Original Research Article

Multicollinearity Effect in Regression Analysis: A Feed Forward Artificial Neural Network Approach

ABSTRACT

In this study we compared the performance of Ordinary Least Squares Regression (OLSR) and the Artificial Neural Network (ANN) in the presence of multicollinearity using two datasets – a real life insurance data and a simulated data – to know which of the methods, models a highly correlated dataset better using the Root Mean Square Error (RMSE) as the performance measure. The ANN performed better than the OLSR model for all the different ANN models except the models with nine and ten nodes in the hidden layer for the real life data. The network with four hidden nodes was the best model. For the simulated data, the ANN model with two hidden nodes gave us the least RMSE when compared to the OLSR model and the other ANN models in the testing set. The network with two hidden nodes modelled the data very well. In the presence of multicollinearity, ANN model achieves a better fit and forecast than the OLSR.

Keywords: Multicollinearity; Ordinary Least Squares; Artificial Neural Network; Root Mean Square Error.

1.0 INTRODUCTION

In modelling a linear relationship between a dependent variable and one or more independent variables, OLSR is being used to estimate the parameters of the model by minimizing the Residual Sum of Squares. The OLSR gives an unbiased estimate of the regression coefficients, it is very easy to compute and interpret. Though OLSR is preferred, it can only yield the best results when some assumptions are satisfied; There must be a linear relationship between each of the independent variables and the dependent variable, the independent variables must not be highly correlated, the variance of the error must be constant, the errors must not be correlated and there must not be an outlier. Most real life data does not always satisfy all the assumptions of the OLSR and if we insist on using the OLSR method to estimate the parameters, we will not be able to achieve a better fit for the data and a good prediction with the model. Wonsuk *et al.* (2014), Onoghojobi *et al.* (2016) and Olewuezi *et al.* (2016) studied the nature of multicollinearity, their consequences, how to detect them and some remedial measures that can be taken to get a good estimate of the regression coefficients.

In this study, we considered a solution to the OLSR method when the multicollinearity assumption is not satisfied. In the presence of multicollinearity, it is impossible to estimate the unique effects of individual variables in the regression equation, the variance and covariance of the Least Squares (LS) estimates become too large though still the Best Linear Unbiased Estimator (BLUE), most of the regression coefficients are not significant and there is a high R^2 value even though the t values for most of the regression coefficients are small. Multicollinearity becomes one of the serious problems in linear regression analysis, Yazid and Mowafaq (2009). Many attempts have been made to improve the OLSR estimation procedure, some of which are Ridge Regression Onoghojobi *et al.* (2016), Latent Root Regression, Partial Least Squares Olewuezi *et al.* (2016), Principal Component Regression Onoghojobi *et al.* (2016) etc. and more recently, machine learning method which have smaller Mean Square Error (MSE) than the OLSR method Zou *et al.* (2008).

Artificial Neural Network (ANN) is an example of a machine learning method that evolved from the idea of simulating the human brain Zou *et al.* (2008). They are networks of simple processing elements called neurons or nodes. The ANN models complex nonlinear relationships between the predictor variables and the response with great flexibility by defining input neurons – nodes – which are the predictor variables, a hidden layer with a number of nodes connected to each of the input nodes and lastly, an output layer with one or more nodes. The theoretical advantage of ANNs is that the relationship between the variables need not to be specified in advance since the method establishes the relationship through a learning process. The model learns the relationship from the data used to train it. The ANNs do not also require any assumptions about the underlying population distribution. Hsiao-Tien (2008) and Kumar (2005) compared the performance of ANN and OLSR. Hsiao-Tien (2008) compared OLSR and ANN models with seven explanatory variables of corporation's feature and three external macro-economic control variables to analyse the important determinants of capital structures of the high-tech and traditional industries respectively in Taiwan. He used the RMSE as the criterion to know the best model. The ANN model achieved a better fit and forecast than the OLSR model as it had the least RMSE. Ramirez *et al.* (2000) also compared ANN and OLSR model. They found out that the ANN method performed better than the OLSR, although both methods showed good performance for daily rainfall. Asep *et al.* (2015) applied ANN-Linear perceptron in the development of decision support system for a fishery industry and compared the result with Multiple Linear Regression-OLS (MLR-OLS). They discovered that the ANN-LP is as good as the MLR-OLS in estimating both the growth parameters and the maximum sustainable yield of the fishery and can be used in situations when the MLR-OLS is unable or difficult to find the estimate of the parameters. Soukayna and Jan (2015) predicted the Standardized Precipitation and Evapotranspiration Index (SPEI) using OLSR and ANN in Wilson Promontory Station, Victoria, Canada. They compared the performance of both models using the coefficient of determination and the RMSE. The ANN model provided greater accuracy than the OLSR in forecasting the 1, 3, 6 and 12 months SPEI. Ebiendele and Ebiendele (2018) also compared OLSR and ANN models in seasonal rainfall prediction in North East Nigeria using four performance criteria. The results showed that the ANN model performed better than the OLSR model. The ANN had the minimum mean absolute error, RMSE and prediction error, and the highest coefficient of determination.

