IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Predicting Stock Prices in National Stock Exchange of India using Principal Component Analysis and Neural Networks

G. G. Rajput[1] , Bhagwat H. Kaulwar[2*]

[1]Department of Computer Science, Rani Channamma University, Belagavi, India
[2]Deparment of Computer Science, Solapur University, Solapur, India

[*]*Corresponding Author:* bhagwat_kaulwar@yahoo.com, *Tel.: +91-99221-71399*

***Abstract*—** The prediction of a particular stock price serves as recommendation system for investors. Most of stock prediction studies focus on using macroeconomic indicators to train the prediction model. Due to difficulty in obtaining this data on daily basis, we directly employ the daily prices data to train the model for predicting the stock price. This study focuses on identifying significant inputs among the financial indicators using Principal Component Analysis to construct a model for prediction. A Multilayer Feed-Forward Nonlinear Autoregressive with External (Exogenous) Input (NARX) network is trained and used to predict closing price of a share listed in National Stock Exchange (NSE). Financial indicators of State Bank of India (SBI) are used as case study to train & test the proposed system. NARX network designed for year 2012 and tested for year 2013.

***Keywords*—** Artificial Neural Network (ANN), Nonlinear Autoregressive with External Input (NARX), Principal Component Analysis (PCA), stock price prediction

I. INTRODUCTION

In share market, prediction of stock prices direction is a key for better trading strategy and decision-making by the investors [1]. People invest and trade in share market to gain premium on their capital. Before taking investment decision, investor has to perform analysis of company so that his/her decision doesn't go wrong. There are two types of analysis in share market. They are fundamental analysis and technical analysis. The fundamental analysis is used for studying overall financial and operational health of company under observation. It requires knowledge of balance sheet. Such analysis is used for long term investments. On the other hands technical analysis doesn't required financial knowledge of company. In this analysis trends are given more importance. This type of analysis is used for short term or medium term investments. The fundamental analysis requires macro-economic data and technical analysis requires daily transaction data of particular share.

The emergence of machine learning and artificial intelligence techniques has made it possible to derive mathematical models in prediction of stock price direction. Popular methods used for study include artificial neural networks (ANNs), Bayesian networks, and support vector machine (SVM). Amongst them, ANNs has been extensively studied by several researchers in the stock price forecasting in the past decades [2]. The study in the area of Neural Network (NN) has proved that NN can be used to predict INDEX or stock price. The daily transaction data can be used to train the NN. The trained NN then can be used to predict future value of Index or share.

It is observed that, the daily trading data of a particular share is generally correlated. If NN is trained with such data, model will try to identify similar pattern in data during its use. The Principal Component Analysis (PCA) is one of the methods to analyse the data and convert it to linearly uncorrelated data [3]. PCA also can be used to reduce dimensionality of input data. In this paper, we present analysis of daily trading data using PCA and select significant principal components as inputs to NN for predicting stock price.

The paper is organised as follows, Section I contains introduction about the problem under study, Section II contains literature survey of related work, Section III contains objective and essential steps of PCA, Section IV contains architecture of Neural Network and NARX model, Section V A contains information of data used along with pre-processing methods and PCA, Section V B contains details of training NN, Section V C contains measures used to validate model performance, Section VI discusses experimental results, Section VII concludes the research work with highlighting results of prediction model .

II. LITERATURE SURVEY

Because irregular fluctuations occur in stock market, it is very difficult to model its behavior. Stock market can be

considered as non-linear deterministic system [4]. But ANN has ability to discover nonlinear relationship in input data set without a priori assumption of the knowledge of relation between input and output (Hagen et. al. 1996). Many researchers have proposed different types of models for prediction of an Index or stock. The neural network model is one of the popular models for classification and prediction. Some researchers have worked on PCA along with prediction models like NN or Support Vector Machine.

Mbeledogu N. N. et. al. (2012) conducted a study to exhibit PCA's advantage of quantifying the importance of each dimension for describing the variability of a data set. For the study they had obtained daily stock dataset of 300 records from public database that contained Nigerian stock exchange data. The dataset contained 19 attributes. Initially they ignored attributes with constant values which reduced number of attributes to 14. They performed PCA on the data. From PCA they conclude that the entire dataset can be represented by first 9 Eigen-values. Thus original dataset was reduced from 19 attributes to 9 attributes [5].

