# A COMPARATIVE STUDY FOR BACK PROPAGATION NEURAL NETWORK AND NONLINEAR REGRESSION MODELS FOR PREDICTING DENGUE OUTBREAK

Nor Azura Husin[1], Naomie Salim[2]

[1]Faculty of Computer Science and Information Technology
University of Putra Malaysia
43400 Serdang, Selangor

[2]Faculty of Computer Science and Information Systems
University Teknologi Malaysia
81300 Skudai, Johor

Email: [1]nazura1112@gmail.com, [2]naomie@fsksm.utm.my

**Abstract**: Malaysia has a good dengue surveillance system but there have been insufficient findings on suitable model to predict future dengue outbreak since conventional method is still being used. This study aims to design a Neural Network Model (NNM) and Nonlinear Regression Model (NLRM) using different architectures and parameters incorporating time series, location and rainfall data to define the best architecture for early prediction of dengue outbreak. Four architecture of NNM and NLRM were developed in this study. Architecture I involved only dengue cases data, Architecture II involved combination of dengue cases data and rainfall data, Architecture III involved proximity location dengue cases data, while Architecture IV involved the combination of all criteria. The parameters studied in this research were adjusted for optimal performance. These parameters are the learning rate, momentum rate and number of neurons in the hidden layer. The performance of overall architecture was analyzed and the result shows that the MSE for all architectures by using NNM is better compared by NLRM. Furthermore, the results also indicate that architecture IV performs significantly better than other architectures in predicting dengue outbreak using NNM compared to NLRM. It is therefore proposed as a useful approach in the problem of time series prediction of dengue outbreak. These results can help the government especially the Vector Borne Disease Control (VBDC) Section of Health Ministry to develop a contingency plan to mobilize expertise, vaccines and other supplies that may be necessary in order to face dengue epidemic issues.

**Keywords**: Dengue outbreak prediction, Neural Network Model (NNM), Nonlinear Regression Model (NLRM).

## 1. INTRODUCTION

Recently, predictions on dengue outbreak become very important. With prediction, government and health departments may provide plans and arrange early intervention programs including campaigns to those susceptible groups of communities before an outbreak occurs. In Malaysia, despite having a good laboratory based surveillance system, with both serology and virology capability, it is basically a passive system and has little predictive capability [1].

Many researches that had been conducted before had proven a strong relation between past dengue cases, climate factor and environment factors to the spread of dengue outbreak ([2] and [3]). Various factors such as dengue fever prevalence, population distribution and meteorological factor like rainfall are important in determining the mosquito survival and reproduction [2]. The increased incidence may first occur in regions bordering endemic zones in latitude or altitude. Endemic locations may be at higher risk from hemorrhagic dengue if transmission intensity increases [3].

Many experiments had compared the results of using both Neural Network and Regression for modeling and predicting system. Previous studies on prediction proved that both of these models have the advantages and disadvantages. Most papers concluded that the used of neural network model produced data predictions more accurate or at least comparable to regression.

In term of application, regression model is mostly implied to predict disease outbreak over the other prediction models ([4] and [5]). Nevertheless, [6] found that NNM has biggest potential within general purpose control and able to model a wide class of option in many applications. Furthermore, there are some theoretical advantages comparing a predictive neural network model over regression model. One such advantage is that NNM allows the inclusion of a large number of variables. Another advantage of the NNM approach is that there are not many assumptions that need to be verified before the models can be constructed [7].

## 2. NEURAL NETWORK MODEL

Prediction process using NNM can be dividing into three steps, building the neural network structure, learning processes and testing process.

### i. BUILDING THE NNM

The steps involved to build neural network are as the following: -

1- Define the form of training and testing

2- Define the architecture

a. Number of hidden layer

As far as the number of hidden layers is concerned, there is no theory yet to tell how many hidden layers are needed ([8] and [9]). One hidden layer may be enough for most forecasting problems [10]. This observation is supported in [11] that a network never needs more than two hidden layers to solve most problems including forecasting. The number of hidden layers used in the network is set to one, as it is said to give faster speed in convergence [12].

b. Number of neuron in each hidden layer

Networks with the number of hidden nodes being equal to the number of input nodes are reported to have better forecasting results in several studies [10]. According to [13], the number of hidden nodes can be determined by the formula *2n+1*, where *n* is the number of input nodes. Techniques applied in determining the numbers of hidden node for the networks is trial and error.