Aysenur *et al.* (2019) investigated colorimetric parameters and mass loss of heat-treated bamboo and modelled the results gotten using OLSR and ANN. They used two predictory variables (temperature and duration of heat treatment) on both methods and compared the results using the Mean Absolute Percentage Error (MAPE). The ANN method gave more accurate results than the OLSR method. Ilaboya and Igbinedion (2019) investigated the capability of linear (OLSR) and non-linear (ANN) regression techniques for long-term rainfall prediction. They restricted their study to Benin City, Nigeria. The ANN method also performed better than the OLSR using the coefficient of determination as the performance measure. This paper compares the performance of ANN and OLSR method to model a highly correlated real life Nigerian Insurance Company's data and a simulated data.

2.0 METHODOLOGY

The OLSR and ANN were used to model the two datasets to know which of the methods, models a highly correlated dataset better using the RMSE as the performance measure. The model with the least RMSE is chosen as the best model. Correlation coefficient is used to test for multicollinearity in the two datasets. There is high multicollinearity in the data if the correlation coefficient is high (i.e. greater than 0.8 or less than – 0.8).

2.1 ORDINARY LEAST SQUARES METHOD.

Let us consider the standard model for Multiple Regression Analysis

$$Y = X\beta + \varepsilon \tag{1}$$

where

Y is $(n \times 1)$ vector of the dependent variable.

X is $(n \times p)$ matrix of independent variables.

β is $(p \times 1)$ vector of regression parameters.

ε is $(n \times 1)$ vector of errors.

From equation (1), we have

$$\varepsilon^T \varepsilon = (Y - X\beta)^T (Y - X\beta) \quad (2)$$

This term is differentiated with respect to β and set equal to 0 to obtain an estimate of β provided the inverse of $X^T X$ exists and is unique. We therefore have:

$$\hat{\beta}_{OLS} = \left(X^{'} X\right)^{-1} X^T Y \quad (3)$$

where β_{OLS} is $p \times 1$ vector of OLSR estimated parameters.

2.2 ARTIFICIAL NEURAL NETWORK (ANN)

The ANN models complex nonlinear relationships between the predictor variables and the response with great flexibility by defining input neurons – nodes – which are the predictor variables, a hidden layer with a number of nodes connected to each of the input nodes and lastly, an output layer with one or more nodes. An activation function is applied to both the hidden and the output layers. The connections between the nodes (input nodes and the hidden layer nodes) are assigned weights. The weights are the parameters the Neural Network estimates, and they are chosen so as to minimize a pre-defined loss function. Neural Network tries to minimize the difference between the observed responses and the output. Figure 1 is an example of an Artificial Neural Network. Three layers of nodes are defined – an input layer that comprises of three input nodes and a bias node, a single hidden layer and an output layer.

