^2} = \sum_{i=1}^{t} h_p^i h_p^i = \sum_{i=1}^{t} \left|h_p^i\right| > 0 \tag{14}$$

If second derivative of the error function $J(W)$ is positive, the following clarification can be made form of equation (12).

i. J is a convex function.
ii. The achieved weight $W*$ is the weight corresponding to the minima of the error function of J.

The solution for the system of equations in (11) can be determined as follows:

$$W^* = A^{-1}B \tag{15}$$

The most favorable network weights are projected with its minimum total error as follows,

$$W^* = \underset{W \in R^{K\times n}}{\arg \min} \tag{16}$$

The closing stages of the BPNN, values are converted to de-normalized values using normalization method.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

4.1. Stock Price Prediction

The stock price prediction is one of the major issues and extremely crucial in stock market. Many researchers have been focused for finding the stock market forecasting. There are many factors affecting stock market such as economic condition, political events and investor's sentiment etc. This paper discussed about the daily closing price of the well known dataset are used for training and testing that related to their historical datasets which is obtained from the finance section of Yahoo.

4.2. Performance Evaluations

In order to measure the performance of the proposed Projection Based Learning-Back Propagation Neural Network model is compared with conventional Back Propagation Neural Network. The proposed model is applied to predicting the future closing price of the stock that is related to State Bank of India, HCL, and Infosys. Performance of the proposed model is evaluated based on various statistical methods. The following performance measurements are used to calculate the prediction accuracy of the prediction algorithms including Mean Square Error, Root Mean Square Error (RMSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2 \tag{17}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2} \tag{18}$$

4.3. Data Collections

The real time stock datasets are collected from yahoo finance section (" https://in.finance.yahoo.com/,") including State Bank of India, HCL, Infosys. In experimental, each dataset contains 4042 working days of data point which covers the period from January 3, 2000 to February 2016, 75% of data points are used for training and 25% are used for testing the collected total of 4042 pairs are an observation. Datasets contains open price of the day, low price of day, average price of the day, and high price of day which are used as input variable for the networks, and closing price of day is used as target price for the networks.

To assessment the forecasting performance of our proposed method, the experiment is carried out for three time series datasets including State Bank of India, HCL, and Infosys. It also evaluates the performance of proposed method with that of conventional Back Propagation Neural Networs. Table 1 is given a picture of performance analysis of the proposed Projection Based Learning for Back Propagation Neural Network in numerical representation.

4.4. Discussions

The learning algorithm is very important in order to make high-quality generalization performance for stock price prediction in neural networks. The Projection Based Learning algorithm, finding optimal weight parameter by solving first order system linear equation is used to update the synaptic weights of the neural networks. The proposed Projection Based Learning algorithms for Back Propagation Neural Network successfully applied for predicting next day closing price of the stock for real-time well known datasets including SBIN, HCL, and Infosys.

Table 1: Performance Analysis of the Stock Market Datasets

METHODS	SBIN		HCL		INFOSYS	
	MSE	RMSE	MSE	RMSE	MSE	RMSE
BPNN	0.0332	0.1833	0.0405	0.2012	0.0670	0.2588
PBL_BPNN	0.0295	0.1702	0.0241	0.1552	0.0621	0.2492

The Mean Square Error of the proposed model of the PBL_BPNN is an improved 0.03 % and its Root Mean Square is improved by 0.0131 % for SBIN. In term of HCL datasets, the mean square error of the proposed model is improved by 0.0164 and its RMSE is improved by 0.0460% and MSE of the proposed model is improved 0.0049 % and its RMSE is improved by 0.0096 % for Infosys datasets.

The sample datasets with its results of the used architectures of neural network are depicted Table 2, Table 3 and Table 4 for SBIN, HCL, and Infosys datasets respectively and their different are also mentioned in same. The good performance of the architectures its result is mentioned with bold letter for every day of the datasets. Hence, In case of SBIN dataset, the performance of the proposed neural networks is 87.5 % increased. In case of HCL dataset, 90 % is increased and Infosys 85% is increased than Standard BPNN.

