International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 99-107
Published Online November 2019 in IJEAST (http://www.ijeast.com)

EVALUATION OF POLLUTION MORTALITY USING NARX AND NIO TIME SERIES PREDICTIVE ALGORITHM ON MATLAB

Afolabi Basit Bolaji, Olabanji Ayodele Olawale, Ibiye Abdultawwab Ayotunde
Department of Electrical and Electronics Engineering
Federal University of Technology, P.M.B 704, Akure, Ondo State, Nigeria.

***Abstract*— Mortality is an essential health effect of ambient air pollution and has been studied extensively. The earliest signal relates to fog occurrences, but with the advancement of more accurate methods of investigation and prediction, it is still possible to differentiate short-term chronological associations with day-to-day mortality at the historically low levels of air pollution now occurring in most developed countries. This paper studies and explores the methodologies for modeling, simulation, and controls in ANN-based on time series application of pollution mortality. To show and prove the effectiveness, simulated and operational data sets are employed to demonstrate the ability of neural networks in capturing complex nonlinear dynamics where NARX and NIO models are set up to explore and relate both steady-state and transient features on pollution mortality. The structures were configured, generated, and ran in MATLAB to create and train the platform. The validation, testing, and results validate that the techniques can be accurately applied, which implies both models effectively capture dynamics of the system up to a certain degree of acceptance. The associated parameters for the design and simulation are varied and set up according to the requirements which display that ANN can perform better than most conventional methods. Finally, it was established that NARX model outperforms more than the NIO model.**

Keywords— NARX, NIO, Regression, Time series prediction, Neural Networks, Machine Learning, Levenberg-Marquardt

I. INTRODUCTION

Serious air pollutions struck Athens, which became apparent in the early 1970s. Studies for the years 1975–1982 have shown a positive link of Sulphur Dioxide (SO_2) with total daily death. Pollution profile in Athens since 1983 has progressively changed but the levels of smoke, SO_2 Furthermore, carbon monoxide (CO) remain comparatively high [1]. The relationship between daily mortality and short-term variations in the ambient levels of ozone(O_3), black smoke(BS), sulphur dioxide(SO_2), carbon monoxide (CO), nitrogen dioxide (NO2), and particulate matter was considered in the Netherlands. Daily total and cause-specific mortality count air quality, temperature, relative humidity, and influenza data were acquired from 1986–1994. The correlation between daily mortality and air pollution was demonstrated using Poisson regression analysis, and it was revealed that more substantial relative risks for air pollution were generally found in the elderly, excluding for ozone and for death-cause pneumonia, which exhibited more substantial relative risk in younger age groups [2].

This paper will evaluate the performance between two Algorithms that are used to predict pollution mortality. Air pollution factors measurements are in the same dataset storing all historical air pollution factors, and data will be treated as training sets for the program in the two models. The primary purpose of the prediction is to reduce doubt associated with decision making in investment, predicting future possibilities. In other to ascertain which model is best to predict, their performance evaluation the mean absolute error (MAE), mean percentage error (MPE) and root mean square error (RMSE) encountered during the model testing are considered, but we are specific about mean square error (MSE) in this research work.

II. LITERATURE REVIEW

Several works have been done in using ANN to forecast future occurrences. Stock prediction using both ANN and LSTM. The stock market price was predicted using Artificial Neural Networks that is a feed-forward multi-layer perceptron with error back propagation, the model was tested on 2008-2012 data obtained from stock markets Nairobi Securities Exchange and New York Stock Exchange. The prediction result shows Mean Absolute Percentage Error between 0.71% and 2.77%. After the necessary validation, the model was capable of prediction on typical stock markets [3]. The stock market was also predicted using Machine Learning. Available stocks data were used to train the machine to gain intelligence, then use the developed knowledge for an accurate prediction. Machine learning called Support Vector Machine (SVM) was used to predict stock prices for large and small capitalization in the different markets employing prices with both daily and up-to-the-minute frequencies [4]. The stock market price was also predicted using machine learning algorithm that integrates least square support vector machine (LS-SVM) and Particle swarm optimization (PSO). The PSO algorithm employed seeks to enhance LS-SVM to predict daily stock prices. The model employed was based on the study of stock's historical data and technical indicators. The best free parameters were selected by PSO algorithm for combinations for LS-SVM to evade over-fitting and local minima issues and enhance prediction accuracy. The model applied was evaluated using thirteen benchmark financials datasets and related with artificial neural network of Levenberg-Marquardt (LM) algorithm. The results obtained revealed that the SVM model has better prediction accuracy and a good potential of PSO algorithm in improving LS-SVM [5].

