Artificial Neural Network Model with Adaptive GMDH Technique for Non-Linear Regression Prediction

Abdul Joseph Fofanah[1], Saidu Koroma[1], Alpha Bilhal Sesay[1], Ahmed Fofana[1]
Department of Mathematics and Computer Science, Milton Margai College of Education and Technology

Abstract—A theoretical concept of the GMDH technique using a non-linear regression model, multilayered neural nets, model assessment, and selection to determine the prediction error versus selection model complexity was reviewed and evaluated. The model selection was experimented and evaluated with MATLAB. The multilayered neural networks and the bias-variance decomposition was reviewed theoretically with its core behavioural bias and variance (model space, estimation variance, and restricted model space). The proposed algorithm (with architecture) for the GMDH network using a non-linear regression model was designed and implemented in MATLAB and evaluated. Different parameters (network layers, neurons, selection pressure, and train ratio) were tested and evaluated to determine the robustness of the GMDH-Non-Linear Regression Model Network (GMDH-NLRMN). Our experimental results revealed that for different parameter values, the GMDH network produces slightly with the latter and the greater the number of layers and neurons the better the performance and more accurate the prediction model. According to the trained data, an accuracy of 99.2% was obtained with a minimum of 0.0111 MSE for $n = 5$ and for $n = 8$ the performance of the model increases to 99.4% and with a minimum of 0.01149. However, as the number of parameters value increases, the higher the computational complexity required to train the data and the value of MSE increases slightly.

Index Terms— Artificial Neural Network, GMDH, Linear Regression, Multilayered Neural Net, Bias, Variance, Model Assessment.

1 Introduction

The GMDH is a type of neural network algorithm and organizes machine learning algorithms for the modeling of complex systems. In recent years, the GMDH is designed purposely for different machine learning techniques such as pattern recognition, classification, clustering, the approximation of multidimensional models, forecasting/prediction of various fundamental problems. The adaption of the group method of data handling (GMDH) technique with different forecasting methods has been experimented with to determine the accuracy of chemical forecasting. Recently, data mining technique has been extensively utilized for predictions and forecasting gray modeling [1], wavelet transform [2], support vector machine [3], neural networks [4], and many more. The adoption of the data mining technique is effective due to its flexibility for non-linear data.

The GMDH is a mathematical modeling and non-linear regression algorithms, which was proposed by Alexey Grigorevich Ivakhnenko in 1968. The technique is also known as Polynomial Neural Network and can be referred to as a specific type of supervised Artificial Neural Network (ANN). Furthermore, apart from its modeling specifications, GMDH encapsulates the idea of Natural Selection to control the size, complexity, and accuracy of the network. The main implementation of GMDH is the modeling of complex systems, function approximation, non-linear regression, and pattern recognition. In this research, the structured MATLAB implementation of GMDH and which the researchers used to easily perform modeling function approximation and regression tasks. The study proposed a modified with supervised learning applications, such as Time-Series Prediction and Classification.

Specifically, the GMDH algorithm is a probabilistic algorithm that is based on the multilayered theory of statistical decisions [5]. This method is recommended for the recognition and forecasting of binary objects and for the variability of input data control to circumvent the possible experts' errors in it.

Recently, the development of the GMDH technique has led to neuro nets with active neurons, which have appreciated to twice-multilayered architecture: neurons are multilayered, and they are connected into multilayered architecture. Thus, provide and enhance the possibility to optimize the set of input variables at each layer, while the accuracy of the prediction increases. The accuracy of forecasting, estimation, or pattern recognition can be increased beyond the limits which are reached by neuro set with single neurons [5,6,7,8]. This method corresponds to the actions of the human nervous system, the connections between numerous neurons are not immovable but change depending on the neurons themselves. The active neurons are a self-organizing process to approximate which inputs are necessary to mitigate the given objective function of the neuron. In modeling the GMDH algorithm, it is possible for such a condition that every neuron in its turn in a multilayered unit. In order to increase AI challenges in accuracy and complexity of time, neuro net with active neurons and the help of regression area extension for inaccurate, noisy, or limited data samples to test the variability of the algorithm.

- *Fofanah A.J. is with the Department of Mathematics & Computer Science, Faculty of Environmental Sciences, Milton Margai College of Education and Technology. He has an Associate Degree in Mathematics, B. Sc (Hons.) in Computer Science, Master's Degree in Computer Science, Master's in Software Engineering, and currently pursuing his Ph.D. program in Information Health Systems at the Atlantic International University.*

2 RESEARCH METHODOLOGY

In this paper, we present a non-linear regression model for forecasting using the technique of the Group Method of Data Handling (GMDH). The related literature review will enable us to identify and effect further research on GMDH to investigate the performance of GMDH using a non-linear regression model for Time Series Forecasting.