The performance of the proposed PBL_BPNN is evaluated by statistical parameter including MSE, RMSE. The statistical performance analysis clearly indicates the proposed model which forms the superior performance, fast learning capability, quick convergence and stability. In experimental results, the proposed algorithm put into operation by using MATLAB 2015a.

5. CONCLUSION

The present study investigated the time series analysis that is particularly focused on stock price prediction and analyzed by supervised multilayer feed forward neural network using Projection Based Learning-Back Propagation Neural Network. The performance of the PBL-BPN is evaluated using statistical measurement and applied for well-known datasets. The statistical comparison of the proposed model clearly indicates the superior performance.

REFERENCES

Asadi, S., Hadavandi, E., Mehmanpazir, F., & Nakhostin, M. M. (2012). Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. Knowledge-Based Systems, 35, 245-258.

Babu, G. S., Savitha, R., & Suresh, S. (2012). A projection based learning in meta-cognitive radial basis function network for classification problems. In The 2012 international joint conference on neural networks (IJCNN): IEEE, 1-8.

Babu, G. S., & Suresh, S. (2013). Meta-cognitive RBF network and its projection based learning algorithm for classification problems. Applied Soft Computing, 13, 654-666.

Babu, G. S., & Suresh, S. (2013). Sequential projection-based metacognitive learning in a radial basis function network for classification problems. IEEE transactions on neural networks and learning systems, 24, 194-206.

Baruník, J., & Malinska, B. (2016). Forecasting the term structure of crude oil futures prices with neural networks. Applied Energy, 164, 366-379.

Bisoi, R., & Dash, P. K. (2014). A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter. Applied Soft Computing, 19, 41-56.

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. IEEE transactions on Neural Networks, 5, 989-993.

https://in.finance.yahoo.com/.

Jammazi, R., & Aloui, C. (2012). Crude oil price forecasting: Experimental evidence from wavelet decomposition and neural network modeling. Energy Economics, 34, 828-841.

Nawi, N. M., Khan, A., & Rehman, M. (2013). A new Levenberg Marquardt based back propagation algorithm trained with cuckoo search. Procedia Technology, 11, 18-23.

Rajasekaran, S., & Pai, G. V. (2003). Neural networks, fuzzy logic and genetic algorithm: synthesis and applications (with cd): PHI Learning Pvt. Ltd.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. Cognitive modeling, 5, 1.

Saptoro, A. (2012). Extended and unscented kalman filters for artificial neural network modelling of a nonlinear dynamical system. Theoretical Foundations of Chemical Engineering, 46, 274-278.

Sermpinis, G., Theofilatos, K., Karathanasopoulos, A., Georgopoulos, E. F., & Dunis, C. (2013). Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and particle swarm optimization. European Journal of Operational Research, 225, 528-540.

Sheela, K. G., & Deepa, S. (2013). Review on methods to fix number of hidden neurons in neural networks. Mathematical Problems in Engineering, 2013.

Singhal, S., & Wu, L. (1988). Training Multilayer Perceptrons with the Extende Kalman Algorithm. In NIPS 133-140.

Sivanandam, S., & Deepa, S. (2006). Introduction to neural networks using Matlab 6.0: Tata McGraw-Hill Education.

Subramanian, K., Suresh, S., & Cheng, R. P. (2015). A modified projection based learning algorithm for a meta-cognitive radial basis function classifier. In Cognitive Computing and Information Processing (CCIP), 2015 International Conference on: IEEE, 1-6.

Wang, J.-Z., Wang, J.-J., Zhang, Z.-G., & Guo, S.-P. (2011). Forecasting stock indices with back propagation neural network. Expert Systems with Applications, 38, 14346-14355.

Watanabe, K., & Tzafestas, S. G. (1990). Learning algorithms for neural networks with the Kalman filters. Journal of Intelligent and Robotic Systems, 3, 305-319.

Zhang, J.-R., Zhang, J., Lok, T.-M., & Lyu, M. R. (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Applied Mathematics and Computation, 185, 1026-1037.