Yanshan Wang (2014) conducted an experiment to forecast the directions of daily movements of the stock price indices and of individual stocks in KOrea Stock Prices Index 200 (KOSPI 200) and Hang Seng Index (HSI). He used lagged daily prices for indices and the overall constituents. From PCA he observed that first component has 70 % contribution for the KOSPI and first ten components have over 70 % for the HSI. So first one component for KOSPI, and first ten components for HSI were used along with internal and external financial factors in SVM to predict index direction [2].

Marijana Zekic-Susac et. al. (2013) constructed NN with PCA to classify students according to their entrepreneurial intentions. They tested four modeling strategies in order to find most efficient model. Their research revealed benefits from the combination of the PCA and ANN in modeling entrepreneurial intentions. They found that out of 4 models ANN model with PCA saved the time of data collection and the training time. The original data was of 443 records of students with 94 attributes. The PCA revealed that 22 components were representing the original data. After providing these components to NN, they found that NN with PCA was most efficient among other three models [6].

In this paper, we perform PCA on the data for feature reduction and train the NN network for stock price prediction. PCA is performed on 12 attributes of the stock data of SBI to obtain 4 attributes representing a linear combination of these 12 attributes (dimensionality reduction).

III. PRINCIPAL COMPONENT ANALYSIS

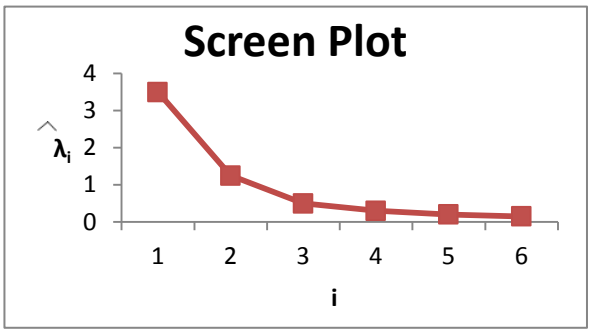
Principal Component Analysis (PCA) is concerned with explaining the variance-covariance structure of a set of variables through a few linear combinations of these variables. The original data are thus projected onto a much smaller space, resulting in dimensionality reduction. Also, PCA often reveals relationships that were not previously suspected and thereby allows interpretations that would not ordinarily result. The basic procedure is as follows.

i. The input data are normalized, so that each attribute falls within the same range.
ii. For the n-dimensional normalized data, PCA searches for k n-dimensional orthogonal vectors that can best be used to represent the data, here $k \leq n$. These vectors are referred to as the principal components. The input data are a linear combination of the principal components.
iii. The principal components are sorted in order of decreasing "significance" or strength; that is first component represents the most variance among the data, the second component represents the next highest variance, and so on.
iv. The p components are required to reproduce the total system variability. However, much of this variability can be accounted for by small number, k, of the principal components. So, weaker components, that is, those with low variance are eliminated. The strongest principal components, k, represent a good approximation of the original data. The k principal components replace the initial p variables, and the original data set consisting of n measurements on p variables is reduced to a data set consisting of n measurements of k principal components [7].

A decision of selecting first k components as stronger components can be taken with the help of screen plot. With the eigenvalues ordered from largest to smallest, a screen plot is a plot of $\hat{\lambda}_i$ versus i – the magnitude of an eigenvalues versus its number. The number of components is taken to be the point at which the remaining eigenvalues are relatively small and all about the same size. Screen plot can be similar to Plot 1 shown below.

PLOT 1

SCREEN PLOT $\hat{\lambda}_i$ VS I – EIGENVALUES MAGNITUDE VS NUMBER



In the screen plot i.e. Plot 1 an elbow occurs at about *i*=3. This indicates that eigenvalues after $\hat{\lambda}_2$ are all relatively small and about the same size. In such case, two or three principal components effectively summarize the total sample variance.

IV. ARTIFICIAL NEURAL NETWORK

A neural network (NN) is a set of connected input/output units in which each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples. Neural network learning is also referred to as connectionist learning due to the connections between units. The most commonly used NN, known as multilayer feed-forward neural network, is shown in **figure 1**.