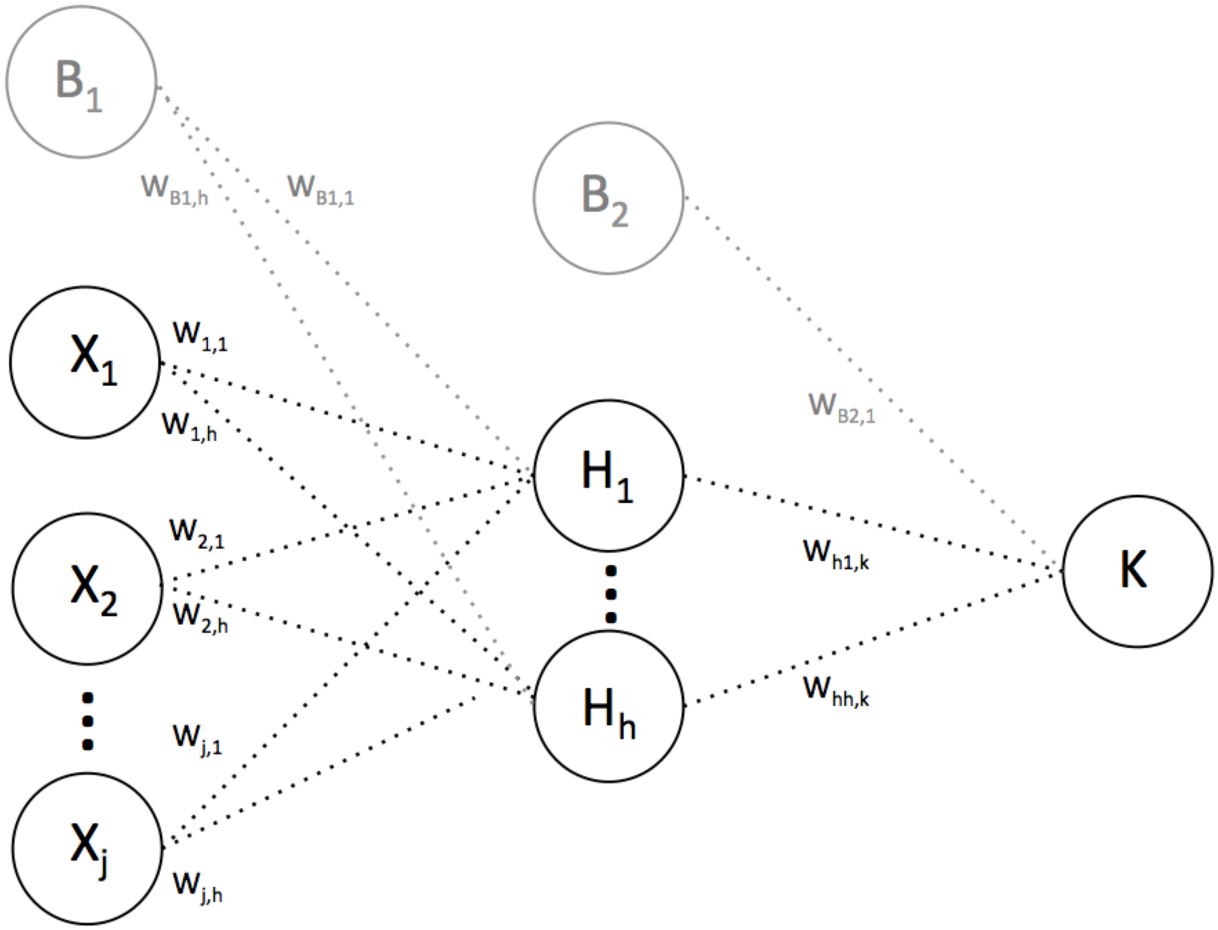
Let X represent the inputs,

H, K and B represent the hidden, output and bias nodes respectively, and

W represent the weights.

The weights in the bias nodes can be interpreted similarly to an intercept in a linear regression.

Figure 1. Single hidden layer feed forward neural network.



A Feed Forward Neural Network (FFNN) is a uni-directional connection from the input to the hidden layer and from the hidden to the output layer. A mathematical representation of a single layer FFNN is given in equation (4).

$$\hat{y}_k(x_i, w) = \Phi_0(\alpha_k + \sum_{h=1}^{H} w_{hk}\Phi_h(\alpha_h + \sum_{j=1}^{J} w_{jh}x_{ij})) \quad (4)$$

That is, the sum of the product of the weights w_{jh} and the inputs x_{ij} plus a bias term α_h gives us a node in the hidden layer. An activation function is applied to each node. An activation function also called a threshold or transfer function is a non-linear transformation applied to the input. The sum is taken over the hidden neurons H of the product of the transformed input and the weights w_{hk} plus a bias term α_k. A final transformation Φ_0 is applied to the output. We have different activation function for both the transmission from the input units to the hidden units and from the hidden units to the output units, namely: linear activation function, unit step activation function, rectified linear unit activation function, hyperbolic tangent activation function, sigmoid activation function, logistic activation function, etc.

The error function is minimized to get an estimate of the weight w for both the input and the hidden layer. The commonly used error function has been the quadratic error function while cross-entropy error function is more suitable for binary classification. The Quadratic error function E_Q and cross entropy error function E_C are given below

$$E_Q = \sum_{k=1}^{K}\sum_{i=1}^{n}(\hat{y}(x_i, w) - y_{ik})^2 \quad (5)$$

$$E_C = -\sum_{k=1}^{K}\sum_{i=1}^{n} y_{ik} \log \hat{y}_k(x_i, w) + (1 - y_{ik}) \log[1 - \hat{y}_k(x_i, w)] \quad (6)$$

The more the nodes in the hidden layer, the more complicated non-linear relationship can be modelled. Increasing the nodes in the hidden layer also increases the likelihood of training an over fitted model. A model is over fitted when it does not generalize well to new observations though it will still perform very well on the training set.

