CHAPTER II

LITERATURE REVIEW

2.0 Introduction

In this chapter, a brief exploration of the WiMAX technology and traffic forecasting techniques are performed. It focuses mostly on the potential of WiMAX in offering a workable alternative to the excessive requirement of bandwidth posed by the high Internet usage in the present world. In doing so, the network architecture as well as the main topologies is explained in relation to the performance of the network as a whole. This is imperative in fulfilling the quality of service of WiMAX on a continuous basis.

The next section deals with the network traffic and forecasting of WiMAX. It involves the elucidation of why network traffic is vital in maintaining a high quality network to the users. Not just that, it also covers the main stages of network forecasting, which will become a base for the actual implementation of forecasting from the respective technological aspects.

In this research, there is a major paradigm of forecasting that would be emphasized. It is intelligent forecasting exploits artificial intelligence to actualize a prediction. Three approaches of intelligent forecasting are delved into. They are artificial neural network, k nearest neighbor, and fuzzy time series.

2.1 Over view of WiMAX

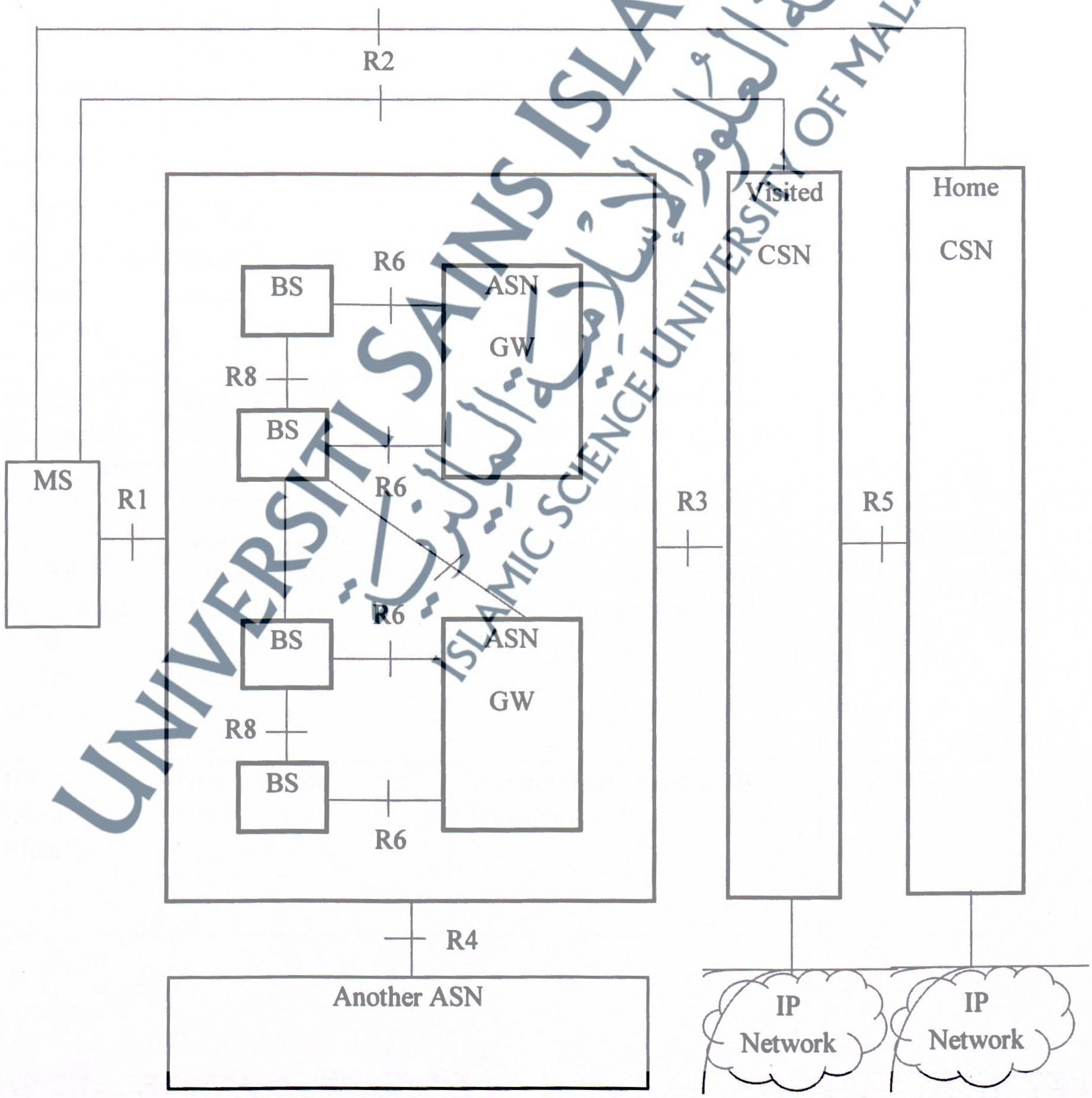
Worldwide interoperability of Microwave Access (WiMAX) is a popular fast wireless communication standard that is gaining unparalleled prevalence in the technological world. It boasts a high data rate of making it highly suitable to the exponential demand of network users nowadays. However, it can be classified into IEEE 802.16a, IEEE 802.16c, IEEE 802.16d, IEEE 802.16e, and lately IEEE 802.16m for mobile devices.