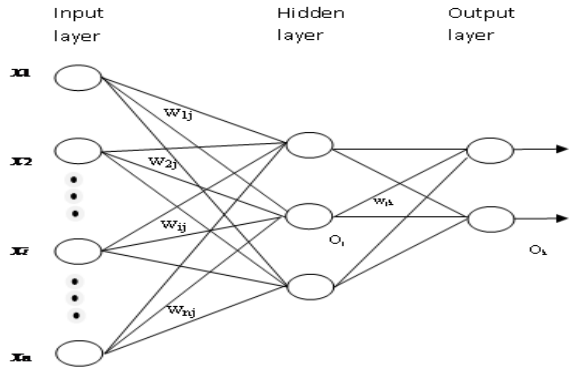


Figure 1 : Neural Network Architecture

The NN shown in fig. 1 consists of three layers i.e. input layer, one or more hidden layer and an output layer. The inputs are given to model simultaneously into the units in input layer. These inputs pass through input layer and then weighted and given to hidden layer. The output of one hidden layer can be given to another hidden layer if any and so on. The weighted outputs of last hidden layer are given to output layer, which gives prediction by model for given tuples. The back-propagation algorithm performs learning on a multilayer feed-forward neural network. It iteratively learns a set of weights for prediction of the class label of tuples. In the proposed work, we considered a special class of ANN called Nonlinear Autoregressive with External (Exogenous) Input (NARX). NARX networks are recurrent neural networks and are well suited for modelling nonlinear systems and specially time series. NARX networks with Gradient descent adaptive back-propagation learning algorithm are considered as better not only because learning is more effective in NARX networks but also they converge much faster and generalize better than other networks [8, 9].

V. DATA AND METHODOLOGY

A. Data pre-Processing and Principal Component selection

From the literature it is observed that technical parameters are more useful to predict share price in short term than macroeconomic indicators [10]. In this paper, we have used daily transaction data like open price, high price, low price, volume, moving averages etc. The details are presented below.

The company under study is State Bank of India (SBI) listed in NSE of India. The historical daily trading data of SBI, for the years 2012 and 2013, is taken from National Stock Exchange website [11].

The daily transaction data included the close price of SBI on previous trading day, open price of SBI for current trading day, high price of share on trading day, low price of share on trading day, average price of share on trading day, total transactions that trading a share on trading day, total turnover in rupees for trading share on trading day. Using these values moving averages of shares like 10DMA, 20DMA, 50DMA, 100DMA and 200DMA were computed. There are 7 attributes and 5 moving averages. Thus, data represents 12 attributes. The closing price of the share on trading day is used as output of the NN.

Through the process of obtaining Principal Components we observed that out of the 12 variables, total transactions that trading a share on trading day was linearly correlated with total turnover in rupees. So we ignored the variable total transactions that trading a share on trading day. The values associated with certain variables out of remaining 11 variables were found to be in different ranges. Few were of range in hundreds and few of range in lacks. Hence, to avoid impact of high scaling data on prediction, Z-Score

normalization was performed on input data using the following formula.

$$v'_i = \frac{v_i - \overline{A}}{\sigma_A} \quad \textbf{(1)}$$

Where $\overline{A}$ and σ_A are respectively, the mean and standard deviation, for the attribute A. The normalization of data gives all attributes an equal weight. Using the neural network back-propagation algorithm for classification learning, normalizing the input values for each attribute measured in the training tuples will help speed up the learning phase [12].

Principal component analysis is a multivariate technique for examining relationships among several quantitative variables and detecting linear relationships. You can use principal components to reduce the number of variables in regression, clustering, and so on [13]. PCA was performed on normalized data in order to obtain Principal Components using Statistical Analysis System (SAS) software. SAS is a software suite developed by SAS Institute for Advanced analytics multivariate analyses, business intelligence, data management, and predictive analytics [14]. Table 1 gives Eigen values and cumulative Eigen values generated using PRINCOMP procedure of SAS software. The Eigenvectors generated after PRINCOMP procedure are presented in Table 2.

TABLE 1
EIGENVALUES BY PRINCOMP PROCEDURE OF SAS

Eigenvalues of the Covariance Matrix				
	Eigenvalue	**Difference**	**Proportion**	**Cumulative**
1	8.01249554	6.52060141	0.7284	0.7284
2	1.49189413	0.59998447	0.1356	0.8640
3	0.89190966	0.48198592	0.0811	0.9451
4	0.40992374	0.29205925	0.0373	0.9824
5	0.11786449	0.06890844	0.0107	0.9931
6	0.04895606	0.03666075	0.0045	0.9975
7	0.01229531	0.00382043	0.0011	0.9987
8	0.00847488	0.00468250	0.0008	0.9994
9	0.00379238	0.00193085	0.0003	0.9998
10	0.00186153	0.00132925	0.0002	1.0000
11	0.00053228		0.0000	1.0000