The dataset is divided into two - the training set and the testing set. It should be noted that 70% of the data were used as the training set and the other 30% as the testing set. The training set is used to train the network and the optimally performing hyper parameters are identified. The final model performance is then tested using the testing set.

3.0 EMPIRICAL ILLUSTRATION

3.1 Illustration 1

The real life data used in this study is a secondary data on Nigerian Insurance Expenditure from 1996 to 2011. The data was tested for missing values to ensure a good quality dataset and no missing value was found. Seven quantitative variables namely: Claims, fire, accident, motor, employers, marine, miscellaneous were used as the independent variables with expenditure as the dependent variable. That is,

$$
\begin{aligned}
X_1 &= Claims \\
X_2 &= Fire \\
X_3 &= Accident \\
X_4 &= Motor \\
X_5 &= Employers \\
X_6 &= Marine \\
X_7 &= Miscellaneous \\
Y &= Expenditure
\end{aligned}
$$

The seven independent variables will be regularized by subtracting the variable mean from each of the variables and dividing it by their respective standard deviation to help the neural network to converge quickly.

3.1.1 Test for Multicollinearity

The correlation matrix was used to test for multicollinearity. Table 1 below is the correlation matrix for the data.

Table 1: Correlation Matrix for the Real Life Data

1	0.800667	0.972993	0.9843	0.931225	0.954349	0.818016099
0.800667	1	0.664078	0.802241	0.559947	0.623471	0.330163173
0.972993	0.664078	1	0.956963	0.96467	0.970487	0.888117153
0.9843	0.802241	0.956963	1	0.900635	0.917368	0.763616208
0.931225	0.559947	0.96467	0.900635	1	0.991298	0.948715023
0.954349	0.623471	0.970487	0.917368	0.991298	1	0.930325184
0.818016	0.330163	0.888117	0.763616	0.948715	0.930325	1

From Table 1, there is high multicollinearity in the data since most of the independent variables are highly correlated.

3.1.2 Ordinary Least Squares Regression for the Real Life Data

Table 2: Overall Fit for the Life Data

Multiple R	0.996305
R Square	0.992623
Adjusted R Square	0.986169
Standard Error	2917.984
Observations	16

Table 3: ANOVA Table for the Real Life Data

				Alpha	0.05	
	df	*SS*	*MS*	*F*	*p-value*	*Sig*
Regression	7	9.17E+09	1.31E+09	153.789	6.84E-08	Yes
Residual	8	68117033	8514629			
Total	15	9.23E+09				

Table 4: OLSR Parameter Estimates for the Real Life Data

	Coeff	*std err*	*t stat*	*p-value*	*lower*	*Upper*
Intercept	1346.411	2425.234	0.555167	0.593959	-4246.19	6939.011915
Claims	2.867821	2.354802	1.217861	0.257975	-2.56236	8.298002766
Fire	-3.78929	3.977479	-0.95269	0.368643	-12.9614	5.382793653
Accident	-2.21072	2.9512	-0.74909	0.475249	-9.0162	4.594762006
Motor	0.778697	2.750741	0.283086	0.784298	-5.56452	7.121915705
Employers	-17.4773	57.36528	-0.30467	0.768395	-149.762	114.8072724
Marine	-1.36654	3.986157	-0.34282	0.740566	-10.5586	7.825552293
Miscellaneous	-3.19616	3.414654	-0.93601	0.376657	-11.0704	4.67804833

Although the R-square value (0.9926) for the model is high, all of the regression coefficients are not significant since their p values > 0.05 at 0.05 level of significance and their confidence intervals are large. This contradiction is as a result of the assumption of multicollinearity not being satisfied.

3.1.3 Artificial Neural Network for the Real Life Data

Ten ANN models with different number of nodes in the hidden layer were trained. We used 1 to 10 nodes in the hidden layers to know which one of them will yield the best estimate of the parameters of the network using the RMSE as the performance measure. The logistic activation function was used for the transmission from input units to hidden units and the linear activation function was used for the transmission from hidden units to output units. The quadratic error function was used to determine the weights of the network. Table 5 below gives us the summary of the result.