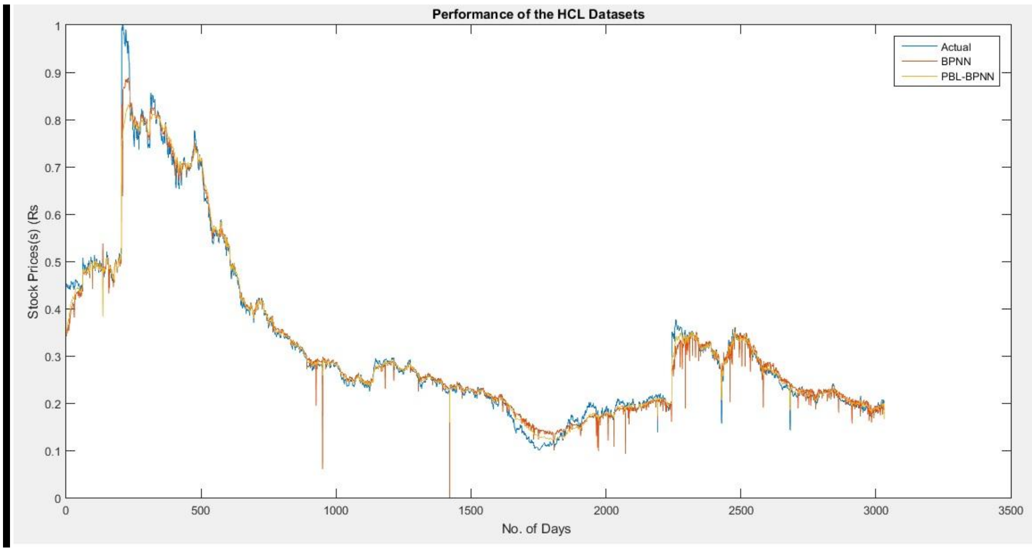

Figure 3 : Performance of the Proposed Model for HCL Datasets

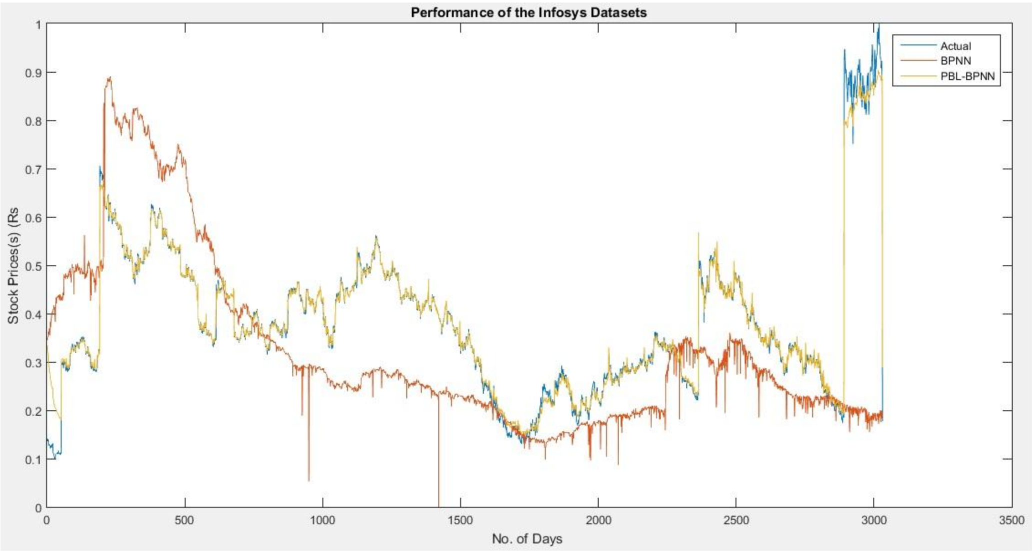

Figure 4 : Performance of the proposed model for Infosys dataset

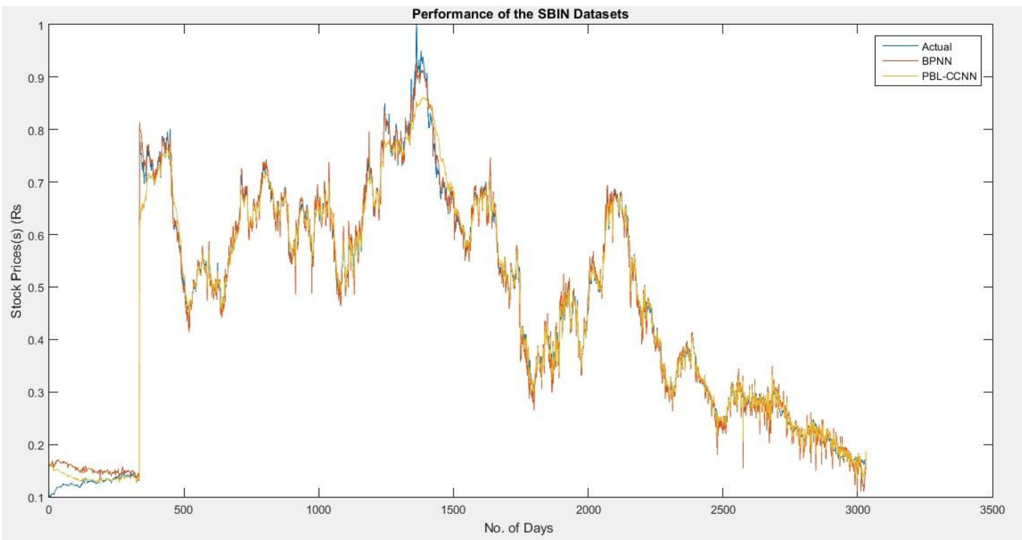

Figure 5: Performance of the proposed model for State Bank of India

Table 2 : Sample Results of the SBIN Datasets

S.No	Date	SBIN (State Bank of India)				
		Actual	**BPN**	**PBL_BPN**	**Differences (Actual-BPN)**	**Differences (Actual-PBL_BPN)**
1.	26-05-16	184.15	187.37	**187.12**	3.22	2.97
2.	27-05-16	195.90	194.47	**194.14**	1.43	1.76
3.	30-05-16	198.85	198.00	**198.20**	0.85	0.65
4.	31-05-16	204.95	203.53	**204.35**	1.42	0.6
5.	01-06-16	198.25	198.37	**198.75**	0.12	0.5
6.	02-06-16	200.55	199.46	**199.87**	1.09	0.68
7.	03-06-16	196.60	**197.42**	197.51	0.82	0.91
8.	06-06-16	198.90	199.20	**199.15**	0.3	0.25
9.	07-06-16	210.15	210.42	**210.30**	0.27	0.15
10.	08-06-16	210.70	211.25	**211.26**	0.55	0.56
11.	09-06-16	209.95	210.56	**210.51**	0.61	0.56
12.	10-06-16	205.95	206.53	**206.62**	0.58	0.67
13.	13-06-16	201.80	201.59	**201.91**	0.21	0.11
14.	14-06-16	207.70	**207.77**	207.88	0.07	0.18
15.	15-06-16	216.05	216.53	**216.47**	0.48	0.42
