CHAPTER IV

ARTIFICIAL INTELLIGENCE FORECASTING MODELS

4.0 Design for Enhancement (ANN)

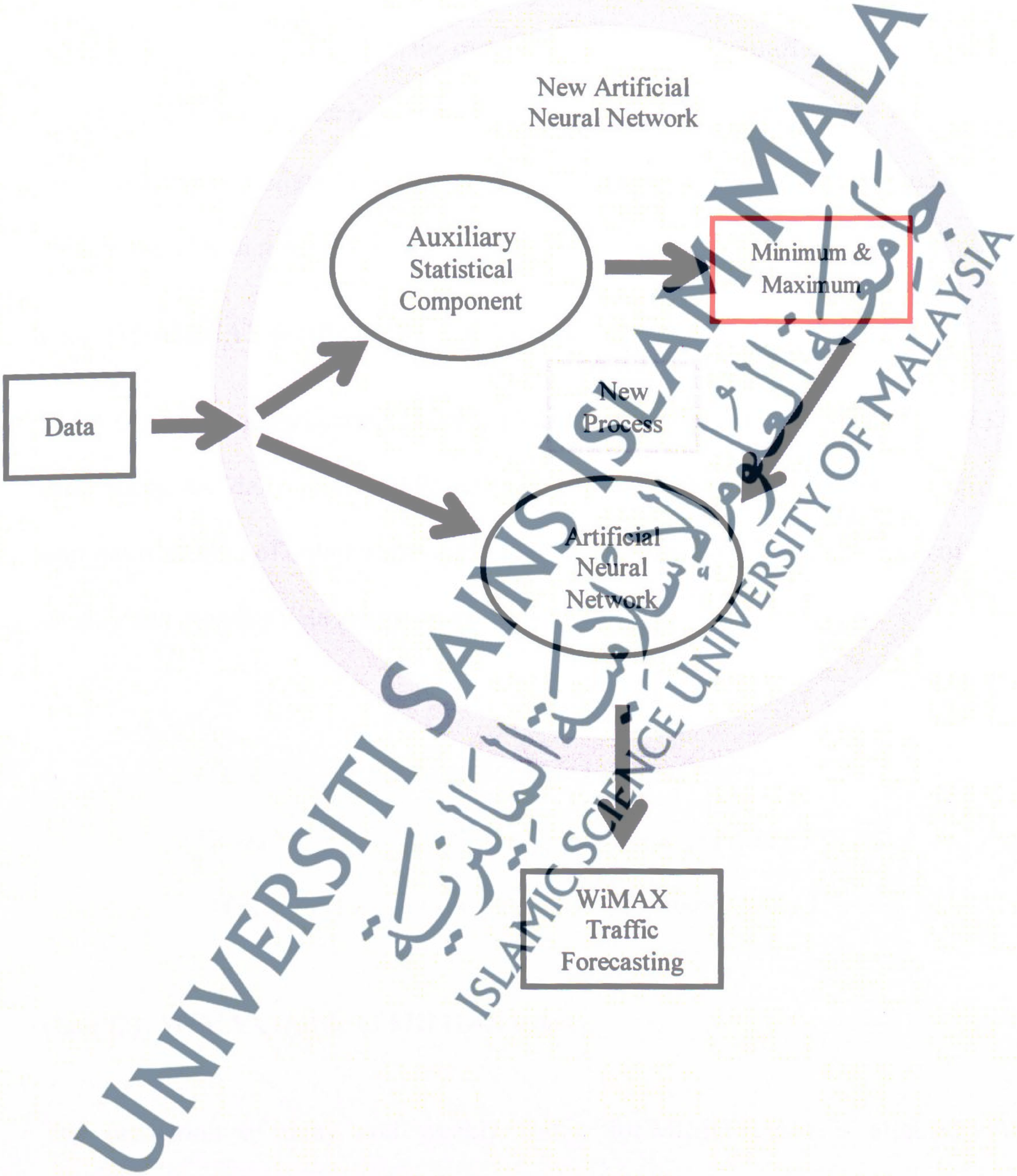
To propose a new enhancement for artificial neural network (ANN), two important criteria must be maintained. First, the improvement should be economical enough that it would not increase the computational complexity of the method at hand. Second, it should be effective enough that the predictive performance of the new approach would be significantly better than the one achieved by its predecessors such as ANFIS. These two criteria are vital to the design of the new method. If one is violated, the outcome may not be as desired.

A known weakness of ANN lies in its inability to predict the minimum and maximum value of network traffic. To combat this, a simple statistical method is integrated with ANN to calculate the minimum and maximum value of the data according to the temporal segregation. As such, the division will depend on how the data is being compartmentalized for processing. It can either be on a daily, weekly or monthly basis. In terms of simplicity, the statistical method fits the criteria. It does not incur a heavy computational cost to ANN.

In order to provide this enhancement (Figure 22), an alteration is needed to the current process of ANN. The data must first be processed by the auxiliary statistical component before being relayed for further examination into ANN. However, the minimum and maximum value can now be integrated as one of the factors that drive the prediction algorithm. ANN is no longer paralyzed by its weakness in forecasting the peak and nadir

of traffic. It can rely on the statistical component for support. In theory, this fulfills the effectiveness criteria.