Due to a lot of complicated financial indicators and also the volatile fluctuation of stock markets, there is a need to harness this negativity through technological advancement. In other to gain a steady fortune from stock market and helps experts to find the most informative pointers to make an improved prediction. Recurrent Neural Networks (RNN) have proved over time to be one of the most powerful models to process sequential data. Long Short-Term Memory (LSTM) has been proved to be most successful RNN's architectures, it introduces memory cell, a unit of computation that substitutes traditional artificial neurons in the hidden layer of the network with these memory cells, networks are able to associate memories effectually and input remote in time, hence ensemble to grasp the structure of data vigorously over time with excellent prediction capacity. The LSTM used stock returns of NIFTY 50 using LSTM. Five years of historical data of NIFTY 50 was used for the training and validation purposes for the model. Recurrent Neural Networks and Long Short-Term Memory unit used helps to forecast stock market by providing a good insight into the future situation of the stock market [6]. The stock market's influence on today's economy is underestimated. Fluctuation in the market prices always has a strong effect on the economy and has a role to play in determining the investor's gain. There are several forecasting methods employed in predicting market prices. Unlike other approaches, the performance of RNN, CNN, and LSTM was quantified using the percentage error. CNN architecture is capable of recognizing the changes in trends. CNN was identified as the best model that uses the information given at a particular instant for prediction. Though the other two models are used in much other time-dependent data analysis, they do not outperform the CNN architecture in this research [7].

III. OVERVIEW OF PAPER

The flowchart is shown in fig. 1 is showing different phases of the predictive approach for both NARX and NIO, which involves various steps required to be attained for this approach. Dataset needed for the predictive system will be computed at the historical dataset phase, which will be transformed into time series format (tonndata) in the form of matrix in time step (cell column, matrix column, matrix row). Noise treatment is necessary to check the effect of noisy data on the performance of classifier learning algorithms, which is obligatory to improve their reliability. There is three kinds of target timesteps employed in this model, and they include training, validation, and testing. Training makes use of 70% of the total dataset presented to the network, and the network is adjusted according to its error. The validation phase is used to measure the system generalization and halt training when generalization stops improving, and this uses 15% of the total dataset. This phase testing phase also uses 15% but has no effect on the training and so make available an independent measure of network performance during and after training in other to ascertain the accuracy of the model.

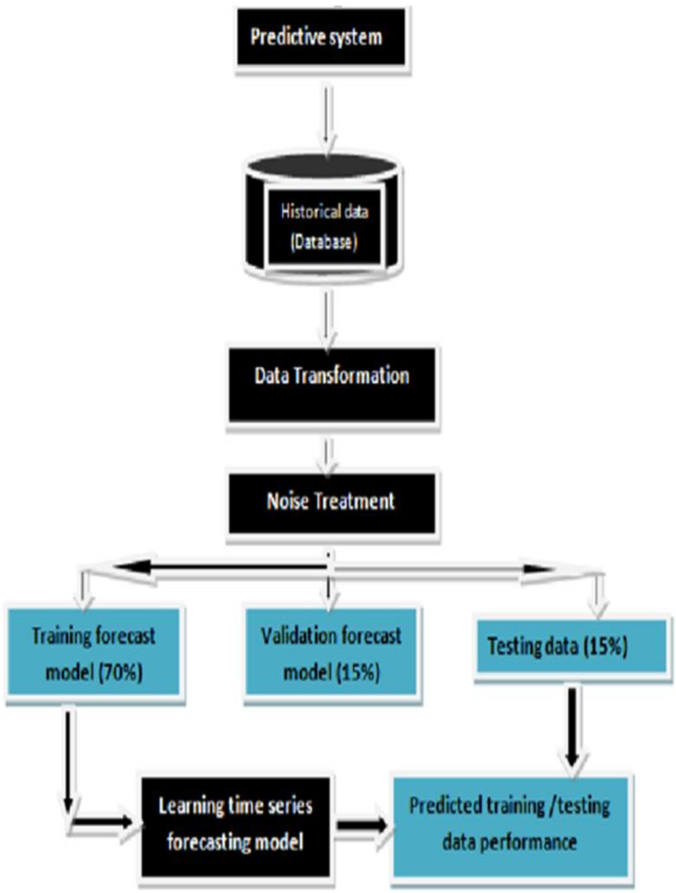

Fig. 1: Predictive Flowchart for both NARX and NIO

A. Time Series Modeling

Traditional time series forecasting technique creates forecasting model from the training data set by determining the unrecognized factors. Traditional time series analysis and artificial neural network (ANN) techniques was to model and forecast the power consumption of Bangkok's metropolitan area. Time series data in form of units of household electricity consumption were gotten from the Metropolitan Electricity Authority of Thailand. The data was gathered monthly from

January 2010 to May 2015. Forecasting models with varying parameters are created from both methods using the training data, which are the series ranging from January 2010 to December 2014. While the remaining data from January 2015 to May 2015 are used as the testing data. Performance of each models were compared using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). Traditional time series forecasting models examined are ARIMA, GLM, and HoltWinters. Four models were examined under ANN using 3 layers with different number of neurons, they include ranging 3L-4N, 3L-5N, 3L-6N, and 3L-7N. It was revealed from the experimental results that ARIMA is greater among the traditional time series models, considering intelligent-based models 3L-6N is the finest of ANN models. Furthermore, MAPE benchmark of the 3L-6N model is less than that of ARIMA model. As a result, it was concluded that ANN model is more powerful in foretelling power distribution units than the traditional time series models [10].
It was pointed out that results achieved by neural networks can be similar to those obtained by traditional statistical methods but there is need to continuously investigate into comparing the efficiency of ANN forecasting models taking into cognizance the architecture network, defining size of hidden layers, learning algorithms, error methods and alternative statistical procedures [11]. A multilayer perceptron was proposed for predicting time series with intervention, it was based on generation in line with a rule emerging from ARIMA models with interventions previously fitted of a set of nonlinear forecasting models with interventions. The multi-layer perceptron was trained by three alternative learning rules incorporating hidden layers and multiple repetitions were computed by means of grid search. Time series shows a better performance of these neural networks over ARIMA with interventions [12].