Based on the previous researches, equations 1 and 2 are applied in determining the number of hidden node for the network in the study. The equation for hidden nodes used in this research, as follows:

$$\textit{hidden nodes} = 2n + 1 \qquad (1)$$

$$\textit{hidden nodes} = (n/2) \times 2 \qquad (2)$$

or

$$\textit{hidden nodes} = n \qquad (3)$$

where *n* is number of input nodes.

Researcher will try various choices of hidden nodes based on that equation to see which choice leads to minimum prediction error. These combinations produce hidden nodes 4 and 9 for Architecture I, hidden nodes 8 and 19 for Architecture II and Architecture III, and hidden nodes 12 and 25 for Architectures IV. Table 1 shows the numbers of nodes and hidden layer used in every layer of the network in this study.

Table 1: Number of nodes and hidden layer

| | **Architectures** | | | |
|---|---|---|---|---|
| **Nodes** | **I** | **II** | **III** | **IV** |
| **Hidden nodes** | 4, 9 | 8, 17 | 8, 17 | 12, 25 |
| **Hidden layer** | 1 | 1 | 1 | 1 |

c. Number of input node

In this study, the number of input node is determined by the number of attributes that exists in the data. Input parameters are dengue cases data (d, d-1, d-2 and d-3), rainfall data (r, r-1, r-2 and r-3) and dengue cases in approximate location (n, n-1, n-2 and n-3). Table 2 shows the number of input in the each network architecture. Architecture I consists 4 input that dengue cases only, Architecture II consists 8 input which combination of 4 dengue cases and 4 rainfall data, Architecture III consists 8 input which combination of 4 dengue cases and 4 dengue cases data at approximate location, and Architecture IV consists 12 input which combine 4 dengue cases data, rainfall data and 4 dengue cases data at approximate location. Refer figure 1, 2, 3 and 4.

Table 2. Number of input in the input network layer

| | **Architectures** | | | |
|---|---|---|---|---|
| **Nodes** | **I** | **II** | **III** | **IV** |
| **Input** | 4 | 8 | 8 | 12 |

d. Number of output node

The number of output nodes is determined based on the number of target output set. The output parameter is the corresponding dengue cases for the location and time stated. Once trained, the network will be able to estimate the dengue cases using the mentioned parameter. As we can see, ***d*** is represent dengue cases for week *i*, *d-1* represent dengue cases for week *i+1*, *d-2* represent dengue cases for week *i+2* and *d-3* represent dengue cases for week *i+3*. ***r*** is represent rainfall data for week *i*, *r-1* represent rainfall data for *i+1*, *r-2* represent rainfall data for week *i+2* and *r-3* represent rainfall data for week *i+3*. Meanwhile, ***n*** is represent approximate location dengue cases for week *i*, n-*1* represent approximate location dengue cases for *i+1*,*n-2* represent approximate location dengue cases for week *i+2* and n-*3* represent approximate location dengue cases for week $i+3$. In output layer, $O$ is represent predicted output of dengue cases for week $i+4$, $O1$ represent predicted output of dengue cases for week $i+5$, $O2$ represent predicted output of dengue cases for week $i+6$ and $O3$ represent predicted output of dengue cases for week $i+7$. The architecture of NN used in this project is shown in Figure 1, 2, 3 and 4.