Figure 22: New Artificial Neural Network (ANN) for WiMAX Traffic Forecasting



4.1 Artificial Neural Network (ANN) model design

To previse the WiMAX traffic using the ANN model, the multilayered feed forward perceptron is exploited. Three sub-models are designed for this purpose whereby each involves network architecture with input and output layers, as well as the hidden layers in between. The data for the input layer consists of the maximum and minimum number of online users. However, data for the output layer involves the traffic of MIMO-A, MIMO-B and MIMO-AB. As such, three configuration patterns or models of training are employed. Each configuration works on two time frames, which are daily and weekly. To explain the idea in more detail, consider the following:

Case (1): WIMAX traffic of MIMO-A users

Here, the effort of estimating the WiMAX Traffic comes from MIMO-A users. This is done using the daily and weekly recorded data, which include the maximum and the minimum number of online users only. Thus, the traffic from MIMO-A users of daily and weekly and monthly data are given as:

$$TDaily(A) = fA(X\,user(Max), X\,user(Min))$$

$$TWeekly(A) = fA(X\,user(Max), X\,user(Min))$$

$$TMonthly(A) = fA(X\,user(Max), X\,user(Min))$$

Case (2): WIMAX traffic of MIMO-B users

The prediction of daily and weekly traffic for MIMO-B user is attained only by considering the maximum and the minimum number of its online users. In effect, so the traffic from MIMO-B users for both time frames, daily and weekly and monthly are:

$$TDaily\ (B) = \square f\,B\ (X\,user\ (Max)\ , X\,user\ (Min))$$

$$TWeekly\ (B)\ \square\square f\,B\ (X\,user\ (Max)\ , X\,user\ (Min))$$

$$TMonthly\ (B) = f\,B\ (X\,user\ (Max)\ , X\,user\ (Min))$$

Case (3): WIMAX traffic of MIMO-AB users

For the final scenario, both aforementioned cases are combined. The prediction of WiMAX traffic for MIMO-AB users from the aspect of daily and weekly and monthly are derived by utilizing the daily and weekly data recorded:

$$TDaily\ (AB) = f\,AB\ (X\,user\ (Max)\ , X\,user\ (Min))$$

$$TWeekly\ (AB) = f\,AB\ (X\,user\ (Max)\ , X\,user\ (Min))$$

$$TMonthly\ (AB) = f\,AB\ (X\,user\ (Max)\ , X\,user\ (Min))$$

Where, *T Daily* (*A* , *B* , *AB*) represents the daily WiMAX traffic from MIMO-A, MIMO-B and MIMO-AB users (byte) respectively while *T Weekly* (*A* , *B* , *AB*) corresponds to the weekly traffic and *T Monthly* (*A* , *B* , *AB*) corresponds to the monthly traffic. (*X*) *User* (*Max*), (X) *user* (*Min*) represents the maximum and the minimum number of online. (Max-online) and (Min-online) respectively, *f A, B, AB* is the function model depend on the architecture of the neural network.

The usage of all the models enables the estimation of traffic in terms of the traffic flow index (output). This provides a prediction on the traffic flow index of the WiMAX network on a daily and weekly basis. Daily and weekly prediction differ from the aspect of overall duration whereby the former is more interested on the short term analysis of the traffic whereby the latter gives a somewhat mid-term portrayal of traffic as a whole.

Working together, both predictions can assist the process of analyzing the network traffic more accurately.

4.1.1 Training Process

To perform the training, the back propagation method is employed whereby neurons are trained and adjusted according to the error, with the synaptic weights appropriately modified. The output of the approach depends on the variability of the input. Essentially, the algorithm is supervised and iterated for multilayer feed forward nets, specifically tailored with nonlinear sigmoidal threshold units, defined by the following equation:

$$f(x) = \frac{1}{(1 + e^{-x})}$$

The characteristic of the problem is reflected in the set of input-output vectors within the modeling stage where 2/3 of the entire data is used as training data. Output is gathered and then compared with the one from the training set. The difference detected between the experimental and desired output will then be utilized to modify the connection weights. Computationally, this is done to minimize error, such that it will reach a stipulated level of tolerance.

At the accepted tolerance level, the connection weights are maintained and used to make decision. The training halts when the mean average error between the measured output and desired output remains unchanged after a number of trials. Consequently, the output attained by this procedure will then be contrasted against the targeted value. This is achieved by measuring the error function, which is defined by the average of square

difference between the output of each neuron in the output layer and the actual desired output.

This process is conducted on both of the datasets – training and testing. For the training, the parameters are set according to the ones below:

Epochs of training	: 10^4
Training goal	: 10^{-3}
Momentum constant	: 0.92
Number of neurons	: [20 10 1]

The maximum time of training, however, is not set at a particular value because it depends on the learning algorithm. Consistency of the model is achieved by normalizing the input and output data within the range of (0, 1). They are returned to the original values after the simulated is completed by employing the following equation:

$$x' = \frac{x_i - x_{Min}}{(x_{max} - x_{Min})}$$

Where, (x) denotes the inputs/outputs of the network and the normalized version is written as (x /). Normalized values are determined by the maximum and minimum value of the input and output. To note, the value of normalized input or output is 1 when the input or output is ^{x}max , and the value of normalized input or output is 0 when the input or output is ^{x}Min .

In designing the ANN, the activation function employs the sigmoid function. On the other hand, the "Tansig" transfer function is used in the hidden layer while the "Purelin" transfer function is deployed in the output layer. The "Tansig" transfer function is:

$$\tan\ sig\ (x) = \frac{2}{1 + e^{(-2x)}} - 1$$

4.1.2 ANN Models Error Analysis

To train, validate and test the feed-forward back-propagation neural network, the data gathered from LibyaMax network is used. Later, it is employed to estimate the WIMAX traffic such that the accuracy can be ascertained in the end. This is crucial in verifying the performance of the prediction model whereby the predicted value generated by the model is compared with the actual data obtained. The stability of the model can then be found by evaluating the difference via the calculation of the statistical error, or more specifically, the mean square error (MSE) that exploits the following equation:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(X_M - X_E)^2$$

Where N number of input-output pairs, XM desired value and XE estimated value

Further statistical analysis will give rise to a parameter known as the model efficiency (*Meff*) for the ANN prediction results. It is defined as:

$$M_{eff} = 1 - \left(\frac{MSE}{Var}\right)$$

Discriminating the best model from the rest can be done by observing the extent of errors that occurred. Statistically speaking, the model having the lowest error (MSE) is deemed as the best one.

The artificial neural network (ANN) uses 80% of the entire data for the training phase (Singh et al., 2014). Training data undergoes normalization (Basu et al., 2010) to make the process more efficient and effectiveness for prediction. This is usually done by finding the mean and standard deviation of the data. The original data is then normalized by finding the difference between the value and mean, divided by the standard deviation.