B. Nonlinear Autoregressive with External (Exogenous) Input (NARX)

Nonlinear Autoregressive Network with Exogenous Inputs (NARX) is a recurrent dynamic network with feedback connections that enclose several layers of the network [8]. The snapshot of NARX network is shown in fig. 2 with tapped delay lines and two-layer feed-forward network, a sigmoid transfer function in the hidden layer, and a linear transfer function in the output layer.

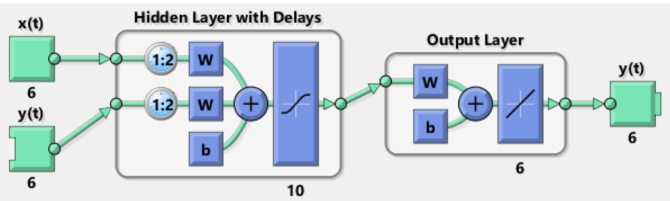

Fig 2: NARX network overview

Considering a scalar (or vector) with output *P* and time *t*

$$\{x(t_o), x(t_1), \ldots\ x\ (t_{i-1}), x(t_i), x\ (t_{i+1}), \ldots\} \qquad (1)$$

$x(t)$ Is a continuous signal while t is known to be real-valued. The discrete signal can be presumed by uniform sampling, as shown in Eqn. (2) where sampling period Δt is introduced agreeing to the Nyquist sample theorem [7] (Levesque, 2014).

$$\{x[t]\} = \{x(o), x(\Delta t), x(2\Delta t), x(3\Delta t), \ldots\} \qquad (2)$$

Estimation of *x* at some future time

$$\hat{x}(t+s) = f(x[t], x[t-1], x[t-2], \ldots) \qquad (3)$$

Where s is called the horizon of prediction. In a situation where s is set the value of $1(s = 1)$, then it is dubbed as a *one-time step* ahead prediction; otherwise, also called *multi-step* ahead prediction.

C. Nonlinear Input-Output (NIO)

The Nonlinear Input-Output (NIO) time series is defined as

$$y(t) = f\big(x(t-1), \ldots, x(t-d)\big) \qquad (4)$$

This is has a salient resemblance with NARX because it includes two series of an input series $x(t)$ and an output/target series $y(t)$. The description of Eqn. 4 gives an insight into predicting values of $y(t)$ from previous values of $x(t)$ without previous knowledge of preceding values of $y(t)$. The overview of NIO network is shown in fig. 3 with tapped delay lines and a sigmoid transfer function in the hidden layer, a layer feed-forward network, and a linear transfer function in the output layer.

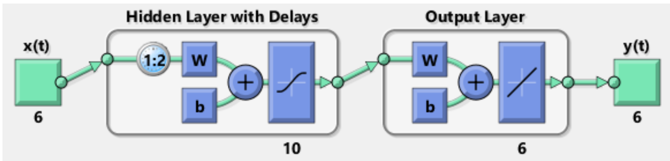

Fig 3: NIO network overview

IV. Methodology

Levenberg-Marquardt (LM) optimization is a virtual standard in nonlinear optimization. It is a pseudo-second-order method which means that it works with only function calculations and gradient information, but it estimates the Hessian matrix using the sum of outer products of the gradients [9].

$$W_k = W_{k-1} - [H_{k-1} + \lambda_{k-1}.I]^{-1}.j_{k-1} \qquad (5)$$

In this research work, random data training with LM (trainlm), and using (dividerand) command along with Mean squared error (MSE) was used. All target vectors and input vectors are randomly separated into three set values of validation, training, and testing with assigned values of $15\ \%, 70\%$, and 15%, respectively. The number of neurons and delays was tuned up to further vary the performance of the network at 10, 20, 30, 40, and 1, 2, 3 4, correspondingly. The network stopped when the target MSE was achieved or after the maximum number of epochs was reached. Consequently, the matrix m by n with m rows and n columns is called the size of the matrix. Thus matrix is of the form:

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ . & . & \dots & . \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix} \qquad (6)$$

A column vector is defined as

$$b = \begin{bmatrix} b_1 \\ b_2 \\ .... \\ b_m \end{bmatrix} \qquad (7)$$

A. Experimental Data

The dataset used to evaluate the performance between NARX and NIO is pollution. set, the dataset was used to train the neural network to predict mortality due to pollution. It is a 1×508 cell array of 8×1 vectors representing eight measurements over 508 timesteps. The measurements are Temperature, Relative humidity, Carbon monoxide, Sulfur Dioxide, and Nitrogen dioxide, Hydrocarbons, Ozone, and Particulates. Pollution Targets is a 2×508 cell array of 3×1 vectors representing 219 timesteps of three kinds of mortality to be predicted. The to-be predicted mortality includes Total mortality, Respiratory mortality and cardiovascular mortality.