Given its capacity to support impressive bandwidth, WiMAX is utilized for a number of main applications such as VoIP (Qureshi et al., 2011), video streaming (Hrudey, 2009) or digital subscriber line, and portable broadband (Shen et al., 2007) for mobile connectivity. This is rather compelling with the rapid increase of nomadic Internet users that rely heavily on the consistent availability of Internet connection. All IP network and wide coverage further support its quick adoption as one of the most popular wireless technologies nowadays.

During the last decade, smartphones have gained popularity all over the world. In July of 2013, over 50% of mobile subscribers in the US and over 65% of mobile subscribers in South Korea are using smartphones, and these percentages keep increasing (Heejune et al., 2013). The key differences between smartphones and previous mobile phones are full-featured Internet access and easy installation of new applications through modern OS platforms and app stores.. A majority of them use the Internet on a frequent basis throughout the day, which poses a huge demand for mobile connectivity. The popularity of certain applications like Youtube (Chetty, et al., 2012) and Instagram are compounding the strain on bandwidth demand when each involves considerably large data transfer to work successfully. WiMAX provides a practical solution to this.

The network architecture of WiMAX (Iyer et al., 2007) can be presented via the network reference model (Figure 2). It consists of MS (Mobile Station), BS (Base Station), ASN GW (Access Service Network Gateway), CSN (Connectivity Service Network), and a series of interfaces R1... R6 that connect distinct elements within the model in a particular way. Understanding the architecture is crucial when it comes to analyzing the flow of traffic within the network in supporting its quality of service.

Figure 2: WiMAX Network Architecture (Iyer et al., 2007)



Quality of Service (QoS) is basically a contract that ensures a particular degree of performance is likely to be exhibited by a network (Anouari & Haqiq, 2012; Cicconetti et al., 2006). Although there are exceptions that are not entirely avoidable, QoS implies that the network will provide the stipulated quality level most of the times. There are five services offered in WiMAX (Table 2). In order to ensure that the QoS is achieved, the forecasting of network traffic must be addressed for all the possible topologies of WiMAX.

Table 2: Quality of Service (QoS) for WiMAX

Service	**Description**	**Parameter**	**Example**
UGS (Unsolicited Grant Service)	Real time Fixed size data packet	Maximum sustained rate Minimum reserved rate Jitter tolerance.	VoIP (without suppression)
rtPS (Real Time Polling Service)	Real time Variable size data packet	Maximum sustained rate Minimum reserved rate Maximum latency tolerance Traffic priority	Video streaming
ertPS (Extended Real Time Polling Service)	Real Time Variable size data packet. Extension of rtPS to improve traffic flow.	Maximum sustained rate Maximum latency tolerance Jitter tolerance Traffic priority	VoIP Voice Activity Detection
nrtPS (Non-Real Time Polling Service)	Non real time Delay tolerant. Variable size data packet	Maximum sustained rate Minimum reserved rate Traffic priority	HTTP FTP
BE (Best Effort)	Non real time	Maximum sustained rate Traffic priority	Email