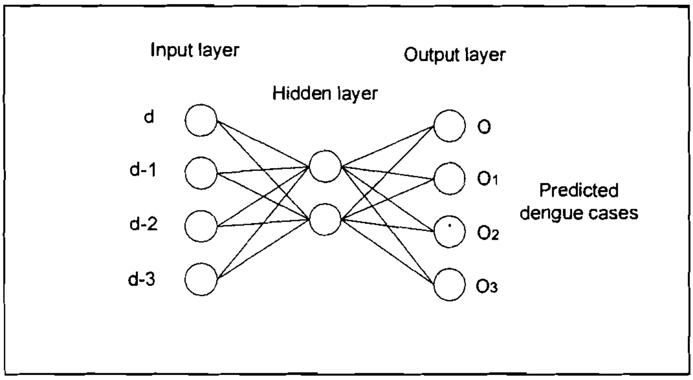


Figure 1. Prediction Architecture I

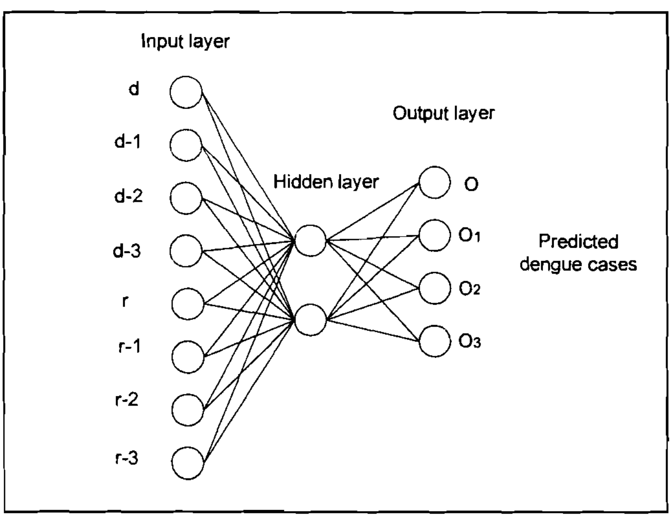


Figure 2. Prediction Architecture II

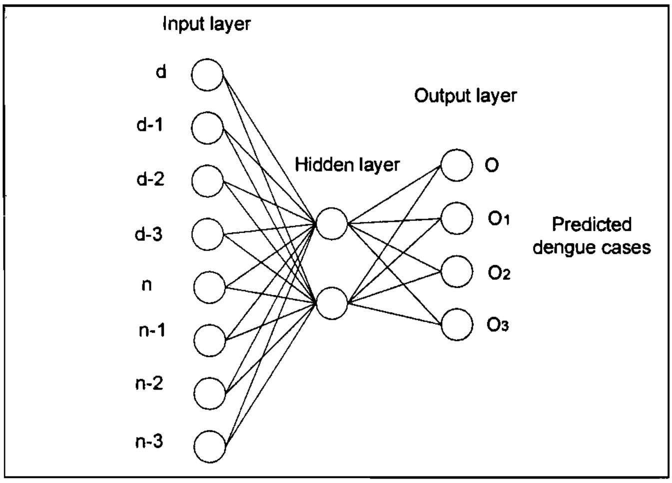


Figure 3. The Prediction of Architecture III

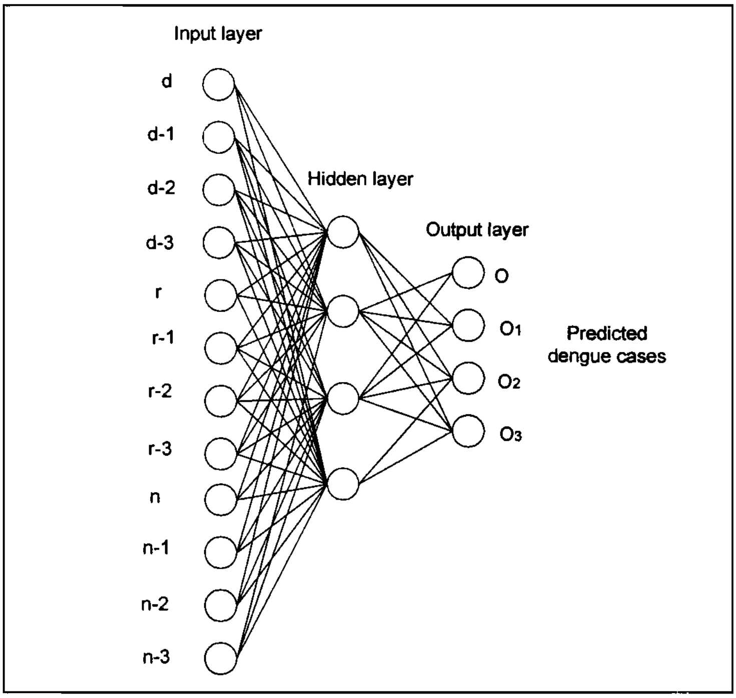


Figure 4. Prediction Architecture IV

3- Select the learning rate and momentum rate

The standard back propagation technique with momentum is adopted by most researchers. Since there are few systematic ways of selecting the learning rate and momentum simultaneously, the best values of these learning parameters are usually chosen through experimentation. As the learning rate and the momentum can take on any value between 0 and 1, it is actually impossible to do an exhaustive search to find the best combinations of these training parameters. Only selected values are considered by the researchers [10]. Learning and momentum rate need to be supplied by user from outside. Researcher will try various choices of learning and momentum rate to see which choice leads to minimum prediction error. Nine combinations of three learning rates (0.5, 0.7 and 0.9) and three momentum rates (0.5, 0.7 and 0.9) used. Table 3 shows the learning and momentum rate used in this research.

Table 3. Learning rate and momentum rate.

| | |
|---|---|
| **Learning Rate** | 0.5, 0.7, 0.9 |
| **Momentum Rate** | 0.5, 0.7, 0.9 |

4- Initialize the network with random weight

Recently, statistical pattern recognition techniques have been used to initialize weight. It can be noted that successful non-random weights initialization can help reduce the convergence time and diminish generalization error ([14] and [15]).However, initial values for weights and bias can be randomly generated so that it would not affect the output to be generated. This is because the values of weights and bias keep on changing during the learning process.

## ii. Training/ Learning Process

The process of learning and training involves forward propagation and backward propagation. Firstly, initial values of weights ($\theta ji$, $\theta kj$), bias ($wji$, $wkj$) and minimum error ($Emin$) are set to compare it with the gradient error generated during back propagation. The steps involved in the learning process is as follows:

i. The input data is fed into network through forward propagation
ii. Output is calculated and compared to the target output to generate error sequences
iii. Test this error through backward propagation.
iv. Backward propagation is executed on the error values in order to produce gradient error, *Egrad*.
v. An error check is performed whereby the gradient error, *Egrad* is compared to the minimum error, *Emin*.
vi. Repeat the learning step until $Egrad > Emin$.

### iii. TESTING PROCESS / VALIDATING PROCESS

Testing process is carried out to validate the network on new or unknown data sets, called the testing set. This is a process which is performed after training the network. A properly built and trained network which can yield the best performance on the validation samples would be the best accurate model. If the validation sample outputs are not acceptable then a new network is to be built by repeating the whole process of learning and testing.

## 3. NONLINEAR REGRESSION MODEL

For the architecture estimation part, the general-to-specific modeling development was used. In this modelling, it begins with a simpler model and ends up with a more complex extension of the original model. The insignificance independent variable will be removed from the model. Through the process of eliminating non-significant variables, then proceed to selecting the most appropriate architecture. The variable reduction processes are iteratively executed until all non-significant variables are eliminated. The decision to drop the irrelevant variables was based on R-Square and mean square error method.

Architecture I consist of *d* Architecture II consist of *d* and r, Architecture III consist of *d* and n and Architecture IV consists of all possible independent variable where:

$yt$ = target value (dependent variable)

$d$ = dengue cases data (independent variable)

$r$ = rainfall data (independent variable)

$n$ = approximate location dengue cases data (independent variable)

The 'order' of polynomial equation tell us how many terms are in the equation. Higher order models wiggle more than do lower order models. Since the equation rarely corresponds to a scientific model, trial and error are used. This experiment show that equation third order is the most appropriate compared to other order.