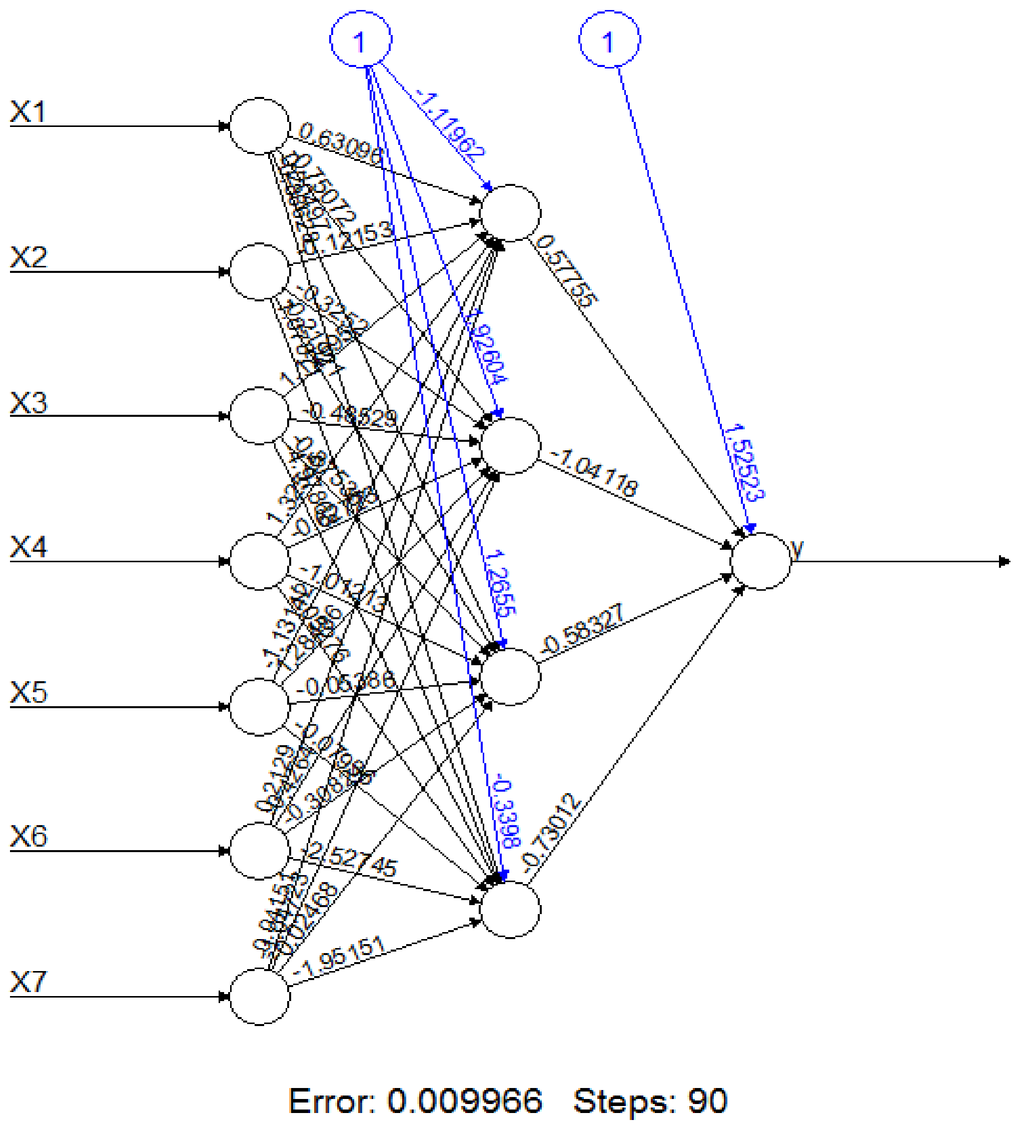
Table 5: RMSE statistics for the Real Life Data

	OLSR	**ANN1**	**ANN2**	**ANN3**	**ANN4**	**ANN5**	**ANN6**	**ANN7**	**ANN8**	**ANN9**	**ANN10**
RMSE (testing)	3188.1	1875.1	2522.9	2105.9	1350.7	2030.2	1526.8	1608.8	2633.7	4512.5	3387.2
RMSE (training)	2786.6	1982.0	2347.9	1781.8	1493.6	2376.6	1663.1	953.1	1916.6	1259.6	1470.3