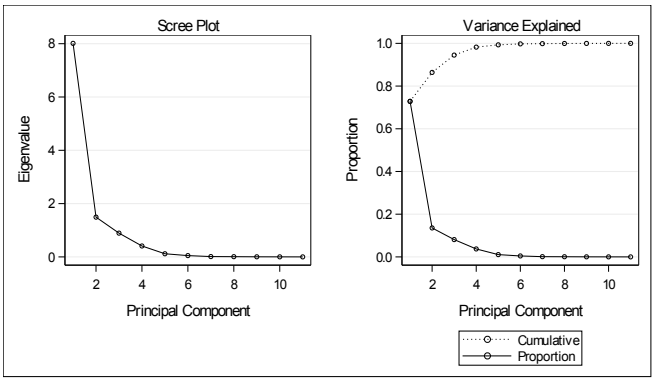
TABLE 2
EIGENVECTORS BY PRINCOMP PROCEDURE OF SAS

	Eigenvectors					
	Prin1	**Prin2**	**Prin3**	**Prin4**	**Prin5**	**Prin6**
PrevClose	0.344778	-.134233	-.082486	0.140012	-.110190	0.027592
OpenPrice	0.344485	-.137975	-.080065	0.152069	-.125276	0.063528
HighPrice	0.343286	-.156055	-.051552	0.172760	-.144021	0.090544
LowPrice	0.343030	-.128386	-.132182	0.141899	-.165304	0.129198
AveragePrice	0.343346	-.144312	-.091947	0.161642	-.162545	0.118537
TurnoverinLacs	0.073994	-.357176	0.925123	0.073971	0.040957	0.003645
DMA10	0.346389	-.068185	-.063902	-.047300	0.324492	-.440118
DMA20	0.339626	0.027241	-.021984	-.253197	0.575461	-.327431
DMA50	0.303821	0.273381	0.113698	-.516519	0.185963	0.693351
DMA100	0.251202	0.485931	0.222806	-.336579	-.604633	-.411069
DMA200	0.113316	0.676350	0.187652	0.655185	0.243688	0.074971

	Eigenvectors				
	Prin7	**Prin8**	**Prin9**	**Prin10**	**Prin11**
PrevClose	-.512313	-.335968	0.188662	0.642516	0.006320
OpenPrice	-.371963	-.294073	0.067083	-.752651	0.131186
HighPrice	0.137191	0.015041	-.639519	0.021339	-.606523
LowPrice	0.461953	0.183942	0.680138	-.025198	-.271597
AveragePrice	0.348218	0.144311	-.290824	0.136888	0.735415
TurnoverinLacs	0.021580	0.011211	0.056369	0.001580	0.012816
DMA10	-.324674	0.681017	0.006853	-.028240	-.001969
DMA20	0.349437	-.508542	-.011789	0.004640	0.008462
DMA50	-.126929	0.146852	-.014050	0.003117	-.005963
DMA100	0.039929	-.040236	-.006858	-.004553	-.000346
DMA200	0.003292	0.007118	0.012163	0.003138	0.002319

A chart of Eigenvalues and principal component or proportion and principal component was plotted to obtain clear idea about how much components or dimensions can be used to represent original input data.

PLOT 2
SCREEN PLOT OF EIGENVALUES & PRINCIPAL COMPONENTS



From plot 2 it is clear that first four major components are good approximate to the original dataset. From plot 2 and table 1 it is clear that first four major principal components covers 98.24 % of original data.

B. Network Training

Through initial study we found that NARX network with 12 inputs variables, 1 hidden layer with 10 neurons, 1 output neuron, feedback delay of 2, Gradient descent adaptive back-propagation (GDA) training algorithm, symmetric sigmoid transfer function in hidden layer and pure linear transfer function in output layer is optimal network.

The Principal Components obtained in 5.A were used as input to a NARX network with 1 hidden layer with 6 neurons, 1 output neuron, feedback delay of 2, Gradient descent adaptive back-propagation (GDA) training algorithm, symmetric sigmoid transfer function in hidden layer and pure linear transfer function in output layer. As discussed in 5.A only first four major principal components were used as input to NARX network. As PCA has reduced

dimensionality to 4 from 12, we used 6 neurons in hidden layer instead of 10.

C. Performance Measurement

To measure the prediction accuracy of the model, the predicted values were compared with actual outputs of sample data. Normalized Mean Square Error (NMSE) is used to evaluate prediction of accuracy of the model. Following formula is used to calculate NMSE.

$$NMSE = \frac{\sum_{t=1}^{N_1}(P_t - O_t)^2}{\sum_{t=1}^{N_1}(P_t - \overline{P}_t)^2} \quad \textbf{(2)}$$

where P_t represents actual value of the pre-processed data series i.e. closing price of share, O_t represents observed value or the predicted value i.e. predicted closing price of share for the same day and $\overline{P}_t$ is the mean of the actual value.