V. RESULT AND DISCUSSION

This section describes the complete series of tests implemented, simulation output, and experimental set-up.

A. Simulated NARX Results

The simulated results for NARX using some variable parameters are illustrated in Table 1 to 4, Figures 4 to 10. The Levenberg-Marquardt (LM) algorithm converged upon a resolution after maximum 40 iterations with no significant error cross-correlation or autocorrelation issues identified. Therefore, there are highly significant ($p < 0.001$) correlations between output and target data at good fit (R values that more significant than 0.9).

Table 1: Result of NARX with parameters n=10, d=1 (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	21.9366e-0	9.97501e-1	14	0:00:02	16.6	10.0	6
Validation	35.4214e-0	9.96014e-1					
Testing	64.4903e-0	9.92713e-1					

Table 2: Result of NARX with parameters n=20,d=2 (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	2.99491e-3	9.99786e-1	40	0:00:13	0.00222	0.167	6
Validation	2.14314e-1	9.85169e-1					
Testing	8.84875e-2	9.93513e-1					

Table 3: Result of NARX with parameters n=30,d=3 (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	1.45384e-2	9.98963e-1	26	0:00:19	0.00838	0.147	6
Validation	6.32914e-2	9.95153e-1					
Testing	6.59202e-2	9.95014e-1					

Table 4: Result of NARX with parameters n=40,d=4 (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	4.25874e-3	9.99691e-1	21	0:00:031	0.00267	0.145	6
Validation	7.21633e-3	9.99484e-1					
Testing	1.15341e-2	9.99187e-1					

Graphical representations of the simulation of NARX are showcased in figure 4 to figure 10

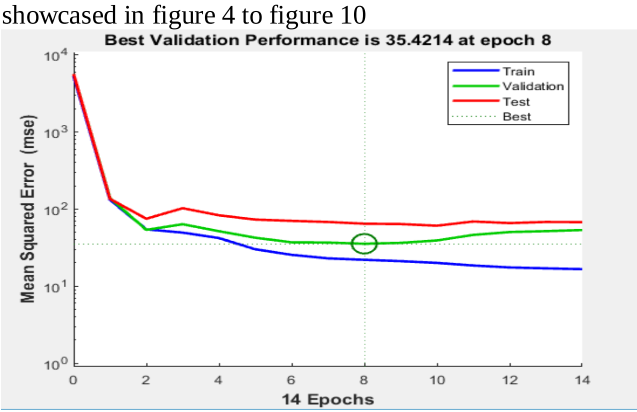

Fig 4: Performance plot of NARX (plotperform)

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 99-107
Published Online November 2019 in IJEAST (http://www.ijeast.com)

There are decreases in errors in training, validation, and testing as shown in fig 4 until iteration 40 is attained which illustrated that there is no element of incidence of overfitting. The training, validation, and testing are performed in open-loop. Likewise, the R values are also calculated based results obtained through open-loop training.

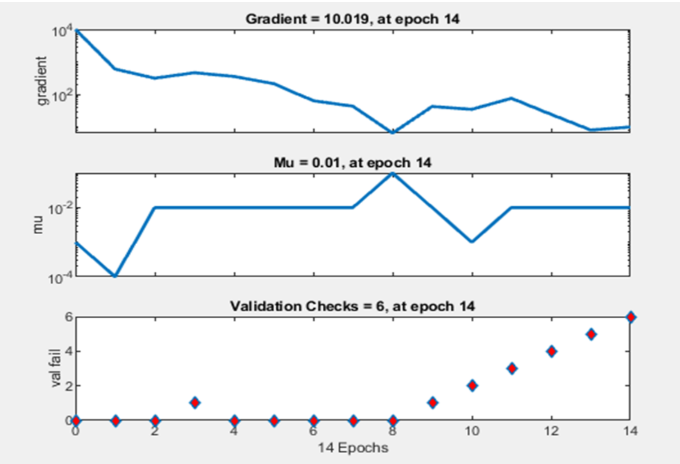

Fig 5: Training state of NARX (plottrainstate)

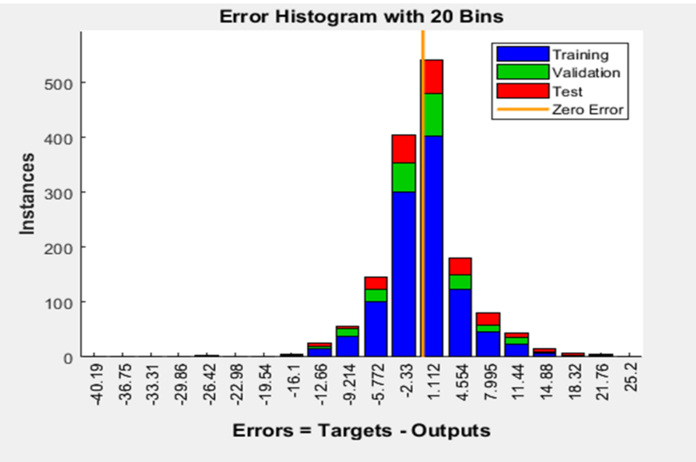

Fig 6: Error histogram of NARX (ploterrhist)