2.1 Group Method of Data Handling (GMDH)

The various method has been developed to handle data for predictions and classification, a heuristic method such as PSO, GMDH, and others. The GMDH is a heuristic method of self-organization to formulate a complex system. Prediction using GMDH models has been utilized in many domains and thus includes chemical [9], geology [10], environment [11], and others. Considerably, the forecasting models are tested by various technique and GMDH is an effective technique which enhances more prediction accuracy. Other forecasting techniques such as short-term load using GMDH was conducted for Australia and produced an easier and more efficient method [12]. The correlations between input and target for non-linear function can be expressed by a series of complex polynomials in the Volterra series as follows [13]:

$$\begin{aligned} \boldsymbol{y(p)} = & \int_{\boldsymbol{0}}^{\boldsymbol{p}} \boldsymbol{h_i(\tau)x(p-\tau)dx} \\ & + \int_{\boldsymbol{0}}^{\boldsymbol{p}}\int_{\boldsymbol{0}}^{\boldsymbol{p}} \boldsymbol{h_2(\tau_1\,\tau_2)x(p} \\ & \boldsymbol{-\tau_1)x(p-p_2)d\tau_1 d\tau_2} \\ & + \int_{\boldsymbol{0}}^{\boldsymbol{p}}\int_{\boldsymbol{0}}^{\boldsymbol{p}}\int_{\boldsymbol{0}}^{\boldsymbol{p}} \boldsymbol{h_3\,(\tau_1\tau_2\,\tau_3)x(p} \\ & \boldsymbol{-\tau_1)x(p-\tau_2)x(p} \\ & \boldsymbol{-\tau_3)d\tau_1 d\tau_2 d\tau_3}+\ldots \end{aligned} \quad [1]$$

From equation [1], the description between the relation of input and target (output) parameters and can be rewritten by the complex discrete form of the Volterra final series:

$$\begin{aligned} y_k = & a_0 + \sum_{i}^{n} a_i x_i \\ & + \sum_{i}^{n}\sum_{j}^{n} a x_i x_j \\ & + \sum_{i}^{n}\sum_{j}^{n}\sum_{k}^{n} a_1 a_i x_j x_{kj} + \cdots \end{aligned} \quad [2]$$

From equation [1] and [2] and by simplified into partial quadratic polynomial system consist of two parameters by equation [3]; The GMDH algorithm procedure can be performed as described in [9]; to choose the input variables that affect the target variable by normalize the data $(x_1, x_2, x_3 \ldots, x_p)$ to be entered for Q input-output data into training data and testing data. Secondly, by generating double variable called second order partial description (PD). $G_k = a_0 + a_1 x_p + a_2 x_q + a_3 x_p x_q + a_4 {x^2}_p + a_5 {x^2}_q + \cdots$ where $a_0 + a_1 \ldots . a_5$ are coefficient, and G_k is a centroid variable. Next, we estimate the coefficients $a_0 + a_1 \ldots . . a_5$ using linear regression model. The coefficient is obtained from the smallest to the equation as described:

$$E_L = \sum_{i=1}^{n_{train}} (y_i - G_{Li})^2, L = 1 \cdots, \frac{m(m-1)}{2} \quad [3]$$

We have to count squared error by using the coefficient obtained from the previous computation, the smallest error be the best, if the polynomial and intermediate variable have fully filled the condition, then end the algorithm. To build the next layer using new input-output data and by repeating the process starting from the second function (equation 4). Predicating performance evaluation can be done by calculating the mean absolute percentage error (MAPE) by forecasting results of the GMDH models with conventional methods of Sumatera, to compare the best results use Mean Absolute Percentage Error (MAPE) as:

$$E_{MAPE(i)} = \frac{1}{N}\left[\sum_{p=1}^{N}\left|\frac{\bar{y}(i,p) - y(i,p)}{y(i,t)}\right|\right] * 100 \quad [4]$$

The results of the predictive data explains that the smallest error show the model is better and accuracy level approach to the real value of the weather. Where N is the number of time periods, $\bar{y}(i,t)$ is the predictive period, and $y(i,t)$ is the real data in the period. The accuracy of the model can be computed by statistically assessing the accuracy of the model as absolute fraction of variance R^2, root-mean squared error ($RMSE$), mean squared error (MSE), and the mean absolute deviation (MAD) which is defined by the mathematical models:

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{M}(Y_{i(model)} - Y_{i(real\ value)^2}}{\sum_{i=1}^{M}(Y_{i(real\ value)^2}}\right] \quad [5]$$

$$RMSE^2 = \left[\frac{\sum_{i=1}^{M} Y_{i(model)} - Y_{i(real)^2}}{M}\right] \quad [6]$$

$$MSE = \sum_{i=1}^{M} \frac{\left(Y_{i(model)} - Y_{i(real\ values)^2}\right)}{M} \quad [7]$$

$$MAD = \sum_{i=1}^{M} \frac{\left(Y_{i(model)} - Y_{i(real\ values)}\right)}{M} \qquad [8]$$

2.2 Multilayered Neural Nets (MNN)

Neurons are measured as the binary, two, or three equilibrium states components of the neuro net. We use the GMDH algorithms as complex neurons and not as binary, where its organization processes are well evaluated. In the neuro net with such neurons, there is a two-fold multilayered structure: by default, neurons are multilayered, and they can be united into a common matrix in the multilayered technique. The GMDH algorithms are a typical example of complex active neurons since they choose the effective inputs and corresponding coefficients by themselves and in the process of it default-organization [14,15,16].

However, each of these neurons is an uncomplicated system that handles the sample task. The main function or objective is to combine many neurons into a network and to achieve better accuracy for the assigned task using input data. In the default organization of a neural network, the exhaustive search by first applying and determining the number of neurons layer and the sets of input and output variables for each neuron. The smaller the discriminating criterion suggests that the variable for which the design and development of the neural network are beneficial and account for the number of neuro net layers should be used [17]. Furthermore, an active neuron is able, during the default-organizing process, and to approximate which inputs are essential to mitigate the given objective function of the neuron. The process can provide the generation of new effective characteristics of the inputs of neurons from the previous layer (special type) and the ability to choose an effective set of factors at each layer of the neurons. Mathematically, the number of active neurons in each layer is equal to the number of variables given in proposed data sampling and thus similar to Kalman's filter: the output set of variables is recurrent to the input set but with filtration of noises.