By using equation third order of polynomial, the general equation of the Architecture I is given by:

$$y_t = \beta_0 + \beta_1 d + \beta_2 d^2 + \beta_3 d^3 + \varepsilon_t \quad (4)$$

The general equation of the Architecture II is given by:

$$y_t = \beta_0 + \beta_1 d + \beta_2 r + \beta_3 d^2 + \beta_4 r^2 + \beta_5 d^3 + \beta_6 r^3 \quad (5)$$

The general equation of the Architecture III is given by:

$$y_t = \beta_0 + \beta_1 d + \beta_2 n + \beta_3 d^2 + \beta_4 n^2 + \beta_5 d^3 + \beta_6 n^3 \quad (6)$$

The general equation of the Architecture IV is given by:

$$\begin{aligned} y_t = {} & \beta_0 + \beta_1 d + \beta_2 r + \beta_3 n + \beta_4 d^2 + \beta_5 r^2 + \beta_6 n^2 + \\ & \beta_7 d^3 + \beta_8 r^3 + \beta_9 n^3 + \varepsilon_t \end{aligned} \quad (7)$$

where $\beta 0$ = intercept

$\beta 1, .., \beta n$ = slope

The equation for intercept of the regression line,

$$\beta_0 = \bar{y} - b\bar{x}$$

where slope, $\beta n$ is calculated as

$$\beta_n = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

## 4. DATA AND IMPLEMENTATION MODEL

Data on consultations for dengue outbreak prediction was available from State Health Department of Selangor (SHD) for the year 2004 to year 2005 that divided by weeks and each year consists of 52 weeks. These cases are separated by locations and time. Locations are including five districts of Selangor that incorporates Sepang, Hulu Selangor, Hulu Langat, Klang and Kuala Selangor.

Meanwhile, rainfall data is supplied by the Malaysian Meteorological Service that includes daily rainfall data from year 2004 until 2005. The daily rainfall amount collected over the 24-hour period beginning from 8.00 a.m. on that day. The daily data will be averaged into weekly data.

Therefore, four Architectures; Architecture I involved only dengue cases data, Architecture II involved combination of dengue cases data and rainfall data, Architecture III involved dengue cases data in proximity location and Architecture IV involved the combination of all criterion were developed in this study by using NN and RM.

## 4.1. IMPLEMENTATION OF NEURAL NETWORK MODEL

The NNM used was a three layer (one hidden layer, an input and an output layer) back propagation model. Four architectures (Architecture I, II, III and IV) and varied parameters are built as shown in Table 4 and Table 5.

The numbers of neurons in the hidden layers for each of models were varied and the corresponding results examined. Networks with the number of hidden nodes being equal to the number of input nodes are reported to have better forecasting results in several studies [8]. The number of hidden nodes can be determined by the formula $2n+1$, where $n$ is the number of input nodes [9].

Through the literature, the number of hidden layers must be adjusted until it shows the best result. However, one hidden layer would give the best result, as the uses of more than one hidden layer will result in the addition of parameter number [10].

Table 4. Parameter Used of the Neural Network for All Architectures

| Nodes | Architectures | | | |
|---|---|---|---|---|
| | I | II | III | IV |
| **Input** | 4 | 8 | 8 | 12 |
| **Hidden** | 4,9 | 8,17 | 8,17 | 12,25 |
| **Output** | 4 | 4 | 4 | 4 |

Table 5. Learning rate and momentum rate.

| | |
|---|---|
| **Learning Rate** | 0.5, 0.7, 0.9 |
| **Momentum Rate** | 0.5, 0.7, 0.9 |

The next attempt was to monitor the effect of momentum and learning rate on the models. As the learning rate and the momentum can take on any value between 0 and 1, it is actually impossible to do an exhaustive search to find the best combinations of these training parameters [8]. Only selected values are considered by the researchers. The models were thus tested by varying their values. The momentum and learning rate were varied from 0.5, 0.7 and 0.9. The final experiment was experiment the use of Gaussian function with the standard sigmoid (logistic) function as the threshold activation function in the neurons. The models were run with each function and their corresponding results recorded.