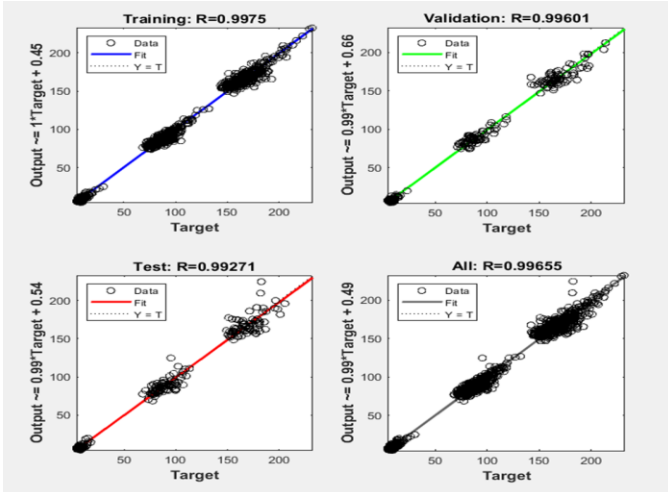

Fig 7: Regression of NARX plot (plotregression)

Fig 7 showed the regression of the NARX plot with four different plots representing the training, validation, and testing and output data. This congested plot of NARX helps show that most of the data points are highly inter-related. The solid straight lines represent the best fit linear regression line between outputs and targets of training (blue), validation (green), testing (red), and output of all (black) while the dashed line in each plot represents the perfect result – outputs = targets. The regression (R) value indicates the relationship between the outputs and targets. If R is close to zero means there is no linear relationship between outputs and targets but, if R = 1, then there is an indication that there is an exact linear relationship between outputs and targets. In this research work, it was shown that the training data (R=0.9975) indicates a good fit, validation (R=0.99601), and test (0.99271) results also show R values that more significant than 0.9.

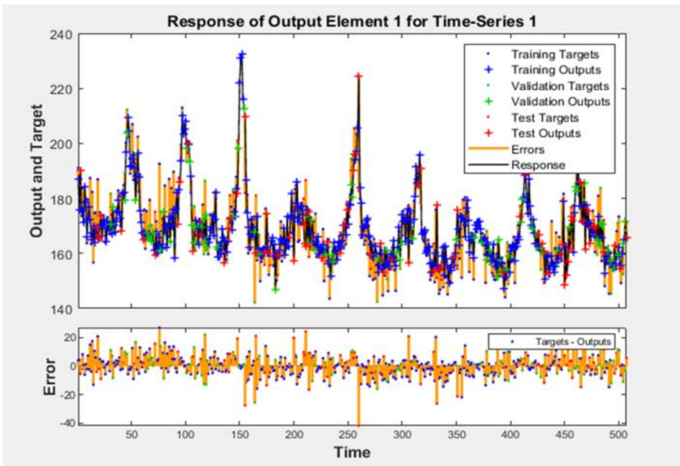

Fig 8: Time-series response of NARX (plotresponse)

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 99-107
Published Online November 2019 in IJEAST (http://www.ijeast.com)

Fig. 8 shows the Time Series Response of NARX, which gives a clear indication where time points were selected for training, testing, and validation. The inputs, targets, and errors versus time were well displayed through time Series Response.

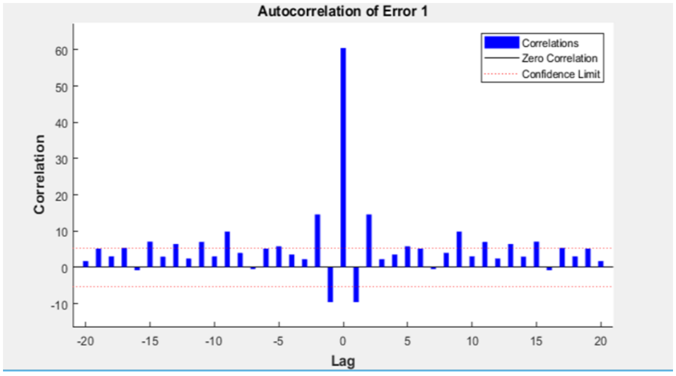

Fig 9: Error autocorrelation of NARX (ploterrcorr)

Fig. 9 displays the error autocorrelation of NARX using error autocorrelation function to certify the performance of the trained network to explain how the prediction errors are related in time. It was observed that there was noteworthy correlation in the prediction errors; most of the trained network falls within the red confidence limits, which can make room for possible upgrading. If the network has been trained as expected all the other lines will be much smaller, and almost all the lines will fall within the red confidence limits