The second stage involves the selection of the layers (Lopez et al., 2005; Wabwoba & Mwakondo, 2011) and algorithm for the artificial neural network. For this research, the hidden layer consists of a single layer. This implies that it is more flexible than multiple layers (Nakama, 2011) when it comes to learning different cases of data. In effect, the number of neurons for each layer is 20 (input layer), 10 (hidden layer), and 1 (output layer) respectively.

In terms of the model, the multilayer feed forward network (Che et al., 2011) is employed with nonlinear sigmoidal threshold unit. The use of the sigmoid function is favorable given that it is naturally smooth and continuous with a properly bounded range (0, 1). Learning is performed via back-propagation (Hinton, 2007; Baboo & Shereef, 2010; Pratiwi et al., 2011).

$$F(x) = 1 / (1 + e^{-x})$$

Following the aforementioned selection stage is the determination of the parameters within the artificial neural network model. The iteration of the testing involves 10,000 epochs or cycles (10^4). For the sake of quality, the training goal is set at 0.001 or 10^{-3}. Controlling the rate of learning (Zegnini et al., 2009; Damiri & Slamet. 2012) is done by

aligning the momentum of constant at 0.92. This means that any previous change to the weight will be multiplied with this constant and added to the current change.

Epochs of training : 10^4

Training goal : 10^{-3}

Momentum constant : 0.92

Theoretically speaking, more iteration signifies better quality of prediction. In reality however, it may not be so. Therefore, the training will ceased when the average error between the measured output and desired output does not vary after a number of iteration (Jayaram & Abdelhamied, 1995). Finally, the output gathered through the model is contested with the one targeted. This is done via the mean square error (MSE) between the two (Gopalakrishnan, 2010). Lower MSE signifies better forecasting and vice versa. Finally, the output is denormalized

$$X = (X_{normalized} - X_{min}) / (X_{max} - X_{min})$$

Two algorithms TrainLM and TrainSCG (Priyadarshini et al., 2010) are employed for training and testing. TrainLM is a network training function that updates weight and bias values according to the Levenberg-Marquardt optimization (Ibrahimy et al., 2013). On the contrary, TrainSCG achieves the same function via the scaled conjugate gradient method (Chel et al., 2011). Both have been improved for WiMAX traffic prediction by including the maximum and minimum number of users (Figure 23). The entire process for the proposed artificial neural network is given in the illustration (Figure 24).