16.	16-06-16	216.00	216.30	**216.13**	0.3	0.13
17.	17-06-16	213.45	**213.69**	213.93	0.24	0.48
18.	20-06-16	215.90	216.17	**216.18**	0.27	0.28
19.	21-06-16	214.00	214.46	**214.35**	0.46	0.35
20.	22-06-16	212.90	213.42	**213.24**	0.52	0.34
21.	23-06-16	217.55	217.69	**217.63**	0.14	0.08
22.	24-06-16	211.25	210.92	**211.34**	0.33	0.09
23.	27-06-16	217.15	217.39	**217.35**	0.24	0.2
24.	28-06-16	215.90	216.01	**216.08**	0.11	0.18
25.	29-06-16	217.20	217.41	**217.32**	0.21	0.12
26.	30-06-16	218.80	218.74	**218.82**	0.06	0.02
27.	01-07-16	219.60	219.49	**219.57**	0.11	0.03
28.	04-07-16	223.00	222.63	**222.78**	0.37	0.22
29.	05-07-16	223.50	223.23	**223.29**	0.27	0.21
30.	07-07-16	220.05	219.94	**220.00**	0.11	0.05
31.	08-07-16	218.30	**218.34**	218.41	0.04	0.11
32.	11-07-16	224.70	224.51	**224.46**	0.19	0.24
33.	12-07-16	226.70	226.48	**226.57**	0.22	0.13
34.	13-07-16	227.70	227.37	**227.45**	0.33	0.25
35.	14-07-16	232.00	231.77	**231.95**	0.23	0.05
36.	15-07-16	231.50	**230.92**	230.85	0.58	0.65
37.	18-07-16	228.75	228.36	**228.42**	0.39	0.33
38.	19-07-16	229.60	229.48	**229.25**	0.12	0.35
39.	20-07-16	231.15	231.03	**230.78**	0.12	0.37
40.	21-07-16	225.65	225.28	**225.40**	0.37	0.25