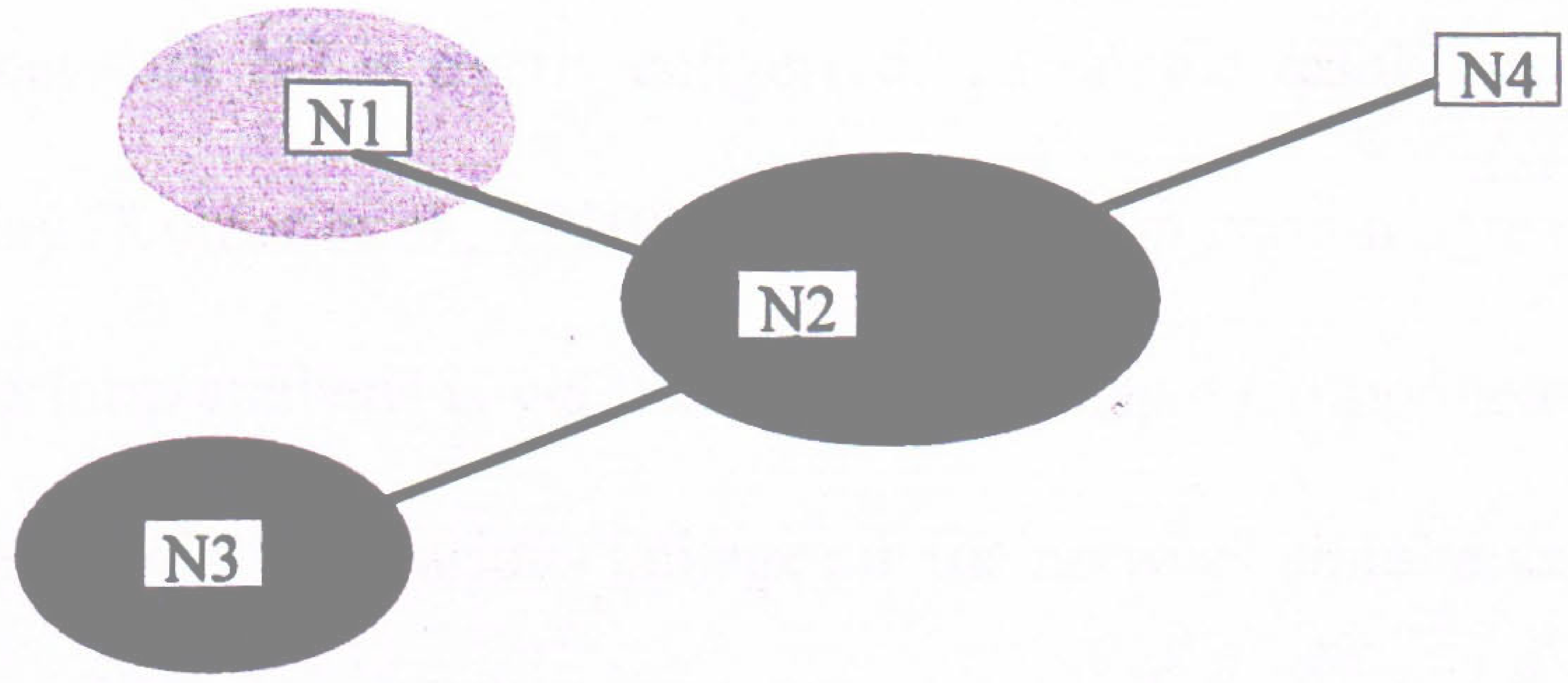
2.2 Network Traffic Forecasting

Traffic forecasting plays an important role in design, management, and optimization of modern telecommunication systems. Accurate and reliable forecasts allow for the planning of the capacity of a network to be on time while sustaining the required level of quality of service. Besides, the properties of network traffic directly influence both the capital costs of equipment and expected income of an operator.

When communication occurs between entities in a network, data is transferred between them. The transfer of data impacts the distribution of density within the network. In other words, certain parts of the network will experience a high amount of data flowing within them. On the other hand, there are portions that are not affected at all. This is what network traffic entails (Benson et al., 2010).

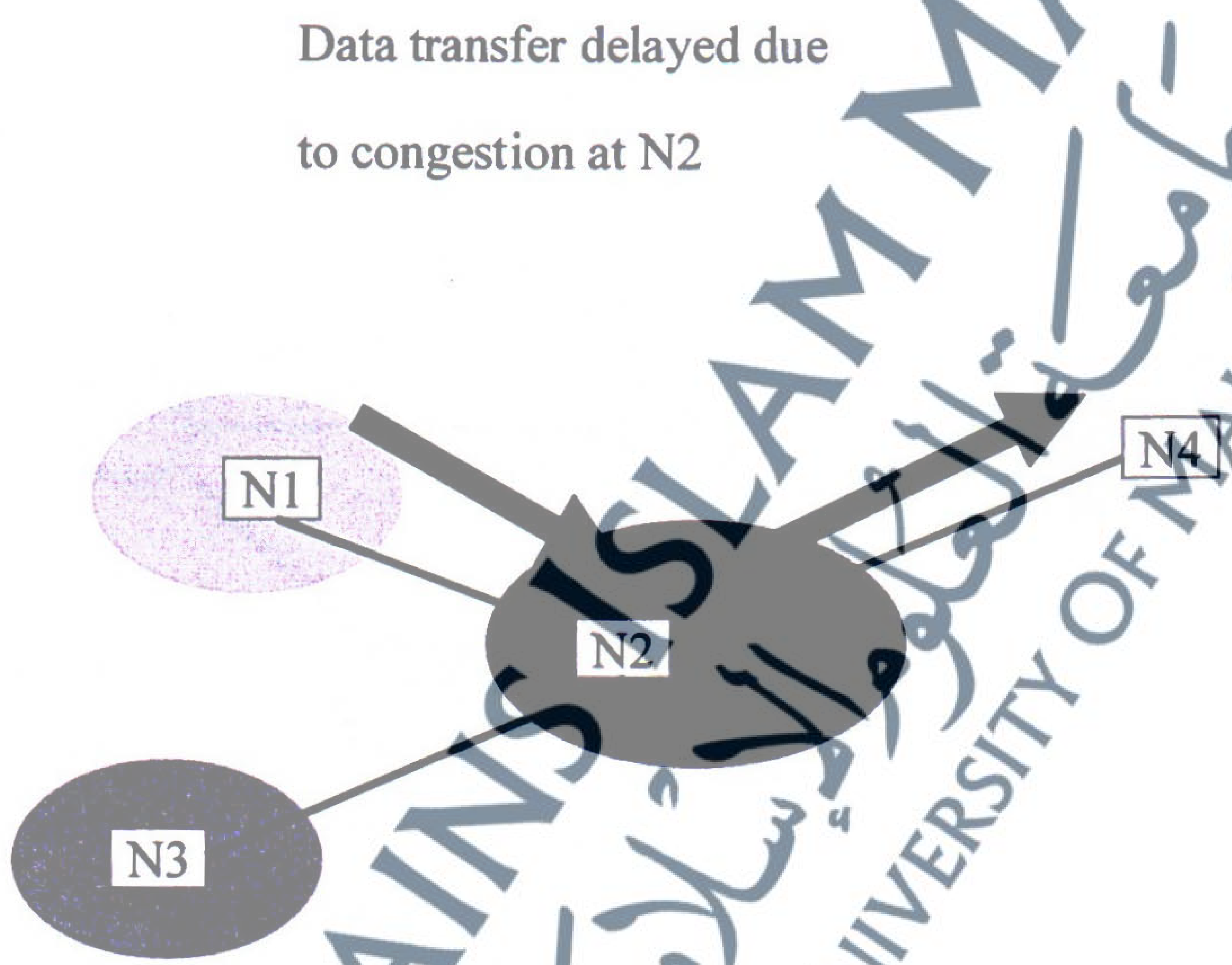
Network traffic signifies the concentration of data transfer within a network (Kalyankar, 2009). Let us assume that a network N consists of a four subnetwork N1... N4 that are connected to one another as shown below (Figure 3). Each subnetwork is associated with its own concentration of data transfer, represented by the encircling region. The degree of concentration is defined as the color for the region. Darker colors imply higher concentration and vice versa.