In order to calculate the error percentage, actual closing price and predicted closing price were compared. The formula to calculate error percentage is as follows.

$$Error\,\% = \frac{|P_t - O_t|}{P_t} * 100 \quad \textbf{(3)}$$

Where P_t represents actual closing price and O_t represents observed or predicted closing price of share. From Eq (3), it is clear that, a network with less error % should be considered as best network. The value of 0 for error % indicates that there are no errors in actual and predicted values which indicate perfect prediction.

VI. EXPERIMENTAL RESULTS

After predicting share prices for year 2013, performance of model was calculated by using **Eq (2) and (3)**. The model with PCA inputs gave NMSE of **0.03** and error % of **2.12%**. Table 3 gives performance details of the proposed model without PCA and with PCA.

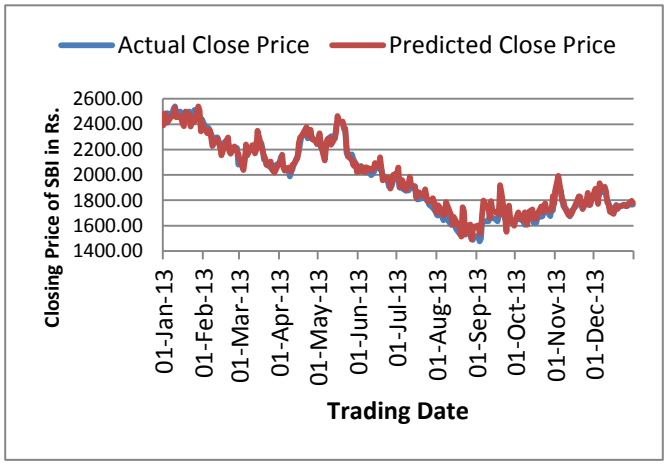
TABLE 3
PERFORMANCE MEASURES OF ANN STRUCTURES USED IN STUDY

Performance measures of ANN			
Sr. No.	**Parameter**	**Model without PCA**	**Model with PCA**
1	Input variables & neurons	12	4
2	Neurons in hidden layer	10	6
3	Epochs	162	121
4	NMSE	0.03	0.03
5	Error %	1.93	2.12

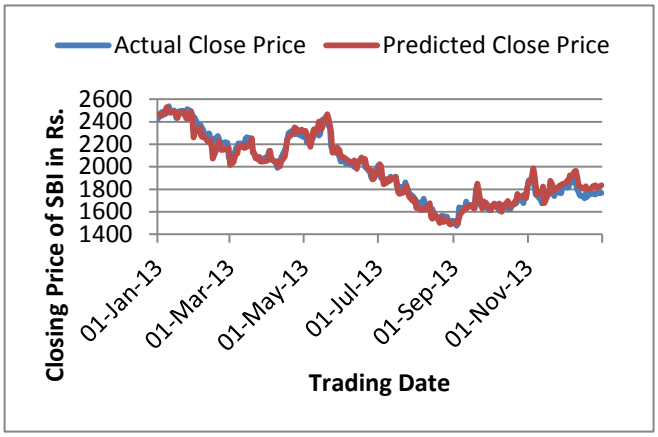
From **table 3**, it is observed that both the models gave NMSE of 0.03. But the model with PCA has less number of input variables, input neurons, and hidden neurons. Also epochs taken during the training phase are less in model with PCA. Though error % in model with PCA rose by 19 basis points i.e. 0.19, it is acceptable as we are comparing the closing values. If we consider the particular trading day, the share price moves in a range where predicted value might have recorded. In such a case the model performance is even higher than what numbers shows. This concludes that NARX network with PCA is faster than NARX network without PCA. As in model with PCA, input data and hidden neurons are minimum, time required to predict closing price will be less as compared to a model without PCA.

Plot 3 and **plot 4** are providing clear idea about prediction accuracy of NN model without PCA and with PCA respectively. Both plots conclude that the predicted values by NN model are in line with that of actual closing values of SBI.