From Table 5, the ANN models had a lesser RMSE than the OLSR model for all the different models except the models with nine and ten hidden nodes, the ANN models with nine and ten hidden nodes over fitted the training set. It modelled well the training set but could not predict well the testing set. It was observed that ANN performed better than the OLSR model for all the different ANN models except the models with nine and ten hidden nodes. The network with four hidden nodes modelled the data very well than the other ANN models. It did not over fit the training set and also predicts well the testing set. It has the least RMSE when used on the testing set. Below is the estimate of the parameters of the ANN model with four hidden nodes and the graphical representation of the model.

```
Intercept.to.1layhid1 -1.119624897
X1.to.1layhid1         0.630956846
X2.to.1layhid1        -0.121533513
X3.to.1layhid1         1.126946335
X4.to.1layhid1         1.325447728
X5.to.1layhid1        -1.131424271
X6.to.1layhid1         0.212902953
X7.to.1layhid1        -0.041508678
Intercept.to.1layhid2  1.926037650
X1.to.1layhid2        -0.750716229
X2.to.1layhid2        -0.325200164
X3.to.1layhid2        -0.485294045
X4.to.1layhid2        -0.627234654
X5.to.1layhid2         1.284860662
X6.to.1layhid2        -0.426399785
X7.to.1layhid2        -1.047225798
Intercept.to.1layhid3  1.265501183
X1.to.1layhid3         0.204966188
X2.to.1layhid3        -0.219213203
X3.to.1layhid3        -0.875358244
X4.to.1layhid3        -1.012126040
X5.to.1layhid3        -0.053855979
X6.to.1layhid3        -0.308241782
X7.to.1layhid3        -0.024684527
Intercept.to.1layhid4 -0.339802202
X1.to.1layhid4        -0.886280005
X2.to.1layhid4         1.078712791
X3.to.1layhid4        -4.978679098
X4.to.1layhid4        -1.052764500
X5.to.1layhid4        -0.079364780
X6.to.1layhid4        -2.527453252
X7.to.1layhid4        -1.951514060
Intercept.to.y         1.525232613
1layhid1.to.y          0.577550927
1layhid2.to.y         -1.041176296
1layhid3.to.y         -0.583267342
1layhid4.to.y         -0.730115784
```

Figure 2. Single Hidden Layer Feed Forward Neural Network for the Real Life Data.



3.2 Illustration 2

We simulated a correlated data with six independent variables and one dependent variable replicated 150 times.

3.2.1 Test for Multicollinearity

The correlation matrix was used to test for multicollinearity. Below is the correlation matrix for the simulated data.

Table 6: Correlation Matrix for the Simulated Data

1	0.98	0.93	0.89	0.87	0.83
0.98	1	0.95	0.90	0.91	0.77
0.93	0.95	1	0.96	0.84	0.71
0.89	0.90	0.96	1	0.80	0.69
0.87	0.91	0.84	0.80	1	0.67
0.83	0.77	0.71	0.69	0.67	1

From the correlation matrix, high multicollinearity was observed in the data since most of the independent variables were highly correlated.

3.2.2 Ordinary Least Squares Regression for the Simulated Data

Table 7: Overall Fit for the Simulated Data

Multiple R	0.693028
R Square	0.480288
Adjusted R Square	0.458482
Standard Error	9.589799
Observations	150

Table 8: ANOVA Table for the Simulated Data

				Alpha	0.05	
	Df	*SS*	*MS*	*F*	*p-value*	*sig*
Regression	6	12153.3	2025.549	22.0254	3.01E-18	yes
Residual	143	13150.89	91.96425			
Total	149	25304.18				

Table 9: OLSR Parameter Estimates for the Simulated Data

	Coeff	*std err*	*t stat*	*p-value*	*lower*	*upper*
Intercept	-92.7866	159.8667	-0.5804	0.562559	-408.794	223.2207
X_1	-1.4703	5.114539	-0.28748	0.774165	-11.5802	8.639566
X_2	5.355521	6.111732	0.876269	0.382353	-6.72549	17.43653
X_3	-1.18115	4.103448	-0.28784	0.773883	-9.29241	6.930103
X_4	2.382363	2.891732	0.823853	0.411395	-3.3337	8.098428
X_5	4.736788	2.019845	2.345124	0.020396	0.744176	8.7294
X_6	-0.656	1.542828	-0.4252	0.671333	-3.7057	2.393691

From Table 9, all of the regression coefficients except X_5 are not significant since their p values > 0.05 at 0.05 level of significance and the confidence intervals are large. This again, is as a result of the assumption of multicollinearity not being satisfied.

3.2.3 Artificial Neural Network for the Simulated Data

Table 10 below gives us the summary of the result.