3 MODEL ASSESSMENT AND SELECTION

3.1 Bias, Variance and Model Complexity

If we consider first the case of a quantitative or interval scale response, a target variable, a vector of inputs, and a prediction model that has been estimated from a training set. The loss function for measuring errors between Y and is represented by and typical choices as thus:

$$L\left(Y, \hat{f}(x)\right) = \left(Y - \hat{f}(x)\right)^2 \dots\dots squared\ error \qquad [9]$$

$$L\left(Y, \hat{f}(x)\right) = \left|Y - \hat{f}(x)\right| \dots\dots absolute\ error$$

The behaviour of test sample and training sample error as the model complexity is varied. The light green curves show the training error err, while the light red curves show the conditional test error Err_T for 2,000 training sets of size 100 each, as the model complexity is increased the solid curves show the expected test error and the expected training error $E[\overline{err}]$. Test error can also be referred to as generalization error is the prediction error over an independent test sample.

$$Err_T = E\left[L\left(Y, \hat{f}(x)\right)|T\right] \qquad [10]$$

Where both X and Y are drawn randomly from their joint distribution (population). In our training set T is fixed, and test error refers to the error for this specific training set. A related quantity is the expected prediction error (or expected test error).

$$Err = E\left[L\left(Y, \hat{f}(x)\right)\right] = E[Err_T] \qquad [11]$$

Estimation Err_T is our goal, although we will see that Err is more amenable to statistical analysis, and most methods effectively estimate the expected error. For conditional estimate error, it does not seem possible or effective which is given only the information in the same training set. To compute the training error', is the average loss over the training sample of the dataset; $\overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$. As the model becomes more and more complex, it uses the training data more and can adapt to more complicated underlying structures and thus decrease in bias but an increase in variance. Training error consistently decreases with model complexity, typically dropping to zero, if we increase the model complexity enough. However, a model with zero training error is overfitted to the training data and typically generalizes poorly.

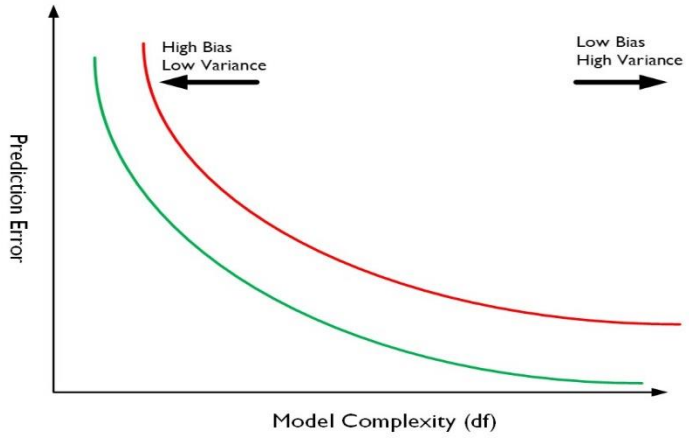


Fig.1: Prediction error vs selection model complexity

In this section we describe several methods for estimating the expected test error for a model. Typically, our model will have a tuning parameter(s) α and so we can write our predictions as $\hat{f}_{\alpha}(x)$. The tuning parameters varies the complexity of our model, and we wish to find the value of α that minimizes error, that is to produce the minimum of the average test error curve in Fig. 1. For the purpose of brevity, we often suppress the dependence of $\hat{f}(x)$ on α. In our model selection there are two important goals that we denote:

Model Selection: estimating the performance of different models in order to choose the best one. In terms of experimenting with the model, Fig. 2 shows a typical example of how GMDH techniques are used to train the model prediction process.

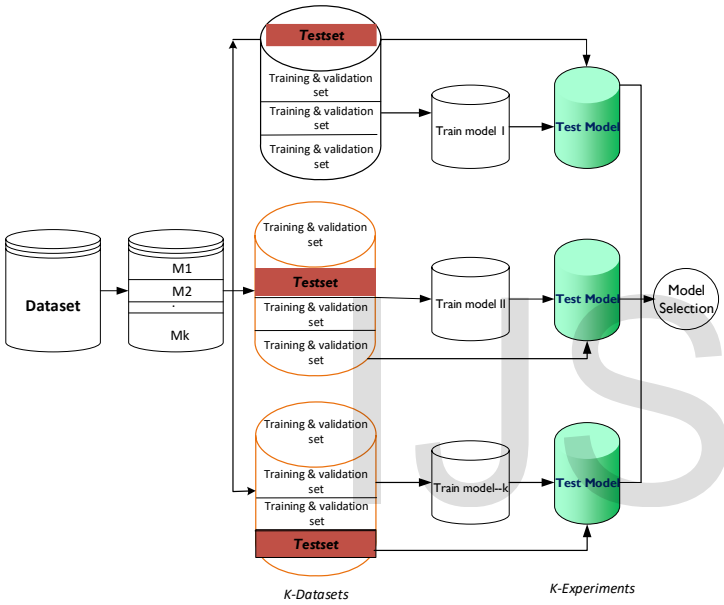


Fig. 2: An experimental model for the model selection process

Model Assessment: having chosen a final model, estimating its prediction error or generalization error on new data. In the situation of data-rich, the best approach for each problem is to randomly divide the dataset into three parts: a training set, a validation set, and a testing set. The training set is used to fit the models; the validation set is used to estimate prediction error for model selection; the testing set is used for assessment of the generalization error of the final chosen model. Ideally, the test set should be kept in a "vault" and be brought out only at the end of the data analysis. Suppose instead that we use the test-set repeatedly, choosing the model with the smallest test-set error. Then the test-set error of the final chosen model will underestimate the true test error sometimes substantially. It is challenging to give a general rule on how to choose the number of observations in each of the three parts as this depends on the signal-to-noise ratio in the data and the training sample size.