## 4.2. IMPLEMENTATION OF NONLINEAR REGRESSION MODEL

Architecture IV in Klang had the high $R^2$ value of regression coefficients with 0.592, which indicated high association of the regression coefficients with variances in the predictor values. All these evidences showed a strong relationship between the predictor variables (dengue cases data, rainfall data and proximity location of dengue cases data) and the predicted variable for Architecture IV compared with other architecture.

The results of analysis of variance (ANOVA) of the architectures also supported strong relationships in the architecture (Table 6). The F value of regression were 20.4478 and these high F value indicated a great significance ($\alpha$ = 0.000) for architecture in rejecting the null hypothesis (H0) that every coefficient of the predictor variables in the architecture was zero and the mean square error (MSE) is 26.054.

The coefficients of all predictor variables and the intercept of the architecture are listed in Table 7. According to these coefficients, the nonlinear regression models are built as in equation 8 by using equation third order polynomial.

Table 6. The ANOVA Table of the Nonlinear Regression for Architecture IV

| Item | Value |
|---|---|
| **SSE** | 2084.298 |
| **MSE** | 26.054 |
| **RMSE** | 5.104 |
| **Significant value ($\alpha$)** | 0.00 |
| **R-square** | 0.592 |

$$\begin{aligned} y_t = {} & \beta_0 + \beta_1 d + \beta_2 r + \beta_3 n + \\ & \beta_4 d^2 + \beta_5 r^2 + \beta_6 n^2 + \\ & \beta_7 d^3 + \beta_8 r^3 + \beta_9 n^3 + \varepsilon_t \end{aligned} \tag{8}$$

where

$yt$ = target value (dependent variable)

$d$ = dengue cases data (independent variable)

$r$ = rainfall data (independent variable)

$n$ = approximate location dengue cases data (independent variable)

$\beta 0$ = intercept

$\beta 1$,.., $\beta n$ = slope

Table 7. Coefficient of the Architecture IV

| Predictor Variable | Coefficients | Standard error |
|---|---|---|
| β0 (intercept) | 6.662 | 19.702 |
| β1 | 0.101 | 0.452 |
| β2 | 0.192 | 0.700 |
| β3 | -0.240 | 0.244 |
| β4 | 0.036 | 0.033 |
| β5 | -0.028 | 0.095 |
| β6 | 0.004 | 0.005 |
| β7 | -0.001 | 0.001 |
| β8 | 0.001 | 0.003 |
| β9 | 0.000 | 0.000 |

## 5. COMPARISON RESULT OF NEURAL NETWORK AND NONLINEAR REGRESSION MODEL

Based on the experiment, the comparison of prediction performance in Klang is done and this give the best result compared to other location. Figure 5 and Figure 6 show the comparison of target output and predicted output for dengue cases using Neural Network Model and Nonlinear Regression Model.

The performance of overall architecture was analyzed and the results showed that the MSE for architectures IV by using NNM better compared to NLRM. The best structure of architecture IV for NNM (Klang) are 12 input, four output, one hidden layer, 25 hidden nodes, 0.9 of learning rate, 0.9 of momentum rate and using sigmoid activation function with MSE 0.0278 and NLRM with MSE 26.054 (Table 8).

It is obviously shows that Neural Network Model produces the better prediction on dengue cases for both locations compared to Nonlinear Regression Model. It is occur because Neural Network Model is like one of the neural network algorithm with global learning features where if more data features are present in network learning therefore the prediction is become better.