Figure 3: Varying Network Traffic of Subnetworks



The analysis of network traffic (Djidjev et al., 2011) is vital to guarantee that the transfer of data can proceed as expected. For instance, suppose data needs to be transported between subnetwork N1 and N4 (Figure 4). This can transpire only if the subnetwork N2 participates in the communication. However, if subnetwork N2 is highly congested, the data transfer can be delayed.

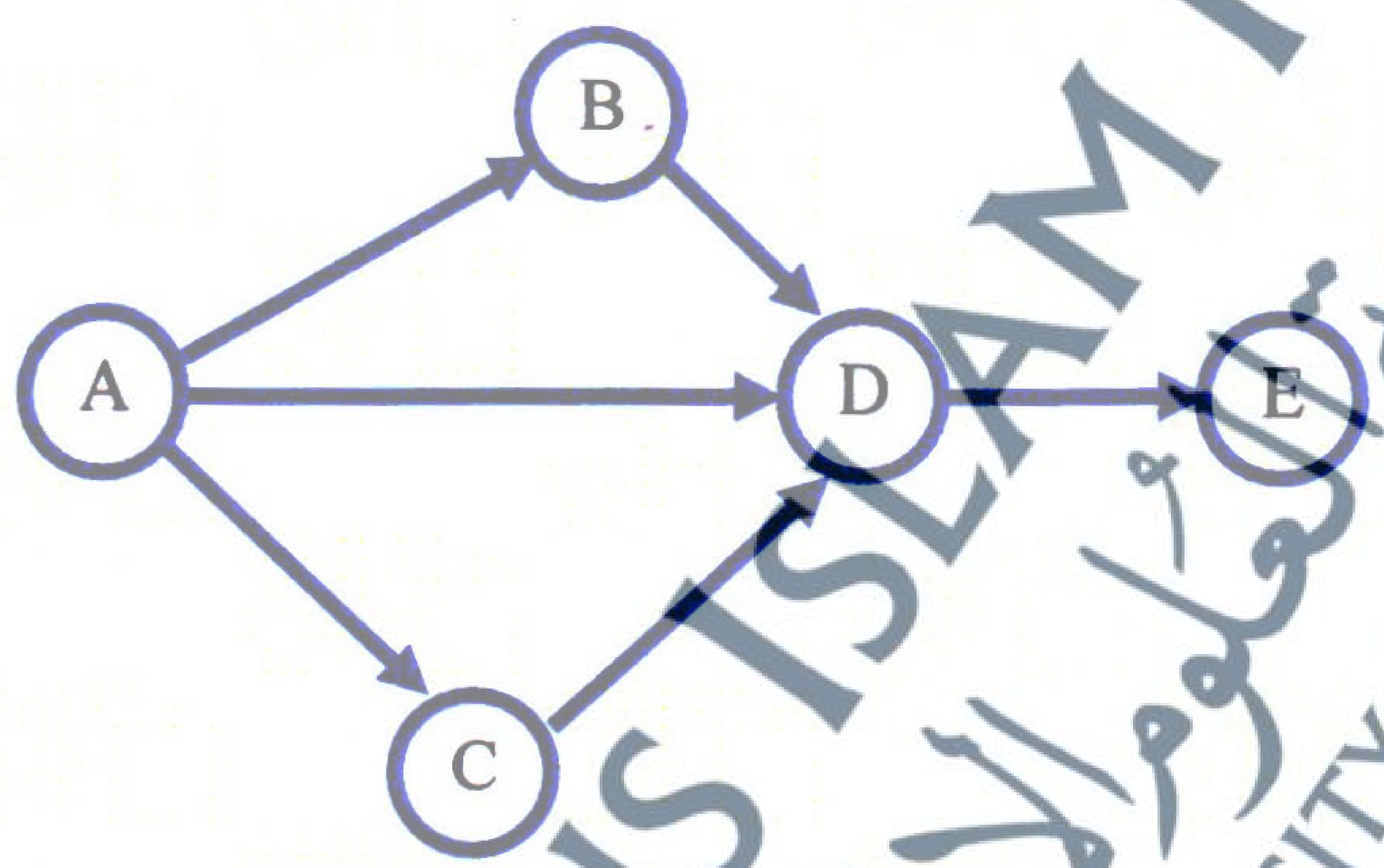
Figure 4: Data Transfer between Subnetworks