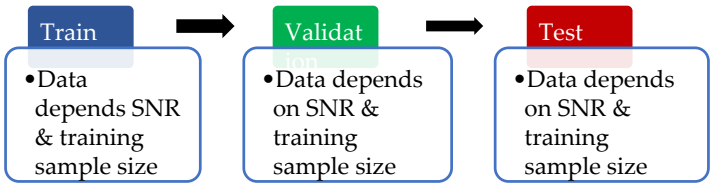


Fig. 3: A typical split for training, for validation and testing data

The techniques in this paper are designed for situations where there is insufficient data to split it into three parts (see Fig. 2). Apparently, it is difficult to give a general rule on how much training data is enough, among other things, this depends on the signal-to-noise ratio (SNR) of the underlying function, and the complexity of the models being fit to the data.

3.2 The Bias-Variance Decomposition

If we assume that $Y = f(X) + \varepsilon$ where $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma_{\varepsilon}^2$, we can derive an expression for the expected prediction error of a regression fit $\hat{f}(X)$ at an input point $X = x_0$, using squared error loss:

$$E_{rr}(x_0) = E\left[\left(Y - \hat{f}(x_0)\right)^2 | X = x_0\right]$$

$$E_{rr}(x_0) = \sigma_{\varepsilon}^2 + \left[E\hat{f}(x_0) - f(x_0)\right]^2 + E\left[\hat{f}(x_0) - E\hat{f}(x_0)\right]^2$$

$$E_{rr}(x_0) = \sigma_{\varepsilon}^2 + Bias^2\left(\hat{f}(x_0)\right) + Var\left(\hat{f}(x_0)\right)$$

$$E_{rr}(x_0) = Irreducible\ Error + Bias^2 + Variance \qquad [12]$$

The first parameter function in the *variance of the target around its true mean* $f(x_0)$, unless $\sigma_{\varepsilon}^2 = 0$. The second parameter function is the *squared bias*, the amount by which the average of our estimate differs from the true mean; the third parameter function is the *variance*, the expected squared deviation of $\hat{f}(x_0)$ around its mean. Ideally the more complex we make the model $\hat{f}$, the lower the (squared) bias and the higher the variance.

For a linear model family such as ridge regression, we can break down the bias more finely. Let γ_* denote the parameters of the best-fitting linear approximation to f:

$$\gamma_* = arg\ arg_{\gamma}^{min} E(f(X) - X^T\gamma)^2 \qquad [13]$$

Where X is the input variables. Then we can write the average squared as:

$$E_{x0}\left[f(x_0) - E\widehat{f_\alpha}(x_0)\right]^2 \quad [14]$$
$$= E_{x0}[f(x_0) - x_0^T\gamma]^2 + E_{x0}[x_0^T\gamma_* - E_{x0}^T\widehat{\gamma_\alpha}]^2$$
$$= Ave[Model\ Bias]^2 + Ave[Estimation\ Bias]^2$$

The first parameter function on the right-hand side is the average squared *model bias*, the error between the best-fitting linear approximation and the true function. The second parameter function is the average squared *estimation bias*, the error between the average estimate $E(x_0^T\hat{\gamma})$ and the best-fitting linear approximation.

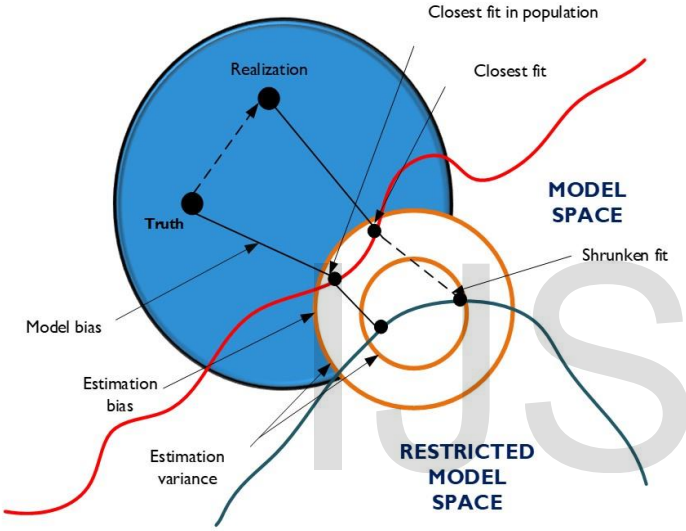


Fig. 3: Schematic of the behaviour of bias and variance

For linear models fit by ordinary least squares, the estimation bias is zero. For restricted fits, such as ridge regression it is positive, and we trade it off with the benefits of a reduced variance. The model bias can only be reduced by enlarging the class of linear models to a richer collection of models, by including interactions and transformations of the variables in the model. In regard to the linear models, the model space is the set of all linear predictions from p inputs, and the *black dot* labeled "closest fit" is $x^T\gamma_*$. The blue-shaded region indicates the error with which we see the truth in the training sample. Schematic of the behaviour of bias and variance (Fig. 3), the model space is the set of all possible predictions from the model with the "closest fit" labeled with a black dot. The model bias from the truth is shown, along with the variance indicated by the large yellow circle centred at the black dot labeled "closest fit in population". A shrunken or regularized fit also shows, having additional estimation bias, but smaller prediction error due to its decreased variance.