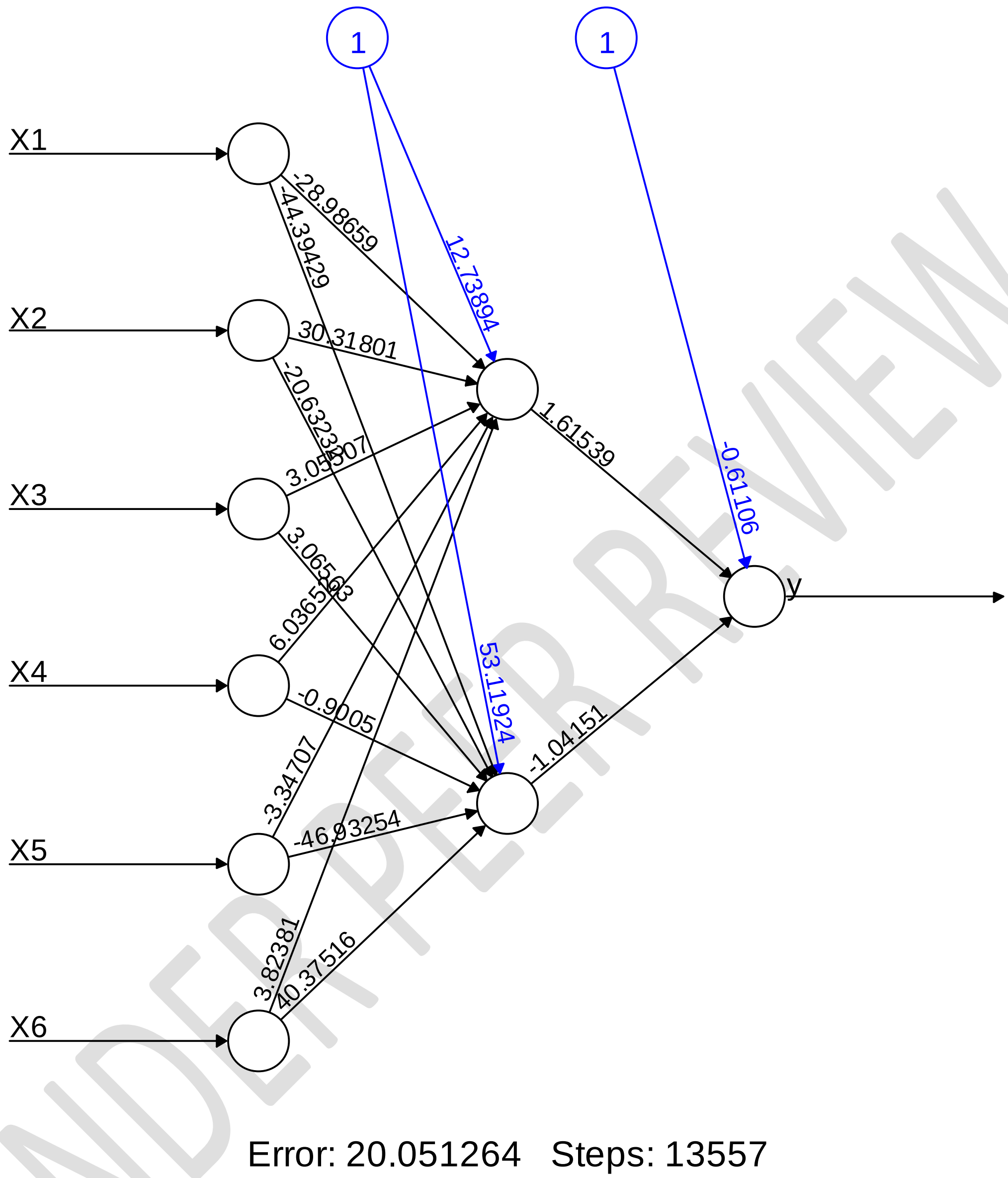
Table 10: RMSE statistics for the Simulated Data

	OLSR	ANN1	ANN2	ANN3	ANN4	ANN5	ANN6	ANN7	ANN8	ANN9	ANN10
RMSE (testing)	13.5	13.9	13.1	31.5	16.5	18.9	20.5	45.0	19.1	20.3	31.8
RMSE (training)	13.1	12.9	11.4	9.5	9.9	8.6	8.2	7.8	6.3	4.1	3.6

The ANN model with two hidden nodes gave us the least RMSE when compared to the OLSR model and the other ANN modelled with one, three, four, five, six, seven, eight, nine and ten hidden nodes in the testing set. The network with two hidden nodes modelled the data very well. It did not over fit the training set and also predicted well the testing set. Below is the estimate of the parameters of the ANN model with two hidden nodes and the graphical representation of the model.

```
Intercept.to.1layhid1  1.273894e+01
X1.to.1layhid1        -2.898659e+01
X2.to.1layhid1         3.031801e+01
X3.to.1layhid1         3.055071e+00
X4.to.1layhid1         6.036519e+00
x5.to.1layhid1        -3.347070e+00
X6.to.1layhid1         3.823806e+00
Intercept.to.1layhid2  5.311924e+01
X1.to.1layhid2        -4.439429e+01
X2.to.1layhid2        -2.063232e+01
X3.to.1layhid2         3.065635e+00
X4.to.1layhid2        -9.005025e-01
x5.to.1layhid2        -4.693254e+01
X6.to.1layhid2         4.037516e+01
Intercept.to.y        -6.110575e-01
1layhid1.to.y          1.615394e+00
1layhid2.to.y         -1.041508e+00
```

Figure 3. Single Hidden Layer Feed Forward Neural Network for the Simulated Data.