Network traffic analysis attempts to safeguard the performance of communication in challenging situations such as the one mentioned above. It allows data transfer to materialize in the best possible manner within a set of imposed constraints. This means that even if subnetwork N2 is overly congested, a strategic resolution must be enforced such that the delay (Kumar et al., 2010) does not exceed a certain agreed threshold. One of the ways to perform analysis is via the usage of Principal Component Analysis (Wanga et al., 2012), which examines various linkages in the network simultaneously.

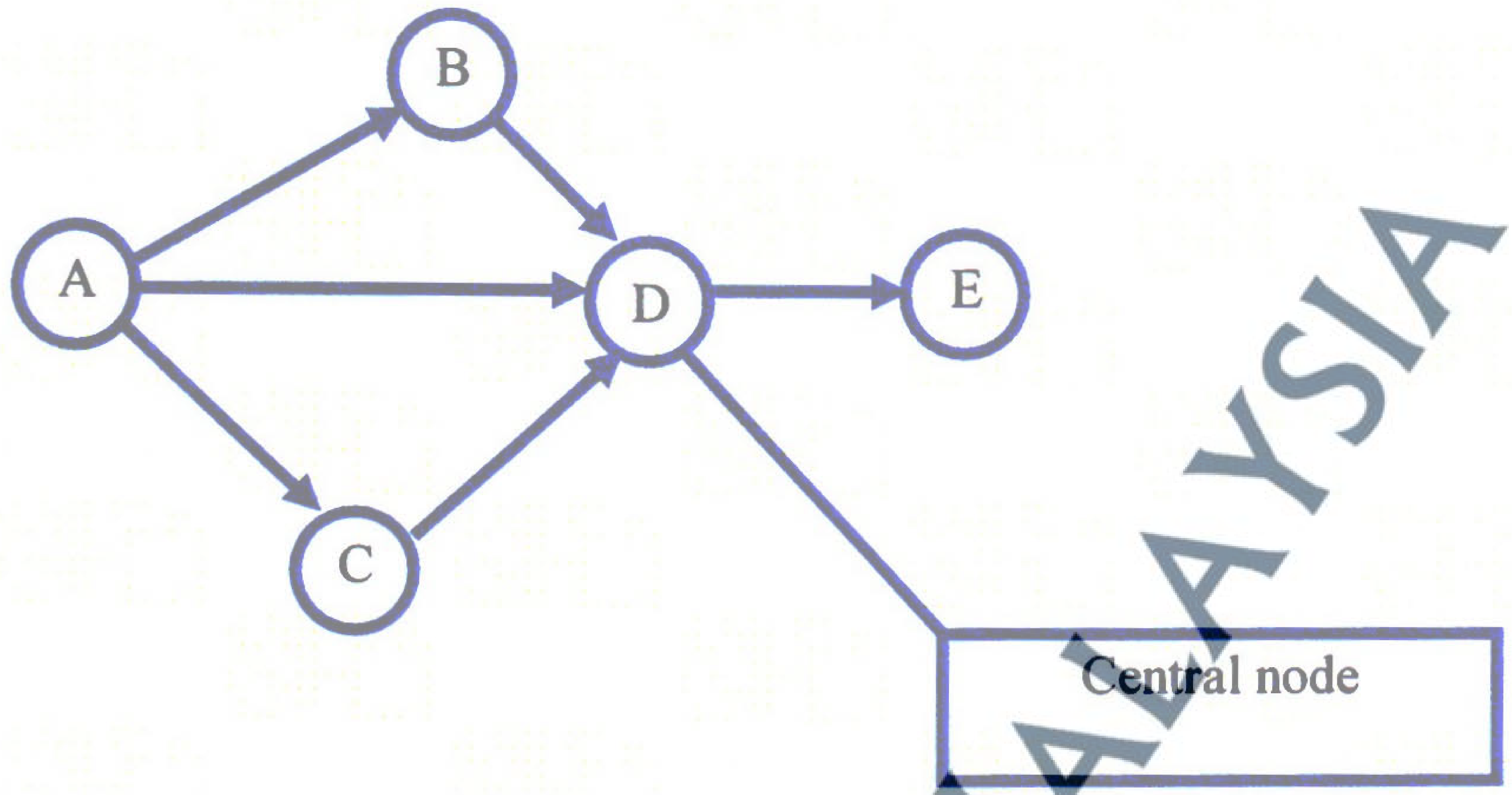
To illustrate the importance of network traffic with better clarity, consider the diagram below (Figure 5). For the sake of simplicity, assume that each node is a station that relays traffic and the edge between them denotes the traffic that occurred. As such, A → B implies that traffic transpired between A and B with the flow moving from A and B.