Table 3 : Sample Results of the HCL Datasets

S.No	Date	HCL (HCL Technology)				
		Actual	**BPN**	**PBL_BPN**	**Differences (Actual-BPN)**	**Differences (Actual-PBL_BPN)**
1.	01-01-16	845.85	846.62	**846.32**	0.77	0.47
2.	04-01-16	845.95	846.39	**846.16**	0.44	0.21
3.	05-01-16	842.80	843.20	**842.83**	0.4	0.03
4.	06-01-16	841.40	842.03	**841.53**	0.63	0.13
5.	07-01-16	825.05	824.41	**824.61**	0.64	0.44
6.	08-01-16	828.35	827.93	**828.04**	0.42	0.31
7.	11-01-16	814.55	813.59	**814.07**	0.96	0.48

8.	12-01-16	810.05	809.05	**809.99**	1	0.06
9.	13-01-16	826.40	826.14	**826.15**	0.26	0.25
10.	14-01-16	828.85	829.04	**828.88**	0.19	0.03
11.	15-01-16	838.05	**837.88**	837.82	0.17	0.23
12.	18-01-16	842.35	843.02	**842.91**	0.67	0.56
13.	19-01-16	838.25	837.64	**837.94**	0.61	0.31
14.	20-01-16	841.90	841.75	**841.83**	0.15	0.07
15.	21-01-16	835.25	835.88	**835.54**	0.63	0.29
16.	22-01-16	839.25	839.82	**839.61**	0.57	0.36
17.	25-01-16	838.70	839.86	**839.38**	1.16	0.68
18.	27-01-16	838.70	838.91	**838.80**	0.21	0.1
19.	28-01-16	844.50	**844.59**	844.86	0.09	0.36
20.	29-01-16	865.75	862.35	**864.05**	3.4	1.7
21.	01-02-16	880.30	871.37	**873.34**	8.93	6.96
22.	02-02-16	872.65	868.40	**869.70**	4.25	2.95
23.	03-02-16	852.01	853.88	**853.79**	1.87	1.78
24.	04-02-16	867.35	864.23	**865.78**	3.12	1.57
25.	05-02-16	869.65	866.22	**867.31**	3.43	2.34
26.	08-02-16	849.75	851.89	**851.56**	2.14	1.81
27.	09-02-16	810.45	809.53	**810.01**	0.92	0.44
28.	10-02-16	827.40	827.80	**827.59**	0.4	0.19
29.	11-02-16	808.00	807.20	**807.87**	0.8	0.13
30.	12-02-16	798.25	797.70	**798.32**	0.55	0.07
31.	15-02-16	827.10	827.63	**827.24**	0.53	0.14
32.	16-02-16	824.30	824.79	**824.47**	0.49	0.17
33.	17-02-16	824.80	825.08	**824.89**	0.28	0.09
34.	18-02-16	843.95	**844.47**	844.52	0.52	0.57
35.	19-02-16	851.55	**851.44**	851.83	0.11	0.28
36.	22-02-16	845.95	846.91	**846.58**	0.96	0.63
37.	23-02-16	828.95	829.81	**829.24**	0.86	0.29
38.	24-02-16	811.05	810.19	**810.50**	0.86	0.55
39.	25-02-16	808.75	808.33	**808.57**	0.42	0.18
40.	01-01-16	824.65	824.92	**824.60**	0.27	0.05