PLOT 3
MODEL WITH ORIGINAL DATA



PLOT 4
MODEL WITH PCA DATA



VII. CONCLUSION

An efficient model for Stock price prediction using PCA and NN has been presented in this paper. During the study we observed that normalization technique like Z-Score normalization is useful to avoid scaling issues in the data. In input data under study, attributes were of different scales. Due to normalization of data, Principal Component Analysis gave better results.

We observed that PCA is better methodology to convert correlated data to linearly uncorrelated data. PCA process removes relations, if any, in the data. Due to such conversion, the prediction process becomes more reliable as it does not rely on correlation in input data. The PCA is also used for dimensionality reduction. In this study, initially there were 12 attributes. Due to correlation between attribute Turnover and attribute Total transactions, we removed attribute total transactions. After performing PCA, we conclude that first four principal components are representing 98.24% of information from original data. So instead of using entire data of 12 attributes, we used only 4 components. Due to this, the size of data on which prediction model has to work is reduced by large extent. This leads to faster response by prediction model.

ANN based NARX model was constructed for prediction of closing price of SBI using the principal components as input to the NARX model. As input is now reduced, the number of input neurons and neurons in hidden layer were reduced which in turn leads to less number of weights. This again leads to faster processing and response by prediction model. We also observed that the model with PCA data takes less epochs during learning phase than model with original data.

The proposed prediction model with original data gave NMSE of **0.03** and error % of **1.93%** and prediction model with PCA data gave NMSE of **0.03** and error % of **2.12%**. This concludes that we can achieve almost same results by using lesser data given by PCA. Thus PCA was found to be very useful for prediction model. The proposed model can be effectively used, after training for respective data, to predict closing price of other companies also.

ACKNOWLEDGMENT

The authors wish to thank Dr. M. S. Prasad, Pune and Mr. Guruprasad Amane, CoreCompete Pvt. Ltd. for their valuable suggestions during the study of the proposed work. The authors also wish to express their thanks to National Stock Exchange (NSE) of India Ltd., a leading stock exchange in India, for providing the data for study.

REFERENCES

[1] Kao, L-J., Chiu, C-C., Lu, C-J. and Yang, J-L. (2013) 'Integration of nonlinear independent component analysis and support vector regression for stock price forecasting', NeuroComputing, Vol. 99, No. 1, pp.534–542

[2] Wang, Y. (2014) 'Stock price direction prediction by directly using prices data: an empirical study on the KOSPI and HSI', Int. J. Business Intelligence and Data Mining, Vol. 9, No. 2, pp.145–160.

[3] J. Edward Jackson, 'A User's Guide To Principal Components', A Wiley-Interscience Publication, Page 1

[4] Abhyankar, A. Copeland, L.S., & Wong, W., Uncovering nonlinear structure in real-time stock-market indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100, Journal of Business & Economic Statistics, 15, 1–14., 1997

[5] Mbeledogu N. N., Odoh M. And Umeh M.N., 'Stock Feature Extraction using Principal Component Analysis', 2012 International Conference on Computer Technology and Science, IPCSIT vol. 47

[6] Marijana Zekić-Sušac, Nataša Sarlija, and Sania Pfeifer, "Combining PCA Analysis and Artificial Neural Networks in Modelling Entrepreneurial Intentions of Students", Croatian Operational Research Review (CRORR), Vol. 4, 2013

[7] Richard A. Johnson, Dean W. Wichern, 'Applied Multivariate Statistical Analysis', 6th Edition, Pearson Prentice Hall, Page 430

[8] Tsungnan Lin, Bill G. Horne, Peter Tino, C. Lee Giles, Learning long-term dependencies in NARX recurrent neural networks, IEEE Transactions on Neural Networks, Vol. 7, No. 6, 1996, pp. 1329-1351

[9] Yang Gao, Meng Joo Er, NARMAX time series model prediction: feed-forward and recurrent fuzzy neural network approaches, Fuzzy Sets and Systems, Vol. 150, No. 2, 2005, pp.331-350

[10] Prashant S Chavan, Prof. Dr. Shrishail. T. Patil, 'Parameters for Stock Market Prediction', Int. J. Computer Technology & Applications, Vol 4 (2), 337-340 , ISSN:2229-6093

[11] www.nseindia.com/products/content/equities/eq_security.htm

[12] Jiawei Han, Micheline Kamber, Jian Pei, Data Mining Concepts and Techniques, MK, 3rd edition, Page 113-114

[13] Support.sas.com/documentation/cdl/en/statug/67523/HTML/default/viewer.htm#statug_princomp_overview.htm
[14] https://en.wikipedia.org/wiki/SAS_(software)