Figure 5: Network Traffic (Wanga et al., 2012)



From the diagram (Figure 6), it is not difficult to see that the central node for the network is node D. Unlike other nodes, the central node interacts with all the other nodes within the network. It becomes the point of intersection for the rest of them. This also implies that the central node would be highly susceptible to congestion. If this node is rendered paralyzed, the entire network would collapse.

Figure 6: Network Traffic and Central Node



The rationale implies that the traffic flow of the network depends on the availability of the aforementioned central node. In other words, for traffic to flow between A and E, the node D must be engaged. Each node has its capacity to process traffic, posing a constraint to the degree of traffic it can entertain. As such, the central node will face the highest strain among them. To recap, network traffic is the study of how much traffic flows between endpoints in a collection of nodes. It is crucial in determining the actual capacity of the network when contrasted with the demands. Having this at hand, the next step would be to predict or forecast the traffic. Forecasting the network is crucial for these purposes:

1. **Optimizing resources**

The construction of a network is often a trade-off between the factor of cost and performance. Given the limited resources that are available in any network, it is imperative to maximize its configuration. This ensures that the stipulated quality of service (Delannoy et al., 2009) can be met in the best possible way (Alarcon-Aquino & Barria, 2006).

2. Providing security

The flow of traffic in a network is potentially useful in detecting security breach. By comparing the actual and predicted traffic, the detection of anomalous behavior (Wang et al., 2009) such as a spam attack is properly conducted. Proper counter measures for providing security can then be implemented.

3. Planning and maintenance

In actuality, a network may not function as designed. It is therefore critical to ascertain the scenario of traffic in advance to enable better planning (Bauschert et al., 2014; De Deus et al., 2014). For instance, forecasting allows the discernment of bottleneck before it occurs. Having this information at hand can assist the act of prevention.

Although the method of forecasting would normally be technology dependent (Hoong et al., 2011; Oduro-Gyimah et al., 2013; Stolojescu-Crisan & Isar, 2014), the general process (Papadopouli et al., 2005) essentially remains the same. It consists of planning, collection, analysis, modeling, prediction, validation, and finally, forecasting (Figure 7).

Forecasting begins with the planning stage (Konstantinopoulou et al., 2000). In this stage, a panoptic plan of the whole forecasting procedure is outlined. Certain core decisions like the goal of the forecasting, duration involved, population for data collection are defined clearly. Having a proper plan ensures that nothing essential is overlooked that may risk the usability of the forecasting in the end.

Once a well thought plan is concocted, data collection (Ye, Szeto & Wong, 2012) can commence. This involves the recording of actual data from a particular traffic source. Data collection can be time consuming. Depending on the requirement of the plan, collection can span from a few months to more than a year. It is crucial to collect only the data that is needed for forecasting since privacy can be an issue.

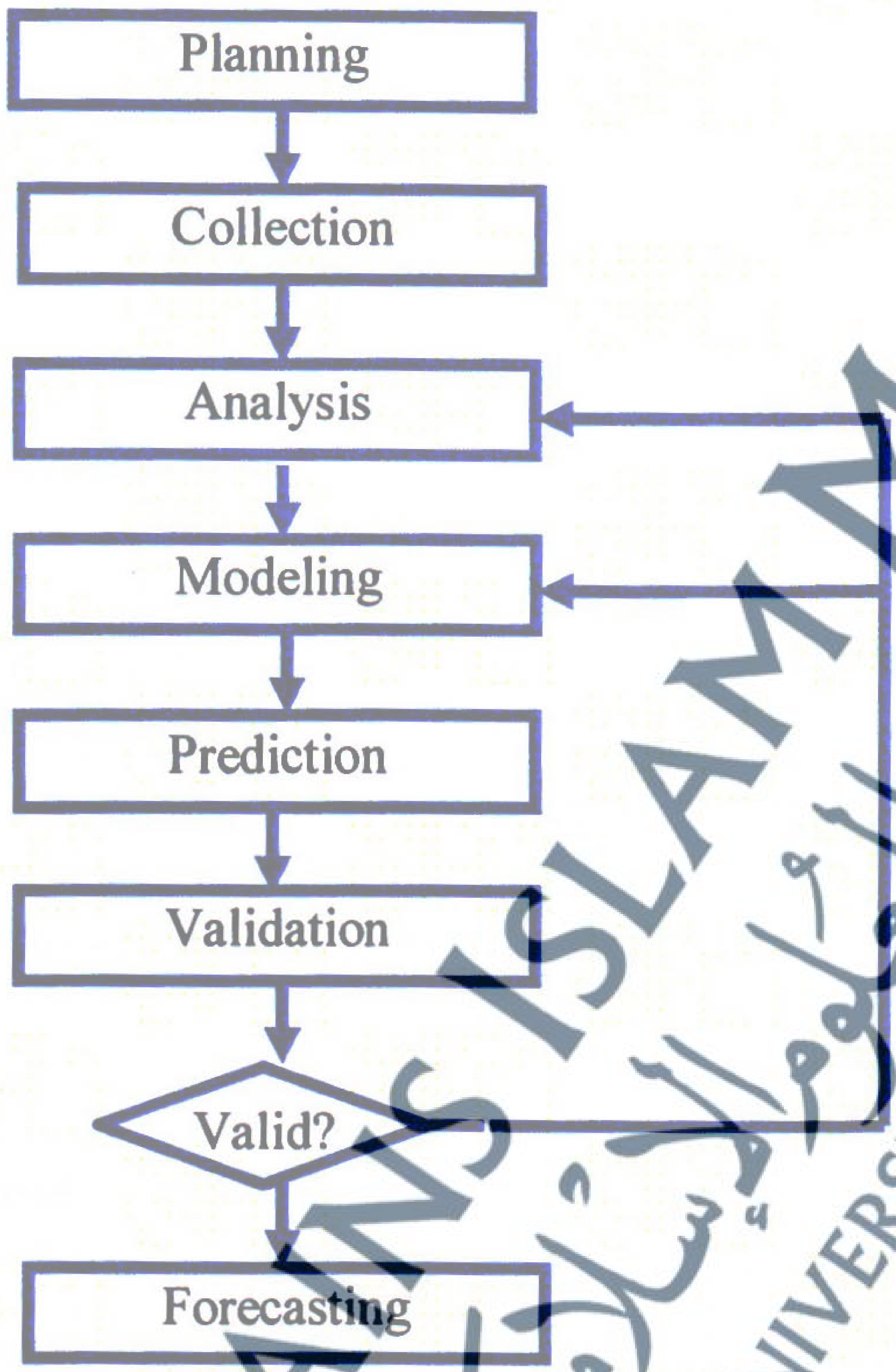
Analysis (Ekola et al., 2004; Kakuru, 2011) is one of the core processes in forecasting. The collected data is pre-processed and then analyzed according to the model to be used later. For instance, if Auto Regressive Integrated Moving Average (ARIMA) is utilized, the analysis would be oriented more towards a statistical stance. However, if other approaches such as Artificial Neural Network (ANN) are performed, the analysis would be rather computational by nature.