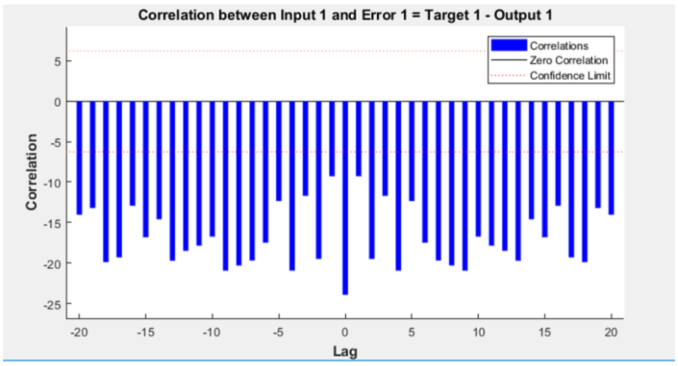

Fig 10: Input-Error Cross-Correlation of NARX (plotinerrcorr)

Fig 10 shows the input-error cross-correlation function that supports attaining extra verification of network performance in the sense that it inspects how the errors are interrelated with the input sequence x(t). It was observed that the input correlates with the error which also indicated that there is room for prediction enhancement by increasing the number of delays in the tapped delay lines.

B. Simulated NIO Results

The results for the simulated NIO using several variable parameters are illustrated in Table 5 to 8. Again, the algorithm for LM converges upon an outcome after a maximum of 18 iterations with no significant error cross-correlation or autocorrelation issues recognized. Therefore, there are highly significant ($p < 0.001$) correlations between output and target data at bad fit (R values are far lesser than 0.9).

Table 5: Result of NIO with parameters n=10, d=1 (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	36.7471e-0	9.95844e-1	16	0:00:02	33.3	86.2	6
Validation	52.4290e-0	9.93975e-1					
Testing	68.8847e-0	9.92110e-1					

Table 6: Result of NIO with parameters n=20, d=2 (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	41.1172e-0	9.95771e-1	12	0:00:07	16.8	57.6	6
Validation	53.8765e-0	9.94095e-1					
Testing	55.1757e-0	9.93942e-1					

Table 7: Result of NIO with parameters n=30, d=3 (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	18.8026e-0	9.97915e-1	12	0:00:42	1.24	24.3	6
Validation	165.7893e-0	9.81779e-1					
Testing	163.4675e-0	9.83133e-1					

Table 8: Result of NIO with parameters n=40, d=4 (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	15.0351e-0	9.98300e-1	18	0:02:46	0.520	9.55	6
Validation	62.4449e-0	9.92891e-1					
Testing	102.1338e-0	9.88609e-1					

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 7, ISSN No. 2455-2143, Pages 99-107
Published Online November 2019 in IJEAST (http://www.ijeast.com)

Graphical representations of the simulation of NIO are showcased in figure 11 to figure 17

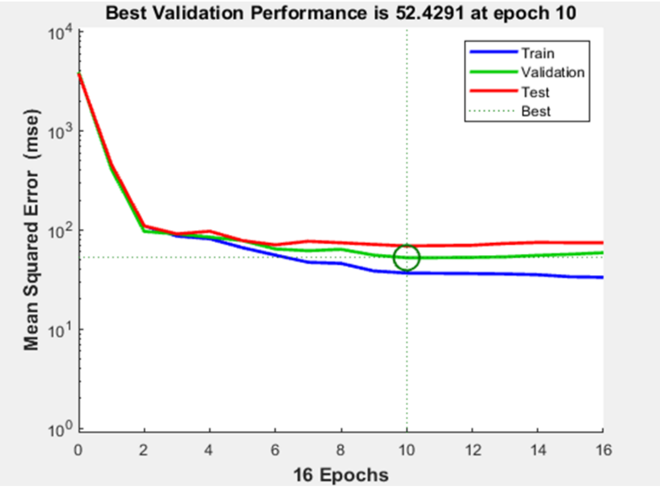

Fig 11: Performance plot of NIO (plotperform)

Fig. 11 demonstrates the performance plot of NIO with a problem of over fitting with the training. It was observed that test curves and validation are similar. The test curve is on the same par with validation curve. Therefore, they are sparsely fitted.

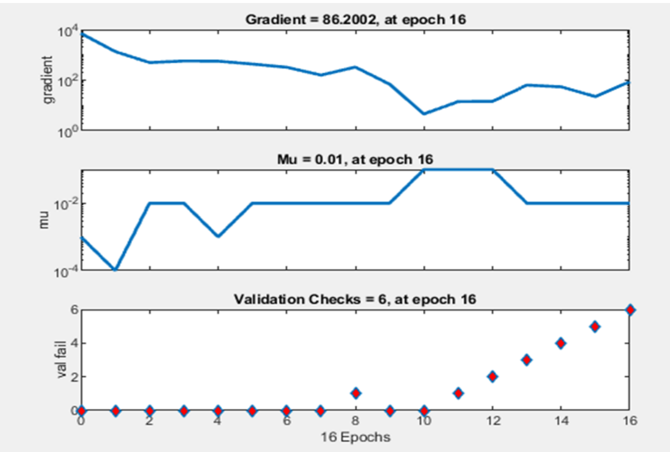

Fig 12: Training state of NIO (plottrainstate)