4 THE PROPOSED ARCHITECTURE OF THE GMDH-NLRMN NON-REGRESSION MODEL FOR FORECASTING

As indicated in Fig. 4, the neural network with GMDH architecture has five inputs. The number of nodes in the second layer is determined to be ten (10). Furthermore, since the approach is a nonlinear model, the expected output is six (6) nodes in layer 4 and only one selected node as the main output of the algorithm. The coefficients of equation 2 are estimated in each neuron. By using the estimated coefficients and input variables in each neuron, the desired output is predicted accordingly. In this research, predicted mean square error (MSE) is used as the external benchmarks. The outputs obtained from selected neurons become the inputs for the next layer. This process continues until the last layer is determined. In the end, in the last layer, only one neuron is selected. The obtained output from the last layer is the predicted value for the time series in hand. The block diagram of the algorithm is represented in Fig. 5. The algorithm is implemented in MATLAB to better understand the performance of the proposed model.

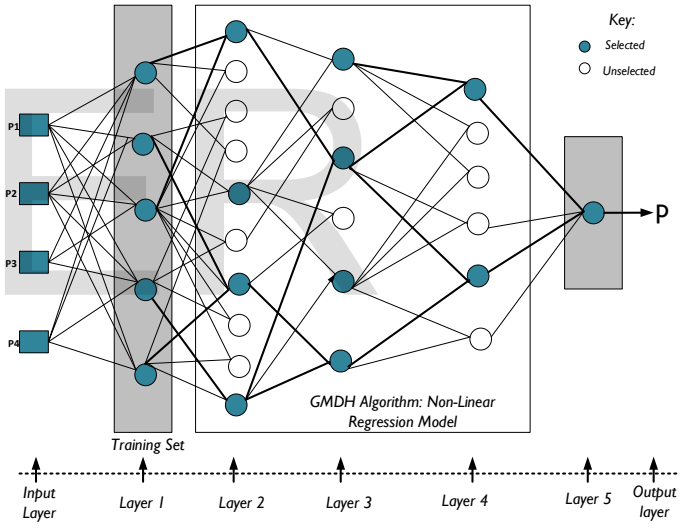


Fig. 5: Architecture of neural network using GMDH-NLRMN for forecasting nonlinear machine learning model

5 EXPERIMENTAL ANALYSIS AND RESULTS

The study was conducted under several tests to determine the performance and accuracy of the GMDH network. Based on the analysis obtained in Table 1, as the number of neurons increases, the minimum error decreases and hence the better the results. However, due to the permutation of data analytics, the result obtained differently as more analyses are implemented. The result shows that in Table 1 and Fig. 6 train data slightly differ from the first test, second test, and so on until all the data is trained to actualize the performance and accuracy of the model (maximum layer of the neurons is trained). Considering the trained data from Fig. 6 (right)

indicates that the value of MSE, RMSE, Error Mean, and Standard Deviation is 0.011168, 0.10568, -0.00072732, and 0.10597, respectively. The prediction for the regression model is 99.3% and the distribution of the trained data is indicated in Fig. 6 (left).

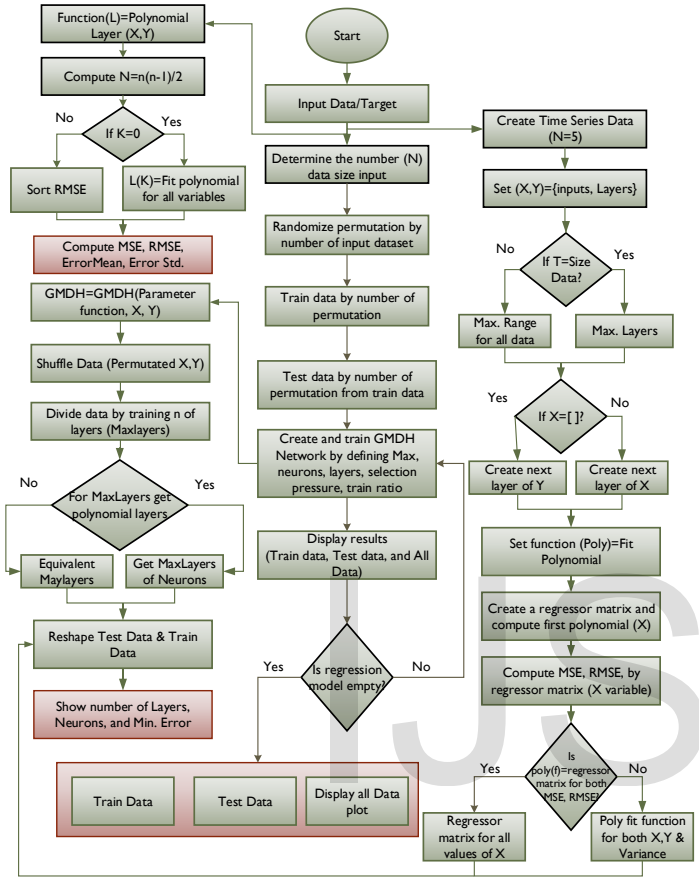
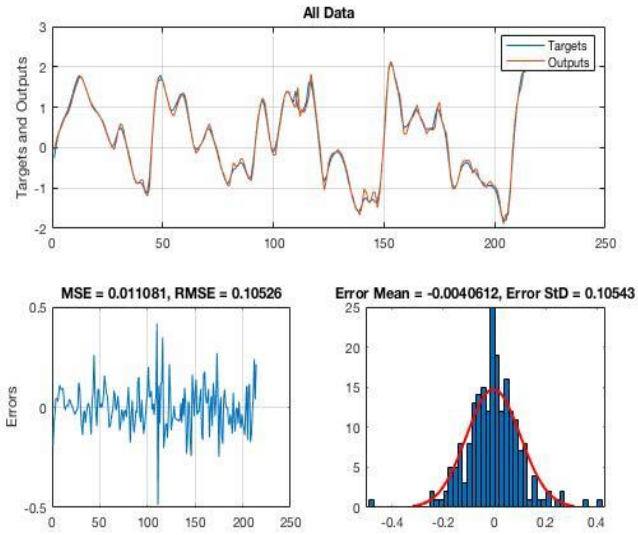