Figure 23: Improved TrainLM and TrainSCG

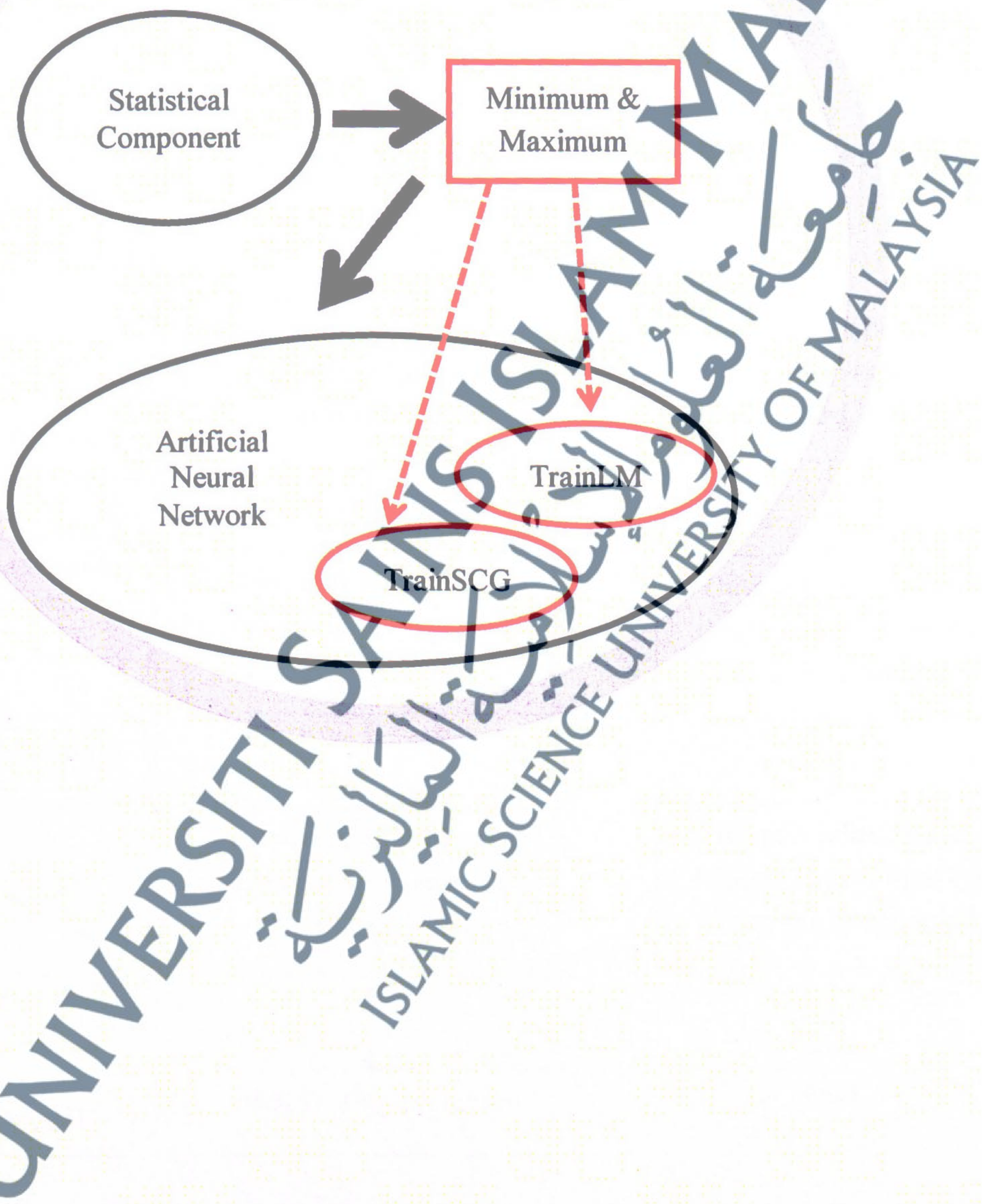
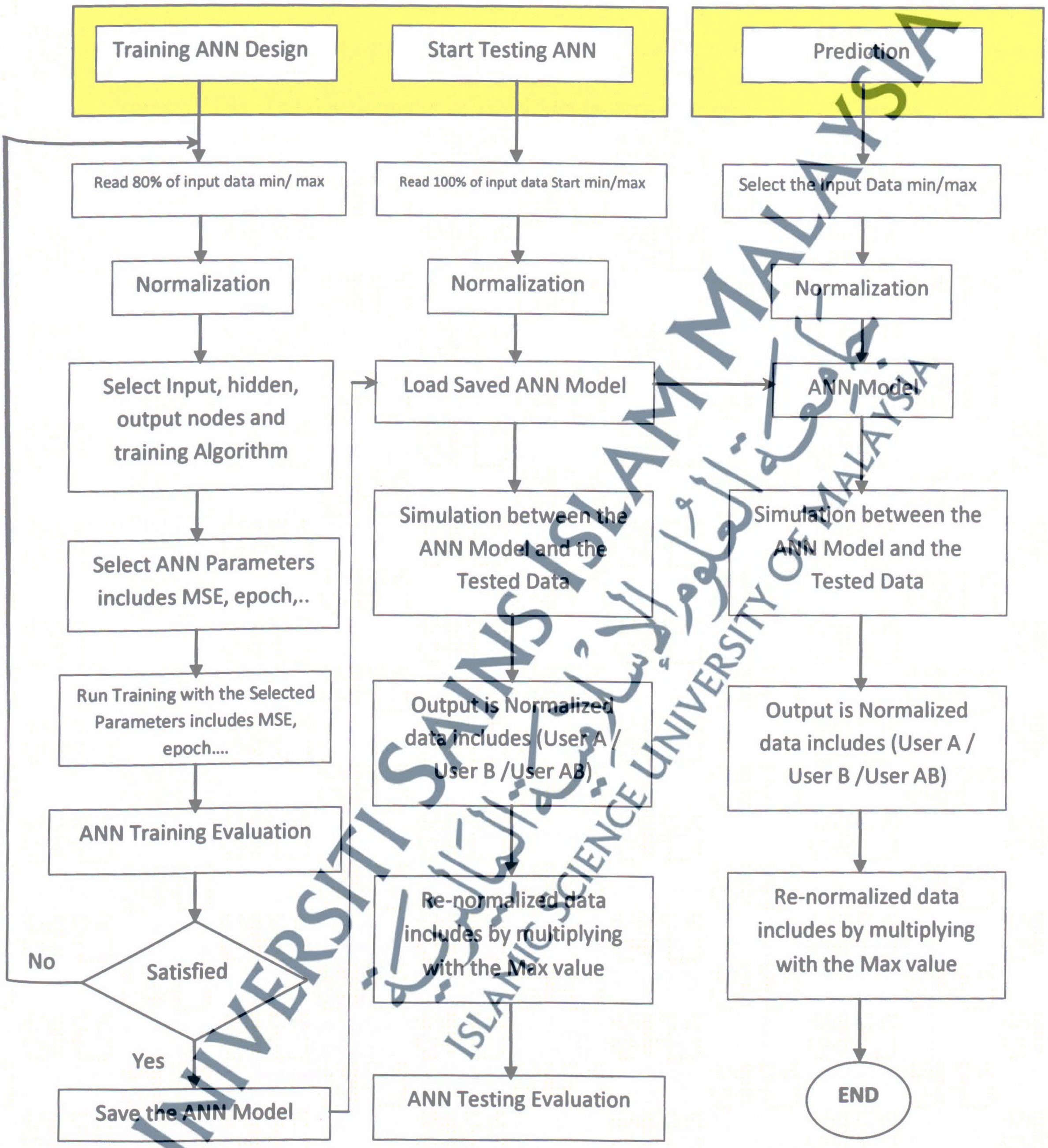


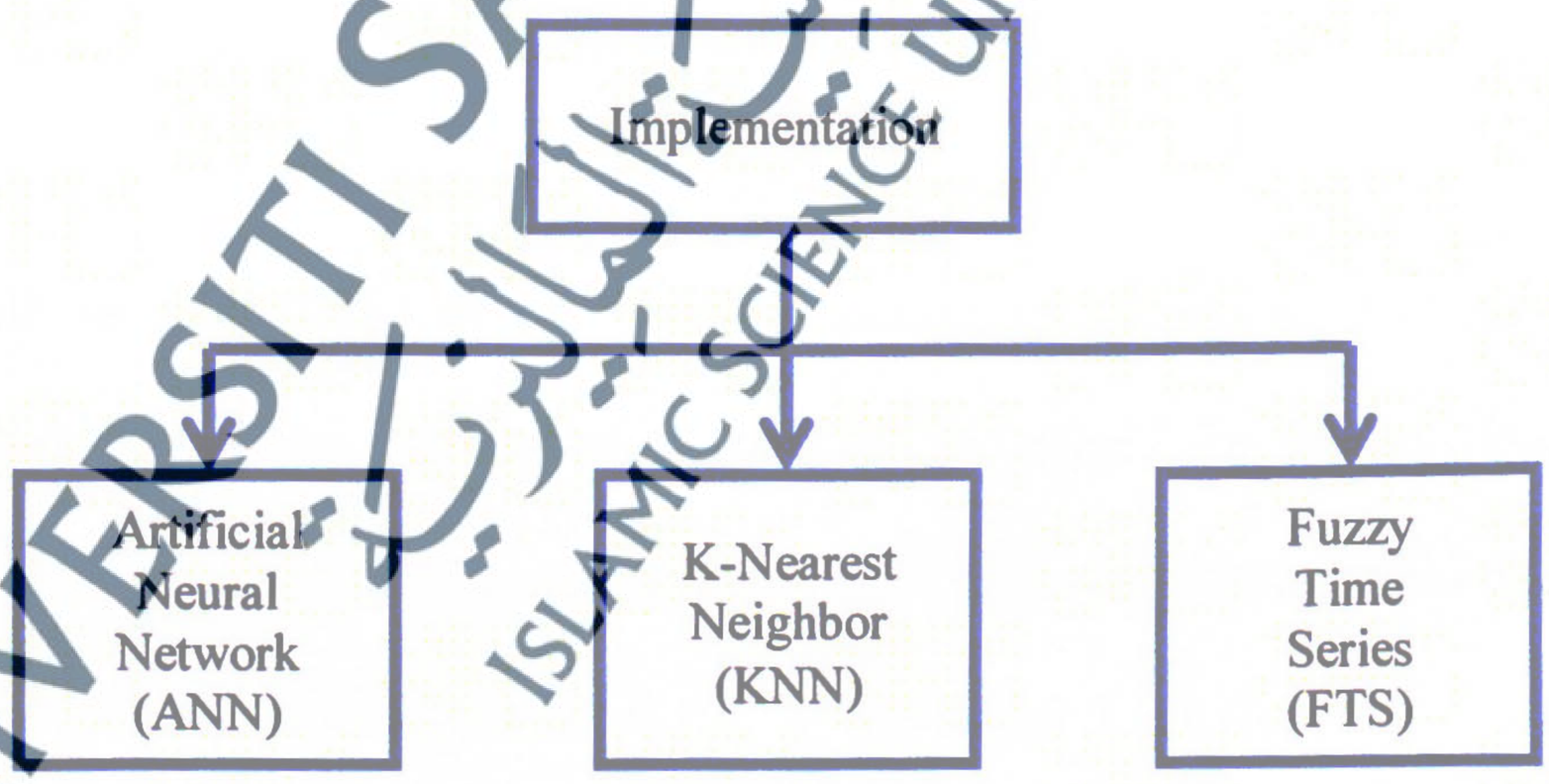
Figure 24: Artificial Neural Network Flowchart