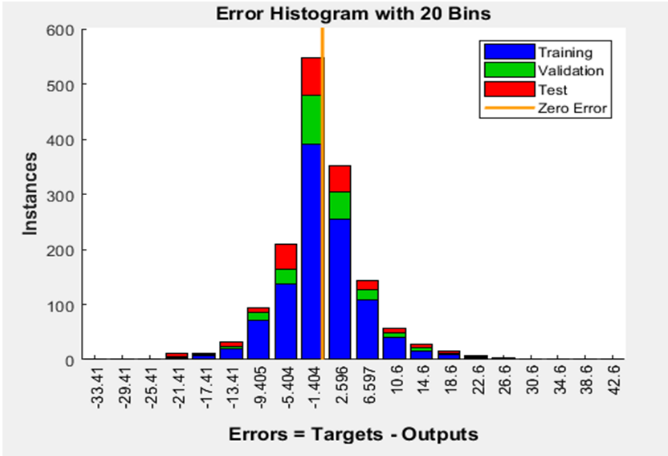

Fig 13: Error Histogram of NIO

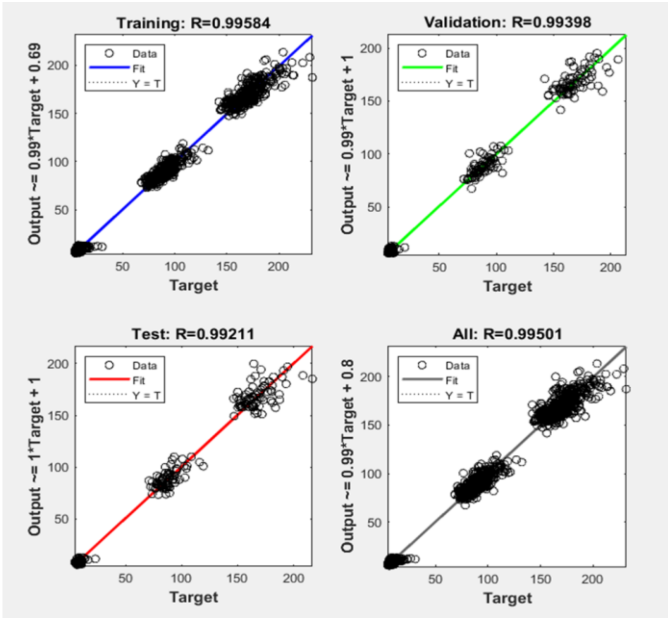

Fig 14: Regression of NIO plot (plotregression).

The regression of NIO alongside the congested plots showing some data points as illustrated in Fig 14 are highly interrelated. In this research work, it was shown that the training data (R=0.99584) indicates a good fit, validation (R=0.99398), and test (0.99211) results also show R values that are slightly more than 0.9.