Fig. 5: A block-diagram for proposed GMDH-NLRMN neural network technique

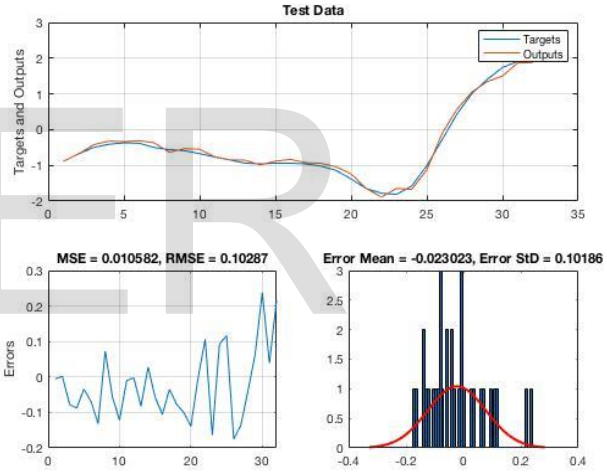
Table 1: GMDH Network with a 250-maximum layer of neurons and 5 maximum layers

Max. Layer of Neurons	250
Number of Layers	5
Selection Pressure	6
Train Ratio	0.85 (85%)

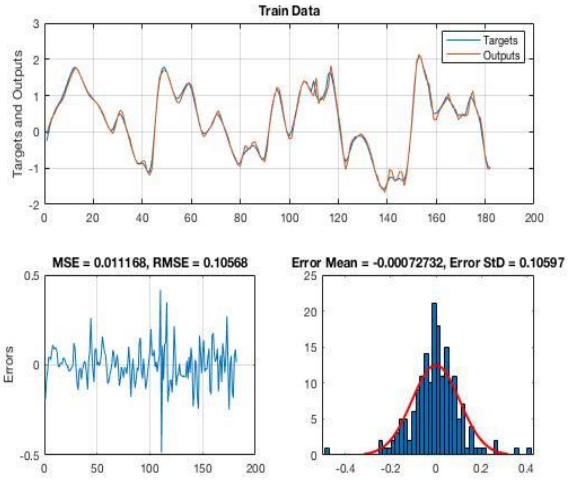
Layer Number	1st Test		2nd Test/Multiple Test	
	Number of Neurons	Min. Error	Number of Neurons	Min. Error
1	6	0.11024	5	0.90995
2	13	0.0907	6	0.079761
3	58	0.085133	14	0.076064
4	250	0.080905	40	0.070071
5	1	0.074628	1	0.069719



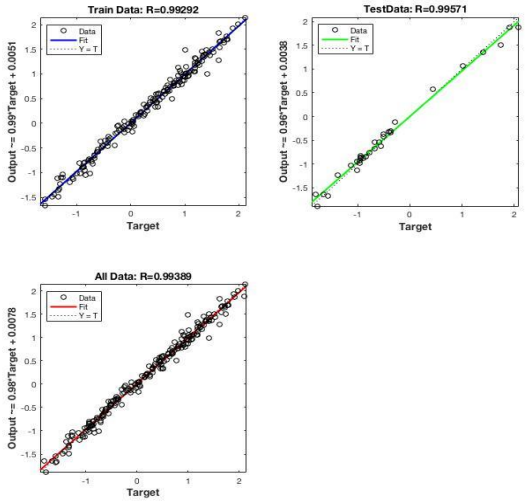
(a) The target and output for all trained data



(b) The test data



(c) Trained data results with a MSE of 0.011168



(d) Predicted regression value for the trained, test and all data

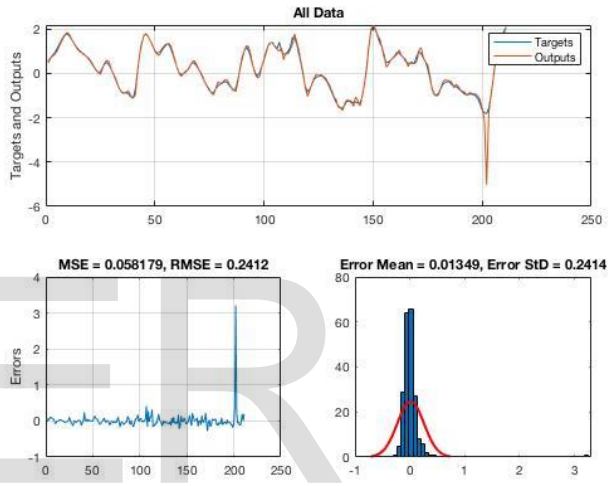
Fig. 6: Experimental results of the first trained data (a)-trained data for all, (b)-test data, (c)-test data, and (d)-trained predicted regression value