4.2 Implementation of Approaches

Three intelligent forecasting approaches are developed for implementation (Figure 25). They are the artificial neural network (ANN), k nearest neighbor (KNN) and fuzzy time series (FTS). The development of ANN is a materialization of the new design. The other two methods, KNN and FTS are merely implemented without any enhancement. Both are selected from the prospect of their potential in forecasting. As such, they are used as a base of comparison to ANN. Explanation is given on the underlying process of each method. Each step within the process is explained and corroborated by a flowchart that depicts the entire governing procedure as a whole. It must be reminded all the three approaches work with different paradigms. As such, it is not feasible to avoid the extensive diversification.

Figure 25: Implementation Approach



4.3 K-Nearest Neighbor (KNN) model design

To utilize the KNN approach, the value of K must first be determined from the data. Five values of K are chosen. They are one, three, five, seven and nine. Each K is separated by the value of two. Choosing the right K is the initialization stage within the whole process flow of KNN (Lee & Ouarda, 2011). It is imperative in deciding the performance of the algorithm. If the value is defined too large or too small, the predictive performance would deteriorate. For the purpose of this research, the values are manually defined.

Once the value of K is decided, the next step is to calculate the distance of the data of interest as compared to the rest of the data (Lina et al., 2014). The computation stage calculates the distance of a point to other referential points (Amores, Sebe & Radeva, 2006). It measures points (x_1, y_1) and (x_2, y_2) within the two dimensional space by computing the separation that exists within each axis x and y such that the distance $d = [(x_1 - x_2)^2 + (y_1 - y_2)^2]^{1/2}$ would reflect the accurate displacement between them.

The computation stage would result to a series of distance $d_1,.. d_n$ between the point at hand to all the other reference points within the universe of K. These measures of distances will be ranked (Li et al., 2014) accordingly to ensure that the dominant reference points are given priority over the rest. In effect, the ranking stage would eventually arrange the distances in descending order such that for distances $d_1,.. d_n$, then the properly ordered form would be $d_i > d_j > \ldots > d_k > d_m$ for $1 \leq i, m \leq n$.

Having a definite ranking of the neighboring classes would now allow the K-nearest neighbors to be found. Here, the value K will discern the top K neighbors from the ones that will be ignored. For instance, if the K = 1, then only the top neighbor is considered

while the others are dismissed. On the other hand, if K = 5, then the five top neighbors are weighted in the algorithm while the rest are omitted.

The methodology tests a set of different K = {1, 3, 5, 7, and 9} to ascertain the best one for a particular forecasting scenario. Since different scenario would probably demand different value of K, the best K (Li et al., 2014a) is deemed as the representative or dominant K for the case in question. This ensures that the quality is maintained even when the parameters are changed.

Given the K nearest neighbors from the previous stage, classification (Yoon & Friel, 2015) can now be performed at the dominance stage by taking the majority (Cao et al., 2012) from the nearest neighbors for the suggested value. This equates to the prediction value for the algorithm. In this respect, the method performs a prediction solely on the example data that are taken during the earlier stages. It cannot provide a prediction that is beyond that.

KNN is the most usable classification algorithm. This algorithm operation is based on comparing a given new record with training records and finding training records that are similar to it. It searches the space for the k training records that are nearest to the new record as the new record neighbors. In this algorithm nearest is defined in terms of a distance metric such as Euclidean distance. Euclidean distance between two records (or two points in n-dimensional space) is defined by:

If $x_1 = (\chi_{11}, \chi_{12}, \chi_{1n})$ and $x_2 = (\chi_{21}, \chi_{22}, \chi_{2n})$

$$dist(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

Where (χ_1) and (χ_2) are two records with ***n*** attributes. This Formula measures the distance two patterns **(χ_1)** and **(χ_2)** .The K-nearest neighbour classifier is a supervised learning algorithm where the result of a new instance query is classified based on majority of the K-nearest neighbour category. The training samples are described by n-dimensional numeric attributes. Each sample represents a point in an n-dimensional pattern space. In this way, all of the training samples are stored in an n-dimensional pattern space.