After analysis, the modeling stage is implemented. The network model (De-Godoy-Stenico & Ling, 2013; Laner et al., 2014) is the vital component that allows forecasting to happen. It must be able to dissect and make sense of the patterns gathered from the analysis of data. Invariably, this involves the exploitation of certain forecasting equations that can calculate the impact of certain parameters on the network traffic.

Prediction (Gao et al., 2012) is normally made possible when the forecasting model is available at hand. It should be able to previse the behavior of the network in terms of traffic based from a set of dynamic parameters such as time and locality. The accuracy of prediction is highly dependent on the analysis and modeling stage. If both are properly executed, the predicted traffic would exhibit minimal error.

Due to the importance of quality, validation (Barrow & Crone, 2013) is vital before the model is employed for actual forecasting. It is imperative that the model fulfills the standard established. As such, the entire process is repeated until the validation stage is successful. This entails comparing the quality of the prediction with the actual data collected. A certain level of minimal quality is normally defined here, which serves as the benchmark for the forecasting procedure. If validation fails, amendment is employed at the modeling part to improve the overall accuracy of forecasting.

Figure 7: General Process of Traffic Forecasting (Barrow & Crone, 2013)



Finally, forecasting (Bai et al., 2010) is available once the model reaches the stipulated contract of quality and passes the validation. Here, it can be harnessed to forecast the network traffic of a particular domain when given the appropriate parameters. Comparison between the forecast and the actual traffic data can still occur at this stage. However, this does not imply reanalysis or remodeling. In other words, the data and model do not go through the entire procedure. Enhancement is achieved merely by modifying some of the modeling variables or factors.

The implementation of the aforementioned process is dependent upon the technology employed. If the technology is statistically based, such as ARIMA, then the process would focus mostly on the selection of the most appropriate statistical model for the prediction task. This can be a challenging task, given that there are abundant models to choose from. In this aspect, the task is often performed manually.

On the other hand, the process of forecasting can also be done via artificial intelligence. Here, the most critical aspect is choosing the right method to conduct forecasting. Each method is unique in its own way and will eventually decide the rest of the process involved. For instance, artificial neural network models the working of the human brain. In this sense it is highly autonomous but non-transparent. As such, it will not require as much human intervention as its statistical counterpart. Most of the stages within the forecasting process defined here will be done automatically without the need of much supervision at the cost of increased opacity. This means that it would be difficult for the approach to explain the rationale behind a particular prediction.

2.3 Statistical Forecasting

Auto-Regressive Integrated Moving Average (ARIMA) model is a statistical approach in forecasting (Ghosh et al., 2005; Papagiannaki, et al., 2005; Hyndman & Khandakar, 2008; Oduro-Gyimah et al., 2013; Sirijunyapong et al., 2014). This method is derived from the Box Jenkins Methodology (Box & Jenkins, 1976; Nihan & Holmesland, 1980) in time series modeling. It analyzes time series data to provide a prediction of future events by manipulating the order of auto-regressive, differencing, and moving average part of the model.

Generally, the process of employing ARIMA consists of model selection, estimation, and checking (Box et al., 1994). It is iterated in a cycle of which the outcome can eventually result to several models, working on the same time series.