Table 8. The best structure of architecture IV (Klang) for NNM and NLRM

| Model | Structure of Architecture | MSE |
|---|---|---|
| NNM | input=12, output=4, hidden node=25, hidden layer=1, learning rate=0.9 and momentum rate=0.7 | 0.0278364 |
| NLRM | $R^2$=0.592 | 26.054 |

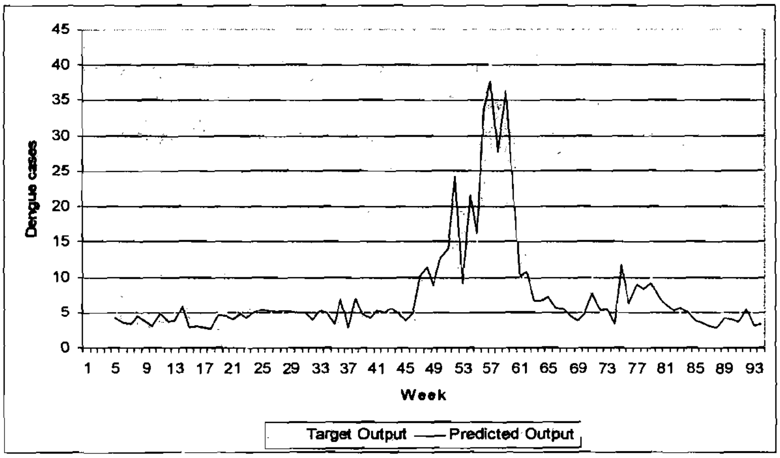


Figure 5. Comparison of target output and predicted output for dengue cases using NNM.

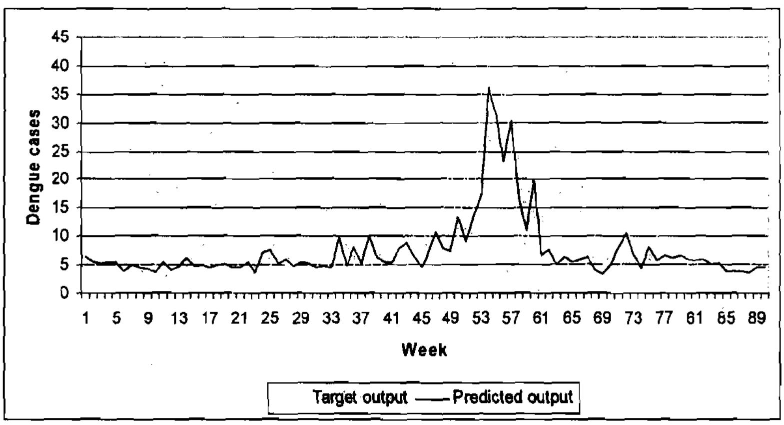


Figure 6. Comparison of target output and predicted output for dengue cases using NLRM.

## 6. DISCUSSION AND CONCLUSION

The findings of this experiment also suggest that the best number of neurons in the hidden layer should be $2n+1$, where $n$ is the number of input. The danger of having too many neurons in the hidden layer is that the model may curve fit the test set thus failing to generalize as it uses the additional neurons to build input-specific relationships. The additional neurons also result in exponential increase in training time.

The results of the learning and momentum rate are related but their relationship is still not clear. The results suggest an adaptive learning and momentum rate of 0.9 and 0.5 respectively being the most appropriate. High learning rate tended to deteriorate in the larger network. The learning network provided a balance result in terms of convergence speed, stability and accuracy. Meanwhile high momentum values speed up the learning process by converging quickly to an optimal result but quickly deteriorate once it reaches the minimum. Low momentum values on the other hand result in very slow convergence especially in the larger networks.

In constructing the prediction regression model, forecaster needs to look at quite a few alternative specifications during the model formulation and later on decide the final model that will fit the historical data well. Some model is said to be miss-specified if it fails to meet some or all of the diagnosis test criteria. The four common situations where miss-specified may occur to prediction regression models is when unimportant variables are being included in the model, the functional form of the model is questionable, related variables are included in the model and issues related to an analysis of the residuals or error associated with any specification regression model are not satisfactorily answered. When using regression model as predictor tool, research must have a deep understanding of statistics to ensure only the necessary independent variables are used.

From this experiment, it shown that neural network model is capable of producing better prediction result compared to the nonlinear regression model. The capability of this model is proven by other studies especially on prediction. However, the results may differ according to the different problems. As a conclusion, the better result of prediction is based on the good architecture model. Therefore, relevant data, pre-processing data, identify of the best parameter, learning algorithm and also the compatible measure method, must be taken into consideration.