Table 4: Sample Results of the Infosys Datasets

S.No	Date	INFY (Infosys)				
		Actual	BPN	PBL_BPN	Differences (Actual-BPN)	Differences (Actual-PBL_BPN)
1.	01-01-16	1105.25	**1104.93**	1106.56	0.32	1.31
2.	04-01-16	1078.90	1079.99	**1079.52**	1.09	0.62
3.	05-01-16	1074.05	1074.37	**1074.18**	0.32	0.13
4.	06-01-16	1069.35	1069.45	**1069.22**	0.1	0.13
5.	07-01-16	1050.80	1051.11	**1050.66**	0.31	0.14
6.	08-01-16	1063.30	1063.26	**1063.33**	0.04	0.03
7.	11-01-16	1055.70	1056.21	**1055.61**	0.51	0.09
8.	12-01-16	1049.95	1047.68	**1048.98**	2.27	0.97
9.	13-01-16	1083.40	**1083.59**	1084.14	0.19	0.74
10.	14-01-16	1133.00	**1133.55**	1133.88	0.55	0.88
11.	15-01-16	1139.90	1137.14	**1139.16**	2.76	0.74
12.	18-01-16	1131.90	1129.65	**1131.26**	2.25	0.64
13.	19-01-16	1139.65	1137.94	**1139.54**	1.71	0.11
14.	20-01-16	1121.25	1119.64	**1120.58**	1.61	0.67
15.	21-01-16	1137.05	1135.31	**1137.81**	1.74	0.76
16.	22-01-16	1136.25	1134.20	**1136.23**	2.05	0.02
17.	25-01-16	1137.65	1136.07	**1138.05**	1.58	0.4
18.	27-01-16	1137.65	1138.83	**1138.05**	1.18	0.4
19.	28-01-16	1129.60	1128.64	**1129.65**	0.96	0.05

20.	29-01-16	1164.90	**1164.95**	1164.59	0.05	0.31
21.	01-02-16	1173.35	1173.80	**1172.90**	0.45	0.45
22.	02-02-16	1175.55	1176.15	**1175.06**	0.6	0.49
23.	03-02-16	1157.70	1156.67	**1157.33**	1.03	0.37
24.	04-02-16	1179.75	1181.43	**1179.49**	1.68	0.26
25.	05-02-16	1174.70	1175.74	**1174.62**	1.04	0.08
26.	08-02-16	1150.80	1148.99	**1150.56**	1.81	0.24
27.	09-02-16	1107.65	**1107.08**	1108.45	0.57	0.8
28.	10-02-16	1108.55	1109.54	**1108.86**	0.99	0.31
29.	11-02-16	1078.65	1080.44	**1079.15**	1.79	0.5
30.	12-02-16	1084.65	1085.87	**1085.28**	1.22	0.63
31.	15-02-16	1091.75	1093.14	**1092.40**	1.39	0.65
32.	16-02-16	1079.15	1080.93	**1079.58**	1.78	0.43
33.	17-02-16	1097.85	1099.01	**1098.63**	1.16	0.78
34.	18-02-16	1123.40	**1123.45**	1124.00	0.05	0.6
35.	19-02-16	1127.25	1126.36	**1127.38**	0.89	0.13
36.	22-02-16	1126.65	1125.08	**1126.81**	1.57	0.16
37.	23-02-16	1123.90	1122.68	**1123.92**	1.22	0.02
38.	24-02-16	1125.35	**1125.05**	1124.84	0.3	0.51
39.	25-02-16	1110.30	1109.70	**1110.25**	0.6	0.05
40.	01-01-16	1121.15	1120.88	**1121.12**	0.27	0.03