Considering the number of parameters that were inserted into the network; maximum layer = 300, number of layers = 8, selection pressure = 0.7, and train ratio = 0.90 (90%), to test the prediction accuracy and performance of the GMDH network as proposed in Fig. 5. The analysis indicated that as the number of neurons increases, the minimum error decreases, and thus the better the performance and accuracy until the maximum threshold layer is achieved. Several tests were trained to determine the performance of the GMDH algorithm. However, as indicated in first training data (Table 1 and Table 2), result changes slightly as the more and more the network is trained until the maximum input layer is reached. Furthermore, as the parameter values increases and the increasing value, the computational complexity increases and better performance of the GMDH algorithm. Inherently, with a watermarking algorithm proposed by Fofanah A.J and Gao T. [18] where genetic programming was fully utilized and for any generation of computation the better the performance of the algorithm. Conversely, a neuro-fussy for non-linear regression model was also proposed for mobile online learning framework of which the network the performance and prediction accuracy are congregant by Fofanah A.J et al. [19].

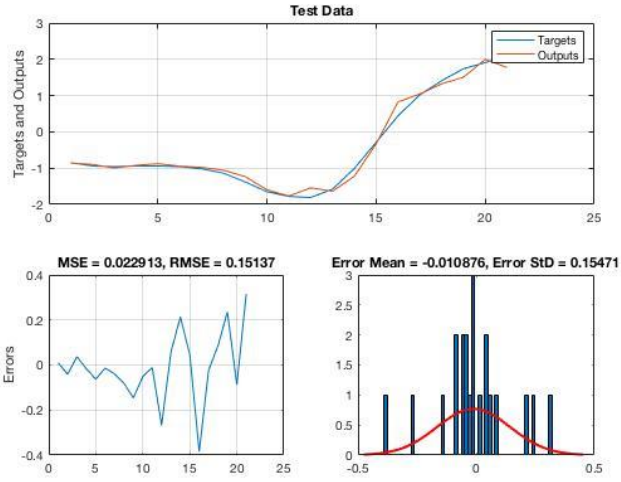
Table 2: GMDH Network with a 300-maximum layer of neurons and 8 maximum layers

Max. Layer of Neurons	300
Number of Layers	8
Selection Pressure	0.7
Train Ratio	0.90 (90%)

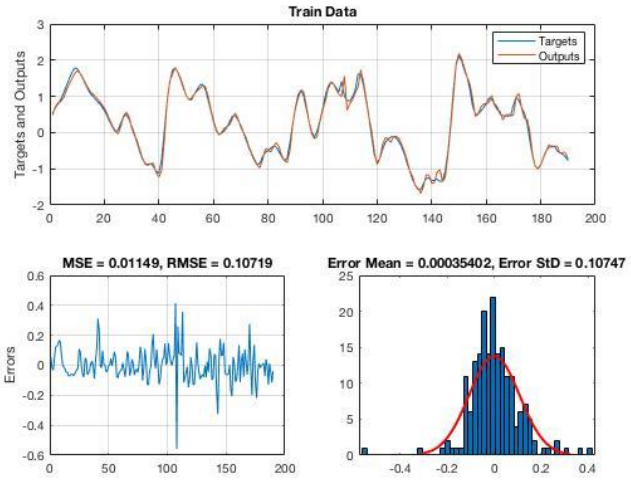
Layer Number	1st Test		2nd Test/Multiple Test	
	Number of Neurons	Min. Error	Number of Neurons	Min. Error
1	8	0.10306	11	0.11393
2	18	0.098246	24	0.081423
3	102	0.084376	48	0.07738
4	250	0.079252	134	0.07356
5	250	0.074391	250	0.074139
6	250	0.070574	250	0.071448
7	250	0.067876	250	0.069129
8	1	0.06633	1	0.06591



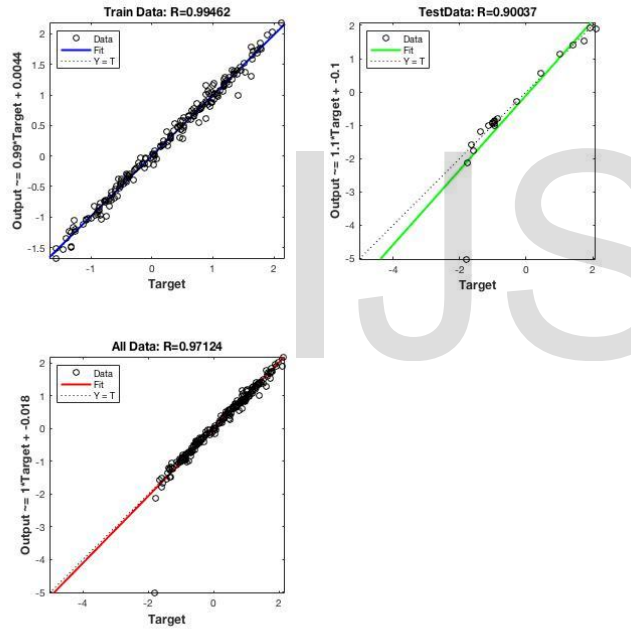
(a) The target and output for all trained data



(b) The test data



(c) Trained data results with a MSE of 0.01149



(d) Predicted regression value for the trained, test and all data

Fig. 7: Experimental results of the first trained data (a)-trained data for all, (b)-test data, (c)-test data, and (d)-trained predicted regression value