## ACKNOWLEDGEMENTS

The author would like to thanks to Dr Nor Aini Bt. Mohd Noor the principal Assistant Director (vector) at State Health Department of Selangor and customer service unit staffs at Malaysian Meteorological Service for their assistance in supplying the relevant data.

## REFERENCES

[1] Gubler, D.J., "How Effectivelly is Epidemiological Surveillance used for Dengue Programme Planning and Epidemic Response?," Dengue Bulletin, Volume 26, Haemorrhagic Fevers, World Health Organization, Geneva, 2002.

[2] Seng, S. B., Chong, A. K. and Moore, A., "Geostatistical Modelling, Analysis and Mapping of Epidemiology of Dengue Fever in Johor State, Malaysia," The 17th Annual Colloquium of the Spatial Information Research Centre University of Otago, Dunedin, New Zealand, 2005.

[3] Patz J.A., Willem J.M.M., Dana A.F, Theo H.J., "Dengue Fever Epidemic Potential as Projected by General Circulation Models of Global Climate Change," Environmental Health Perspective, Vol. 106, No. 3, pp. 147-153, 1998.

[4] Pfeiffer, D.U., Duchateau, L., Kruska, R.L., Ushewokunze-Obatolu, U. and Perry, B.D., "A Spatially Predictive Logistic Regression Model For Occurrence of Theileiriosis Outbreaks in Zimbabwe," Proceeding of 8th Symposium of the International Society for Veterinary Epidemiology and Economics, Paris, France, Special Issues of Epidemiologie et Sente' Animale, 31-32, 12.12.1-3, 1997.

[5] Mooney, J.D., Holmes, E. and Christie, P., "Real Time Modelling of Influenza Outbreak- A Linear Regression Analysis," Euro Surveill, 7:12, pp. 184- 187, 2002.

[6] Norgaard, M., Ravn. O., Poulsen, N.K. and Hansen, L.K., "Neural Network for Modelling and Control of Dynamic Systems: A Practitioner's," Springer-Verlag, London, pp. 4-11, 2000.

[7] Eftekhar, B., Mohammad, K., Ardebilil, H.E., Ghodsi, M. and Ketabchi, E., "Comparison of Artificial Neural Network and Logistic Models for Prediction of Mortality in Head Trauma Based on Initial Clinical Data," BMC Medical Informatics and Making, 5: DOI:10.1186/1472-6947-5-3, 2005.

[8] Toth, E., Brath, A. and Montanari, A., "Comparison of Short-Term Rainfall Prediction Models for Real-Time Flood Forecasting," Journal of Hydrology, 2000.

[9] Hamid, S.A. and Iqbal, Z., "Using Neural Networks For Forcasting Volatility of S&P 500 Index Future Price," Journal of Business Research, 75, pp. 1116-1125, 2004.

[10]Zhang, G., Patuwo, B.E. and Hu, M.Y., "Forecasting with Artificial Neural Networks: The State of the Art," International Journal of Forecasting, 35-62, 1998.

[11]Lippmann, R.P., "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine. 4-32, 1987.

[12]Cabena, P., Hadjinian, P., Stadler, R., Verhees, J. and Zanasi, A., "Discovering Data Mining From Concept to Implementation," Englewood Cliffs: Prentice Hall, 1998.

[13] Subramanian, N., Yajnik, A. and Murthy, R.S.R., "Artificial Neural Network as an Alternative to Multiple Regression Analysis in Optimizing Formulation Parameters of Cytarabine Liposomes," AAPS PharmSciTech, 5(1): pp. 1 – 9, 2003.

[14] Park, Y., "A Mapping from Linear Tree Classifiers," Proceeding of IEEE International Conference of Neural Network, FL, 1, pp. 94-100, 1990.

[15] Seth, I.K., "Entrop. Netts from Decision Tree Neural Networks," Proceeding of IEEE. 78, pp. 105-113, 1990.