4.0 CONCLUSION

Correlation coefficient was used to test for multicollinearity in the two data set and both the real life and simulated data failed to satisfy the multicollinearity assumption. The ANN models had a lesser RMSE than the OLSR model for all the different models except the models with nine and ten nodes in the hidden layer for the real life data, the ANN models with nine and ten hidden nodes over fitted the training set. The network with four hidden nodes had the least RMSE when used on the testing set. It did not over fit the training set and also predicted well the testing set.

For the simulated data, the ANN model with two hidden nodes gave us the least RMSE when compared to the OLSR model and the other ANN models with one, three, four, five, six, seven, eight, nine and ten hidden nodes in the testing set. The network with two hidden nodes modelled the data very well. It did not over fit the training set and also predicted well the testing set.

When there is multicollinearity, it is advisable to use the ANN to model the data since unlike the OLSR method, it has no assumption that must be satisfied and it achieves a better fit and forecast than the OLSR in the presence of multicollinearity as seen from this study using a real life and a simulated data.

REFERENCES

Asep S. Agung P. R. and Asep K. S. (2015). The Application of ANN-Linear Perceptron in the Development of DSS for a Fishery Industry. *Procedia Computer Science,* 72, 67-77.

Aysenur G., Elif T., Derya U., Sibel Y. and Nurgul A. (2019). Prediction of the Colorimetric Parameters and Mass Loss of Heat-Treated Bamboo: Comparison of Multiple Linear Regression and Artificial Neural Network Method. *Colour Research and Application,* 44, 1-10.

Ebiendele E. P. and Ebiendele E. P (2018). Skill Comparison of Multiple-Linear Regression Model and Artificial Neural Network Model in Seasonal Rainfall Prediction-north East Nigeria. *Asian Research Journal of Mathematics,* 11(2), 1-10.

Hsiao-Tien Pao (2008). A Comparison of Neural Network and Multiple Regression Analysis in Modelling Capital Structure. *Expert Systems with Applications,* 35, 720-727.

Ilaboya I. R. and Igbinedion O. E. (2019). Performance of Multiple Linear Regression (MLR) and Artificial Neural Network (ANN) for the Prediction of Monthly Maximum Rainfall in Benin City, Nigeria. *International Journal of Engineering Science and Application,* 3(1), 21-37.

Kumar, U. A. (2005). Comparison of Neural Networks and Regression Analysis: A New Insight. *Expert Systems with Applications*, 29(2), 424–430.

Olewuezi, N.P., Onoghojobi, B., and Bartholomew, D.C. (2016). "Estimation of Nonorthogornal Problem Using Time Series Dataset". *Journal of the Nigerian Association of Mathematical Physics*, Vol. 34, (March, 2016), pp: 141 – 150.

Onoghojobi, B., Olewuezi, N. P. and Obite, C. P. (2016). "Comparison of Linear Prediction Methods in Time Series". *Journal of the Nigerian Association of Mathematical Physics*, vol. (34), pp 151-156.

Ramírez, M. C. V., de Campos Velho, H. F., and Ferreira, N. J. (2005). Artificial neural network technique for rainfall forecasting applied to the São Paulo region, *J. Hydrol.*, 301, 146–162.

Soukayna M. and Jan F. A. (2015). Prediction of SPEI using MLR and ANN: A case study for Wilsons Promontory Station in Victoria. *International Conference on Machine Learning and Application*, 14, 318-324.

Wonsuk Y., Robert M., Sejong B., Karan S., Qinghua P. and James W. L (2014). "A Study of Effects of Multicollinearity in the Multivariable Analysis. *International Journal of Applied Science and Technology*, 4(5), 9-19.

Yazid, M. A. and Mowafaq, M. A. (2009). "A Monte Carlo comparison between ridge and principal components regression methods", *Applied Mathematical Sciences*, Vol. 3, Vol. 42, pp. 2085 – 2098.

Zou J., Han Y., and So S.S (2008). Overview of Artificial Neural Networks, *Methods Mol Biol*, 458, pp. 15-23.