4.3.1 KNN Process

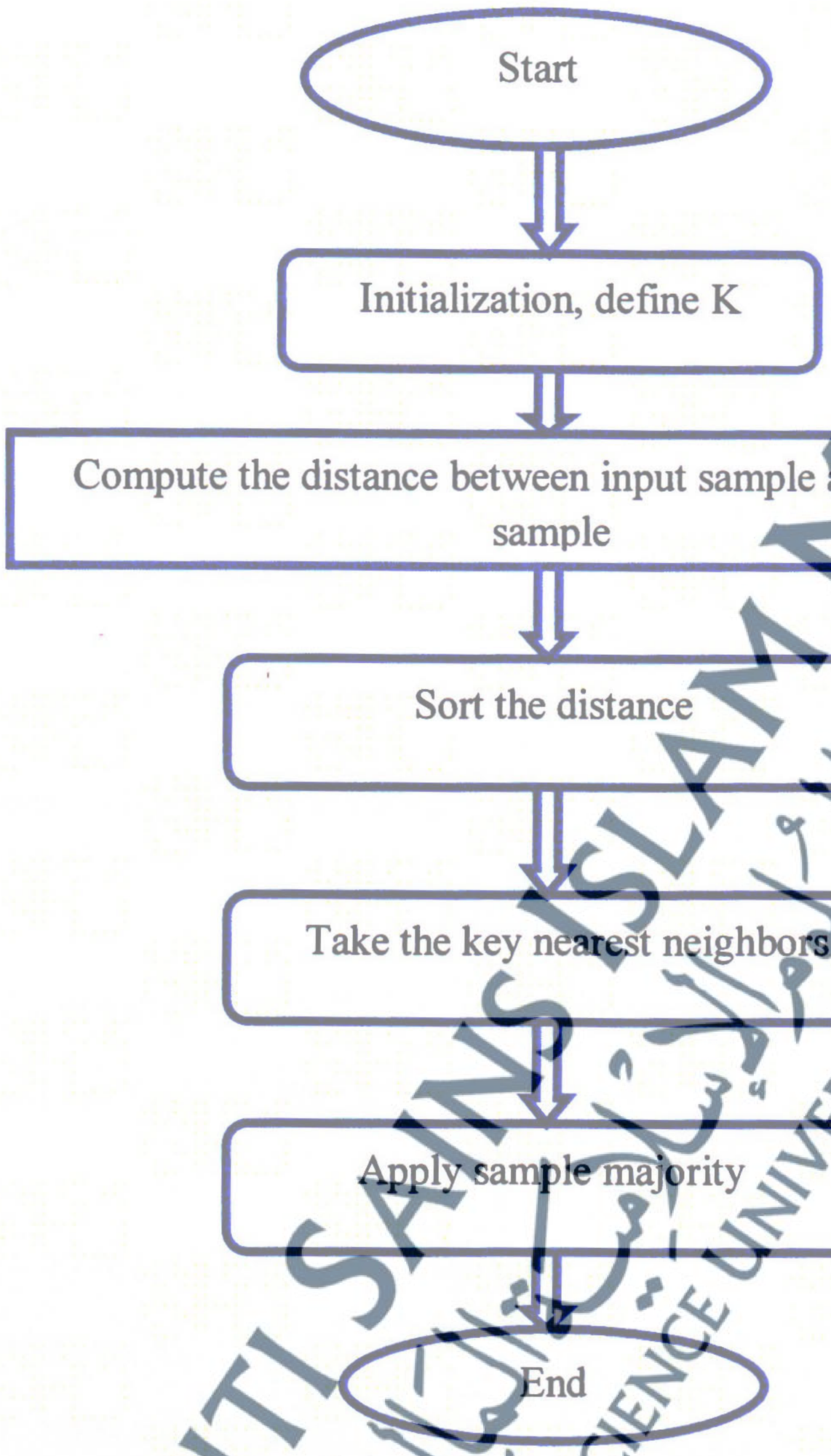
Here is step by step on how to compute KNN algorithm:

1. Determine parameter K = number of nearest neighbours.
2. Calculate the distance between the query-instance and all the training samples.
3. Sort the distance and determine nearest neighbours based on the K^{th} minimum distance.
4. Gather the category Y of the nearest neighbours.
5. Use simple majority of the category of nearest neighbours as the prediction value of the query instance.

The overall process is summarized within the KNN flowchart (Figure 26).

KNN is the simplest approach when compared to artificial neural network and fuzzy time series. However, the simplicity may prove to be highly effective in the end.

Figure 26: K-Nearest Neighbor Flowchart



Gauging the performance of prediction employs two measures, the mean square error (MSE) and the model efficiency (M_{eff}). The mean square error basically calculates the total difference between all the measured value X_M and the expected value XE from the actual data . Then, the average difference is found by dividing with the total number of data N. On the other hand, the model efficiency is calculated by taking the value of one and subtracting it with the ratio between the mean square error (MSE) and the variance (VAR) of the data.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(X_M - X_E)^2$$

$$M_{eff} = 1 - \frac{MSE}{VAR}$$

The general naive algorithm for KNN (Li et al., 2003) is illustrated below. It calculates the distance D(X, Y) between the unknown data Y by comparing it with all the data in the instance $X = X_1 \ldots X_N$. The Euclidean distance is the most common measure of distance. Once the distance between the unknown and all the instances are known, only the k set of data that are nearest would be chosen for classification. For instance, if k = 3, then only three instances with the nearest distance are considered.

CLASS(Y) = MAX {CLASS (K_1) ... CLASS (K_N)}

K = MINIMUM_DISTANCE_SET {D_1 ... D_N}

ENDFOR

D(Y, X) = EUCLIDIAN_DISTANCE(Y, X)

FOR EACH X = {X_1 ... X_N}

NEAREST_NEIGHBOR = K

DISTANCE = D

DATA = X = $X_1 \ldots X_N$

KNN_CLASSIFIER (UNKNOWN, K, DATA){

UNKNOWN = Y}

4.4 Fuzzy Time Series (FTS) model design

For the purpose of this study, the exponential fuzzy time series (Sadaeia et al., 2014; Suhartono, & Lee, 2011) is utilized to predict the WiMAX traffic of the network. The entire process (Figure 27) begins with the definition stage that decides the interval (Abdullah & Ling, 2012; Huarng & Yu, 2006; Huarng, 2001) and the universe of discourse (Ahmed, 2013) for the fuzzy time series. This is succeeded by the establishment of all the observed values in the time series.