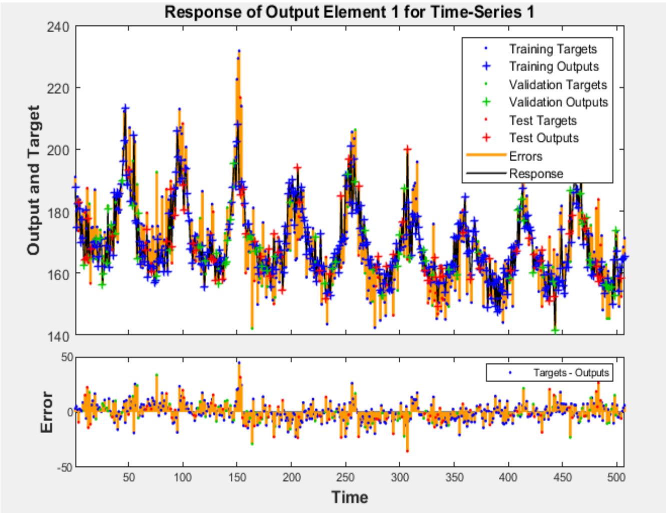

Fig 15: Time Series Response of NIO

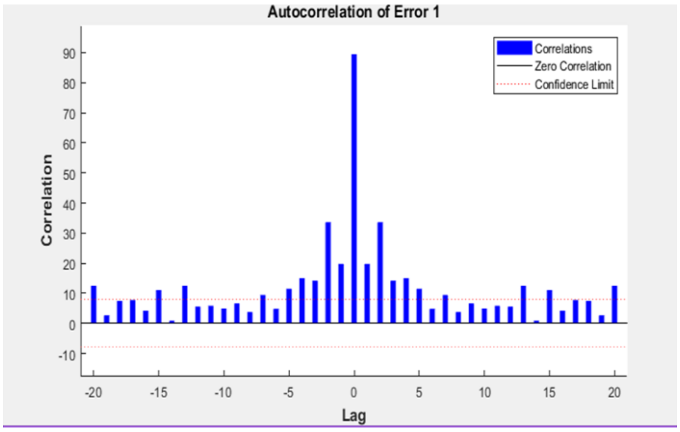

Fig 16: Error Autocorrelation of NIO

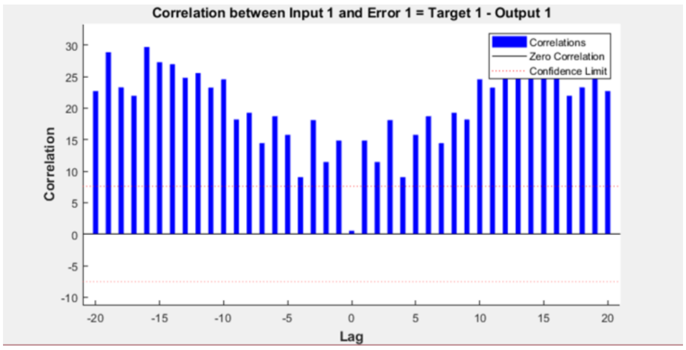

Fig 17: Input-Error Cross-correlation

Fig 17 shows the input-error cross-correlation function that supports attaining extra verification of network performance in the sense that it examines how the errors are interrelated with the input sequence x(t). It was observed that the input correlates with the error which also indicated that there is room for prediction enhancement by increasing the number of delays in the tapped delay lines

C. Performance Evaluation of NARX and NIO

General measures of performance error evaluation (target-output) were achieved and summarized in column 2 of Table 1 to Table 8 for both models (MSE). These results suggested that the NARX model produces a greater predictive capacity for both fit and accuracy while NIO yields a modest predictive capacity for accuracy but bad fit.

VI. CONCLUSION.

We have presented models founded on ANN ideas for stock market prediction and evaluate advantages and drawbacks of the models through results comparison. Despite all the controversial issues regarding ANN, this research work further confirmed that NARX model holds high and strong potential to be considered as a dependable substitute to conservative methods. The NIO results were not close with NARX results due to some applications in which the previous values of y(t) would not be available. However, those are the only cases where one would want to use the input-output model instead of the NARX model. The NARX model argued and explored in this research had provided enhanced predictions than NIO model. This is because NARX uses extra information contained in the earlier values of y(t). In conclusion, it was observed that both NARX and NIO could efficiently learn complex sequences and outdo some well-known, existing models, but NARX is better off than NIO of the same class.

VII. REFERENCE

[1] G. Touloumi, S. Pocock., K. Katsouyann, and D. Trichopou. “Short-Term Effects of Air Pollution on Daily Mortality in Athens: A Time-Series Analysis”. *International Journal of Epidemiology*, Vol. 23, Issue 5, pp. 957–967, 1994.

[2] P. Fischer., G. Hoek., B. Brunekreef., A. Verhoeff., and J. van Wijnen. “Air pollution and mortality in the Netherlands: are the elderly more at risk?” *European Respiratory Journal.* pp. 34-38, 2003

[3] W. W. Barack, and M. Lawrence. “ANN Model to Predict Stock Prices at Stock Exchange Markets. Nairobi”: *School of Computing and Informatics, University of Nairobi, Box 30197,* 2014.

[4] V. Kranthi, and S. Reddy. “Stock Market Prediction Using Machine Learning”. *International Research Journal of Engineering and Technology (IRJET),* pp. 1032-1035, 2018.

[5] H. Osman, S. S. Omar, and A. S. Mustafa. “A Machine Learning Model for Stock Market”. *International Journal*

of Computer Science and Telecommunications. Vol. 4, Issue 12, pp. 17-23, 2013.

[6] R. Murtaza, P. Harshal, and V. Shraddha. "Predicting Stock Prices Using LSTM". *International Journal of Science and Research (IJSR)*. Vol. 6, Issue 4, pp. 1754-1756, 2017.

[7] S. Sreelekshmy, R. Vinayakumar, E. Gopalakrishnan, K. Vijay, and S. Menon. "Stock Price Prediction Using Lstm,Rnnand Cnn-Sliding Window Model". *IEEE International Conference on Advances in Computing, Communications, and Informatics (ICACCI)* pp. 1643-1647, 2017.

[8] O. Omolaye, and T. Badmos. "Predictive and Comparative Analysis of NARX and NIO Time Series Prediction". *American Journal of Engineering Research (AJER)*, pp. 155-165, 2017.

[9] I. Badi, M. Boutalline, and S. Safi. "The Neural Networks: Application And Optimization Application Of Levenberg-Marquardt Algorithm For Tifinagh Character Recognition". *International Journal of Science, Environment and Technology*, Vol. 2, No 5, pp. 779-786, 2013.

[10] R.Chuentawat, S. Bunrit, C. Ruangudomsakul, N. Kerdprasop, and K. Kerdprasop. "Artificial Neural Networks and Time Series Models for Electrical Load Analysis". *Proceedings of the International MultiConference of Engineers and Computer Scientists IMECS,* Vol. I, pp. 1-6, March, 2016.

[11] T. Hill, L. Marquez, M. O'Connor and W. Remus. "Artificial Neural Networks For Forecasting and Decision Making". *International Journal of Forecasting*. Vol. 10 pp. 5-15, 1994.

[12] M. D. Cubiles-de-la-vega, R. Pino Mejias, J. L. Moreno-Rebollo and J. Munoz Garcia. "A Neural Networks Model for Predicting Time Series with Interventions and a Comparative Analysis". *Journal of Official Statistics*, Vol. 17, No. 4, pp. 447-456 2001.