6 Conclusions

The GMDH technique has been tested and evaluated using time series non-linear regression model for various parameter values to determine the robustness and accuracy of the neural network. The proposed algorithm (block-diagram) using the GMDH network indicates that as the number of neurons increases and inserted into the network the more robust and better accuracy of the prediction model. However, the parameter lists (neurons, layers, selection pressure, and train ratio) are significant in determining the performance and prediction accuracy of the model. Furthermore, the proposed algorithm performed better with a prediction accuracy of 99.2% and 99.4% for the first and second parameter value selections, respectively. The contribution of this paper is presented in three (3) main folds; the design and development of a mathematical model and architecture for the GMDH network to existing algorithms of the GMDH techniques; proposed algorithm (block-diagram) using different datasets to determine the prediction accuracy and performance; and the pattern of behavioural bias and variance of the model when implemented in MATLAB programming congregant with genetic programming algorithms.

Acknowledgment

The authors wish to thank the administration of the Milton Margai College of Education and Technology research grant. This work was supported in part by a grant from the MMCET research directorate.

References

[1] Zhao, H., & Guo, S. (2016). An optimized grey model for annual power load forecasting. Energy, 107, 272-286.

[2] Koo, B. G., Lee, H. S., & Park, J. (2015). Short-Term Electric Load Forecasting Based on Wavelet Transform And GMDH. Journal of Electrical Engineering & Technology, 10(3), 832-837.

[3] Niu, D., Wang, Y., & Wu, D. D. (2010). Power load forecasting using support vector machine and ant colony optimization. Expert Systems with Applications,37(3), 2531-2539.

[4] Panapakidis, I. P., & Dagoumas, A. S. (2016). Day-ahead electricity price forecasting via the application of artificial neural network-based models. Applied Energy, 172, 132-151.

[5] Xu, Lixiang et al. (2020). "Probabilistic SVM classifier ensemble selection based on GMDH-type neural network". In: Pattern Recognition106, p. 107373.

[6] Ivakhnenko, Alexey Grigorevich and Johann-Adolf Muller (n.d.). "Dollar Rate Fore-casting by Group Method of Data Handling (GMDH) Algorithms and Accuracy Increase by Twice Multilayered Networks". In: JR, QUINLAN (1986). "Induction of decision trees". In: Machine learning1, pp. 81–106.

[7] Andrakhanov, AA (2013). "Navigation of autonomous mobile robot in homo generous and heterogeneous environments on basis of GMDH neural networks". In: Proceedings of 4th International Conference on Inductive Modelling (ICIM-2013), pp. 133–138.

[8] Andrakhanov, AA (2013). "Navigation of autonomous

mobile robot in homo generous and heterogeneous environments on basis of GMDH neural networks". In: Proceedings of 4th International Conference on Inductive Modelling (ICIM-2013), pp. 133–138.

[9] Ashrafzadeh, Afshin et al. (2020). "Comparative study of time series models, support vector machines, and GMDH in forecasting long-term evapotranspiration ratesin northern Iran". In: Journal of Irrigation and Drainage Engineering146.6, p. 04020010.

[10] Xu, H., Dong, Y., Wu, J., & Zhao, W. (2012). Application of GMDH to short-term load forecasting. In Advances in Intelligent Systems (pp. 27-32). Springer Berlin Heidelberg.

[11] Zhang, H., Liu, X., Cai, E., Huang, G., & Ding, C. (2013). Integration of dynamic rainfall data with environmental factors to forecast debris flow using an improved GMDH model. Computers & Geosciences, 56, 23-31.

[12] Kordnaeij, A., Kalantary, F., Kordtabar, B., & Mola-Abasi, H. (2015). Prediction of recompression index using GMDH-type neural network based on geotechnical soil properties. Soils and Foundations, 55(6), 1335-1345.

[13] Atashrouz, S., Pazuki, G., & Kakhki, S. S. (2015). A GMDH-type neural network for prediction of water activity in glycol and Poly (ethylene glycol) solutions. Journal of Molecular Liquids, 202, 95-100.

[14] Ivakhnenko, Alexey Grigorevich and Johann-Adolf Muller (n.d.). "Dollar Rate Fore-casting by Group Method of Data Handling (GMDH) Algorithms and Accuracy Increase by Twice Multilayered Networks". In: JR, QUINLAN (1986). "Induction of decision trees". In: Machine learning1, pp. 81–106.

[15] Kondo, Tadashi and Junji Ueno (2008). "Multi-layered GMDH-type neural net-work self-selecting optimum neural network architecture and its application to3-dimensional medical image recognition of blood vessels". In: International Journal of innovative computing, information and control4.1, pp. 175–187.

[16] Ivakhnenko, AG, GA Ivakhnenko, and JA Muller (1994). "Self-organization of neural networks with active neurons". In: Pattern Recognition and Image Analysis4.2, pp. 185–196.

[17] Nariman-zadeh, N et al. (2008). "Robust Pareto Design of GMDH-type Neural Net-works for Systems with Probabilistic Uncertainties". In:2nd International Conference on Inductive Modelling, p. 91.

[18] Fofanah, A.J. and Gao, T., 2020, May. Dual Watermarking for Protection of Medical Images based on Watermarking of Frequency Domain and Genetic Programming. In Proceedings of the 2020 the 4th International Conference on Innovation in Artificial Intelligence (pp. 106-115).

[19] Fofanah, A.J., Kalokoh, I., Hwase, K.T. and Namagonya, A.P., Adaptive Neuro-Fuzzy Inference System with Non-Linear Regression Model for Online Learning Framework.