The next step is the determination stage. In this stage, the fuzzy logic relationship (Qiu, Liu & Li, 2013) as well as the fuzzy logic relationship group for all the relationships are properly defined. The fuzzy logic relationship denotes the connection that exists between fuzzy sets that lie in a consecutive manner within the time series, which is often gathered from a series of observations. On the other hand, the fuzzy logic relationship group (Chen & Chen, 2014) is critical in giving rise to the forecasting rules.

Once the fuzzy logic relationships are well established, they are ranked accordingly such that the best one can be extracted in the selection stage. The best fuzzy logic relationship is vital in producing the most accurate forecasting. As such, after this stage, the prediction stage can commence. Here, the most optimal fuzzy logic relationship is exploited to derive a systematic forecasting (Wang & Liu, 2010) of the network such as the traffic (Yarushkina, Unusov, & Afanasyeva, 2009) and delay can then be deployed.(Singh, Dutta & Chakrabarti, 2014)

4.4.1 Lee's Method

Lee and Suhartono proposed uniform and exponential chronologically weights to tackle two issues in fuzzy time series forecasting, namely, recurrence and weighting, as extension of Yu's method. This method produces more accurate forecasts than Chen's,

Yu's, and Cheng's methods. The steps of the algorithm of the weighted method proposed by Lee and Suhartono are given as follows.

Step 1. Define the universe of discourse and partition it into intervals as Yu's method.

Step 2. Establish a related fuzzy set (linguistic value) for each observation in the training dataset.

Step 3. Establish fuzzy relationship.

Step 4. Establish fuzzy relationships groups for all FLRs.

Step 5. Select the best order of FLRs. The graphical orders for FLRs and fluctuation type matrixes are used to identify the best order of FLRs.

Step 6. Forecast.

Step 7. Defuzzify. .

Step 8. Assigning weights. Suppose the forecast of $F(t)$ *is* $Aj_1, Aj_2 \ldots Aj_k$.

The Corresponding weights for $Aj_1, Aj_2 \ldots Aj_k$ *Say* $W_1, W_2 \ldots W_k$ are

$$W(t) = \left[\frac{1}{\sum_{h=1}^{k} w_h}, \frac{c}{\sum_{h=1}^{k} w_h}, \frac{c^2}{\sum_{h=1}^{k} w_h}, \ldots, \frac{c^{k-1}}{\sum_{h=1}^{k} w_h}\right]$$

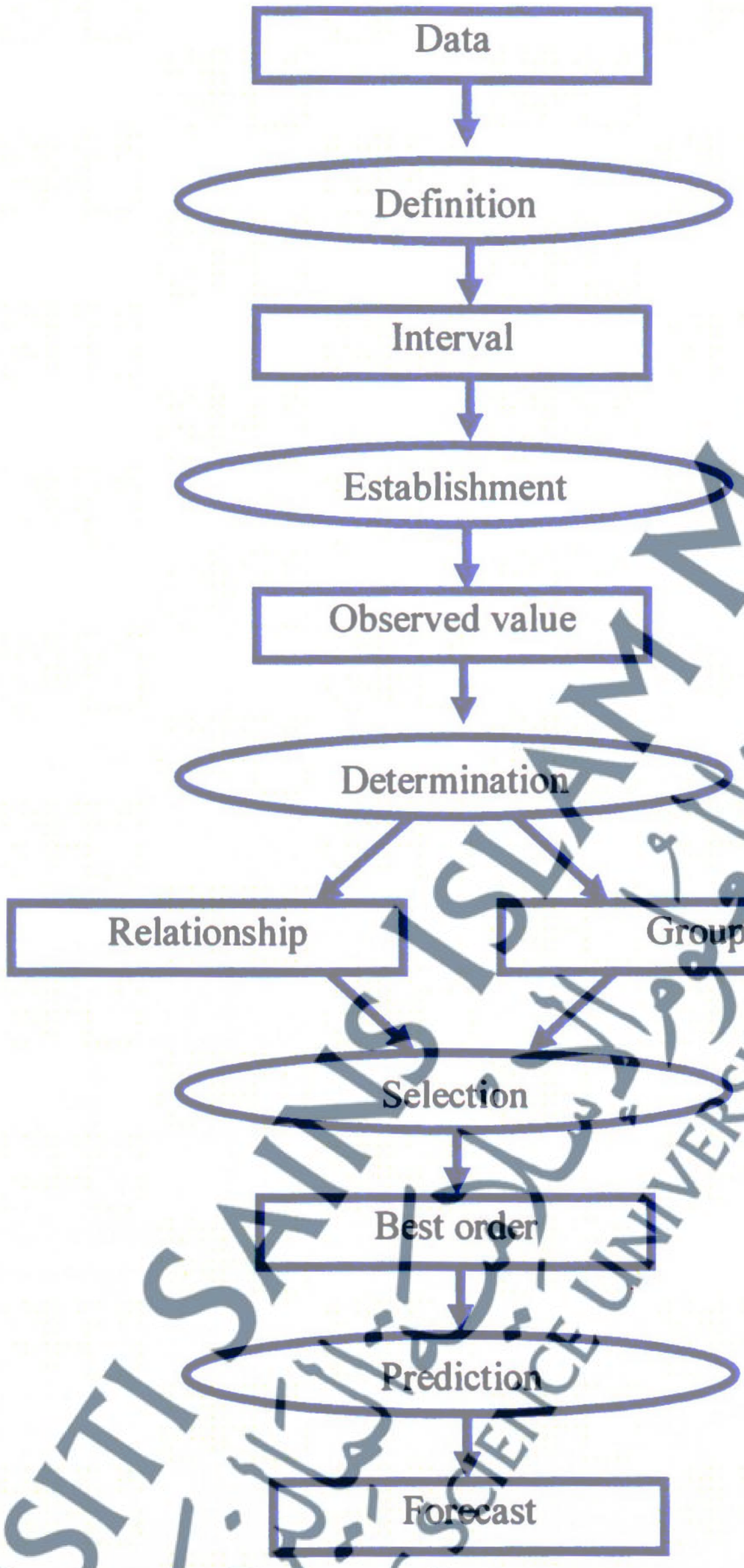
where $w_1 = 1, w_i = c^{i-1}$ for $c \geq 1, 2 \leq i \leq k$, and w_h is the corresponding weight for A_{j_k}.

Step 9. Calculate the final forecast values. The final forecast is equal to the product of the defuzzified matrix and the transpose of the weight matrix:

$$\hat{F}(t) = [m_{j_1}, m_{j_2}, \ldots, m_{j_k}] \times \left[\frac{1}{\sum_{h=1}^{k} w_h}, \frac{c}{\sum_{h=1}^{k} w_h}, \ldots, \frac{c^{k-1}}{\sum_{h=1}^{k} w_h}\right]^T$$

Where x is the matrix product operator.

Figure 27: Fuzzy Time Series Flowchart

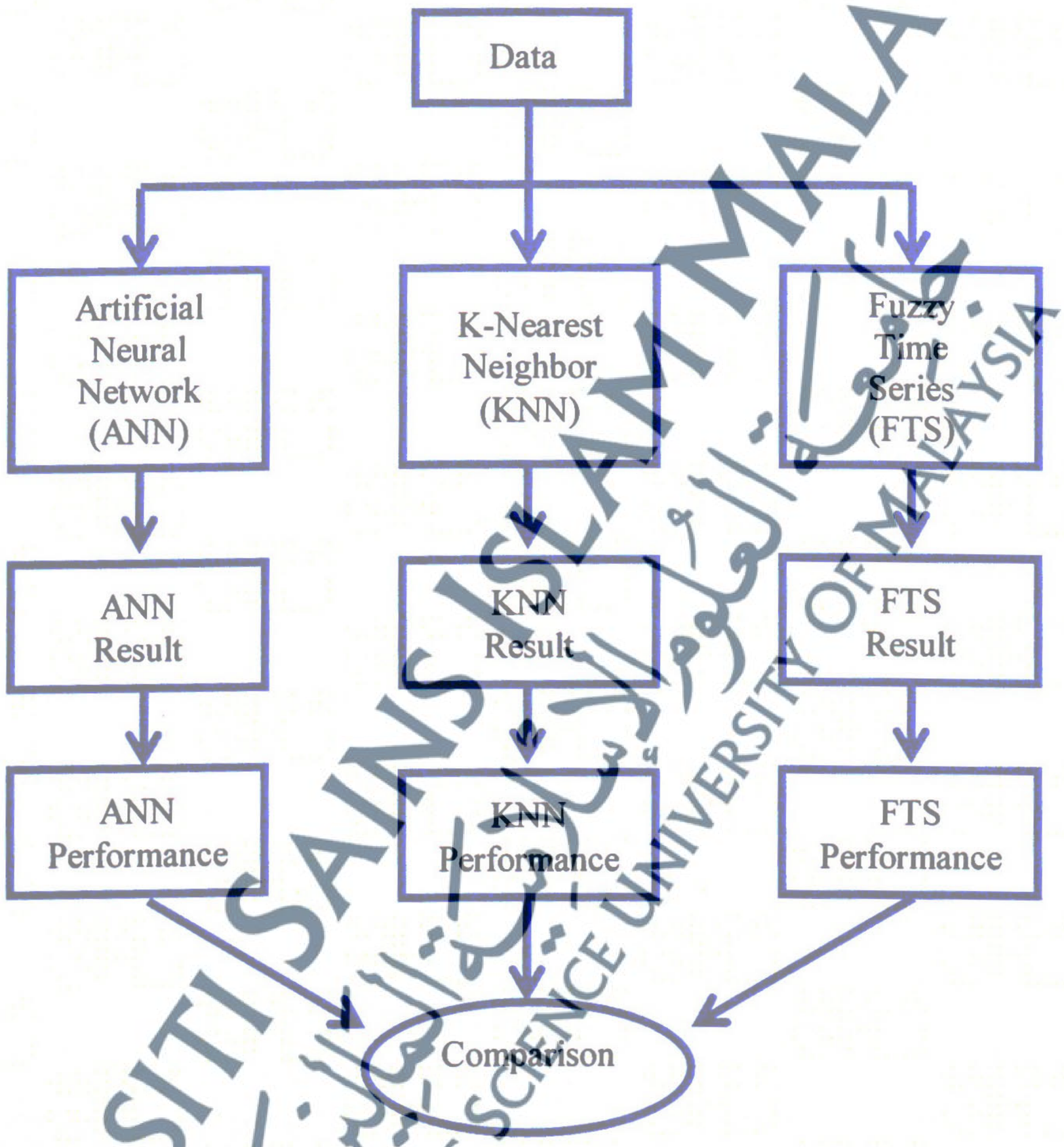


4.5 Evaluation

To evaluate the performance of the newly designed artificial neural network (ANN) in WiMAX traffic forecasting, its accuracy is compared against the other two approaches k-nearest neighbor (KNN) and fuzzy time series (FTS). This is done by using the same data set for all the three approaches. The result for each approach is then analyzed to ascertain

the performance. Finally, the performance of all the approaches is compared to determine the best one (Figure 28).

Figure 28: Evaluation



4.6 Summary

A total of three different techniques of performing intelligence forecasting are covered within the methodology to know the best forecasting. In principle, the simplest approach is the k nearest neighbor that learns prediction by examining a set of given examples. Comparatively, artificial neural network is the most complex approach of which two

different algorithms (Levenberg-Marquardt optimization and scaled conjugate gradient method), are explored to attain the best possible prediction ability. Exponential fuzzy time series is an extension of fuzzy logic that incorporates fuzzy logic relationship and grouping to ascertain the salient rules in deriving the predictive component in forecasting.