2.4 Intelligent Forecasting

Intelligent forecasting (Nikolopoulos & Assimakopoulos, 2003; Valenzuela et al., 2008 Shen, 2010) marks the employment of artificial intelligence for the specific purpose of forecasting. In this sense, it attempts to emulate a somewhat pseudo humanistic approach of predicting future outcome based on past ones. Reflecting upon how humans practice forecasting allows it to be more flexible than its statistical counterpart.

There are numerous methods of intelligent forecasting, which includes fuzzy logic (Aksoy et al., 2014), genetic algorithm (Bas et al., 2014), support vector machine (Kim, 2003) etc. More often than not, intelligent forecasting is domain sensitive. Thus each approach has its own strengths and weaknesses, depending upon the context of application. It is therefore imperative to select the most appropriate technology for the task at hand.

For the purpose of this research, three main approaches of intelligent forecasting will be considered. They are artificial neural network (ANN), k- nearest neighbor (KNN) and fuzzy time series (FTS). The mechanism of ANN is primarily adopted from the working of the neuron in the human brain. It is highly adaptive to changing situational parameters within the environment. Compared to ANN, the process of KNN is more simplistic by nature, which compares the incoming event with the ones before, in terms of their corresponding distance. Finally, FTS provides a more elastic representation of time and synthetic reasoning for the traditional time series.

2.4.1 Artificial Neural Network (ANN)

One of the most prominent technologies in artificial intelligence is the artificial neural network (ANN). It is being used widely in diverse areas for varying applications such as prediction, classification, optimization (Betiku & Taiwo, 2015), clustering, and control (Kumar et al., 2014). Even though each application exploits the data for different purposes, ANN can be adjusted accordingly. As such, the strength of ANN lies in its ability to be adaptive in learning from many kinds of data without much intervention.

For instance, ANN shows impressive agility when deployed for medical purposes (Torbati et al., 2014), such as predicting the probability of a disease (Vukicevic et al., 2014) by processing a range of multifarious data that is gathered from the patient. The versatility of ANN to accommodate this challenging diversity contributes to its constant popularity among researchers.

The most apparent drawback in using ANN is the lack of visibility with regard to explanation (Saad & Wunsch II, 2007; Green et al., 2009). In other words, although ANN is superior in finding the solution to a particular problem, it may not be as good when it comes to explaining the rationale as to why a certain choice was made when compared against the rest. This can be a staggering source of confusion. Especially, when the output suggested by ANN offends common sense or seemingly logical reasoning. Puzzling recommendation however, does not necessarily imply falsity. It might still be correct but remains implausible.

Having the potential of being deployed on many applications, flexibility is a key factor in artificial neural network. It offers two architectures to align itself with different domains. The recurrent networks exploit cyclic processing while the feed forward network does not. For the purpose of this research, emphasis is given more on the feed forward network

(Sodhi & Chandra, 2014). Specifically, on the multilayer perceptron network, which is generally comprised of three layers.

1. Input layer

The input layer receives the data to be processed by the network. Nodes in this layer do not perform any kinds of modification on the data. Their main function is to duplicate the input to the complete multitude of nodes within the hidden layer. Due to this, the input layer is often deemed inactive (Smith, 1997). It only performs replication and the diffusion of data between layers without any processing engagement.

2. Hidden layer

Data from the input layer is relayed to the hidden layer (Matias et al., 2014) for the actual processing. A node in the input layer and all the other nodes in the hidden layer are connected through a series of links with weights. During learning, the weights are adjusted to better approximate the mapping function between the input and output. The number of sublayers within the hidden portion of the neural network is not predefined. In principle however, the total sub-layer is increased when there is a greater need to improve accuracy (Karsoliya, 2012). However, doing so would leave the network more susceptible to the problem of over fitting. As such, there must be a balance between accuracy and coverage when it comes to deciding the optimal number of sub-layers within the hidden layer.

3. Output layer

Computation that occurs in the hidden layer is finally integrated within the output layer to ascertain the result. The outcome from this layer must coincide with the expectation of the external entity that is exploiting the neural network. In other words, the result generated here by must be process able outside the neural network. This is crucial due to the simple fact that the reasoning within the neural network is considered a black box. It is not computable beyond itself.

Mathematically, the artificial neural network relies on the extent of difference exhibited by the mean square error (Rady, 2011) between the desired output d and the one generated y. Amendment on the network is driven by the error such that the total discrepancy that exists between expected and actual is minimized.

The extent of which the artificial neural network requires supervision in learning is defined as the learning paradigm. Although complete automaticity is a desirable trait, it is not always a viable agenda. Now, there are three main learning paradigms:

1. Unsupervised learning

Unsupervised learning (Chakraborty & Chakraborty, 2000; Herrero et al., 2011) occurs when the correct output is not given to the neural network during the process of training. This implies that the neural network must be able to explore the data and find the pattern of relationship by itself. Patterns found are then used to make the connection explicit.

2. Supervised learning

Compared to unsupervised learning that does not offer any examples of valid output to the neural network, supervised learning (Taghavifar & Mardani, 2014) works based on a set of correct output given externally. Having the targeted output encourages the neural network to modify itself to a specific behavior that reflects a consistent association between the input and output.

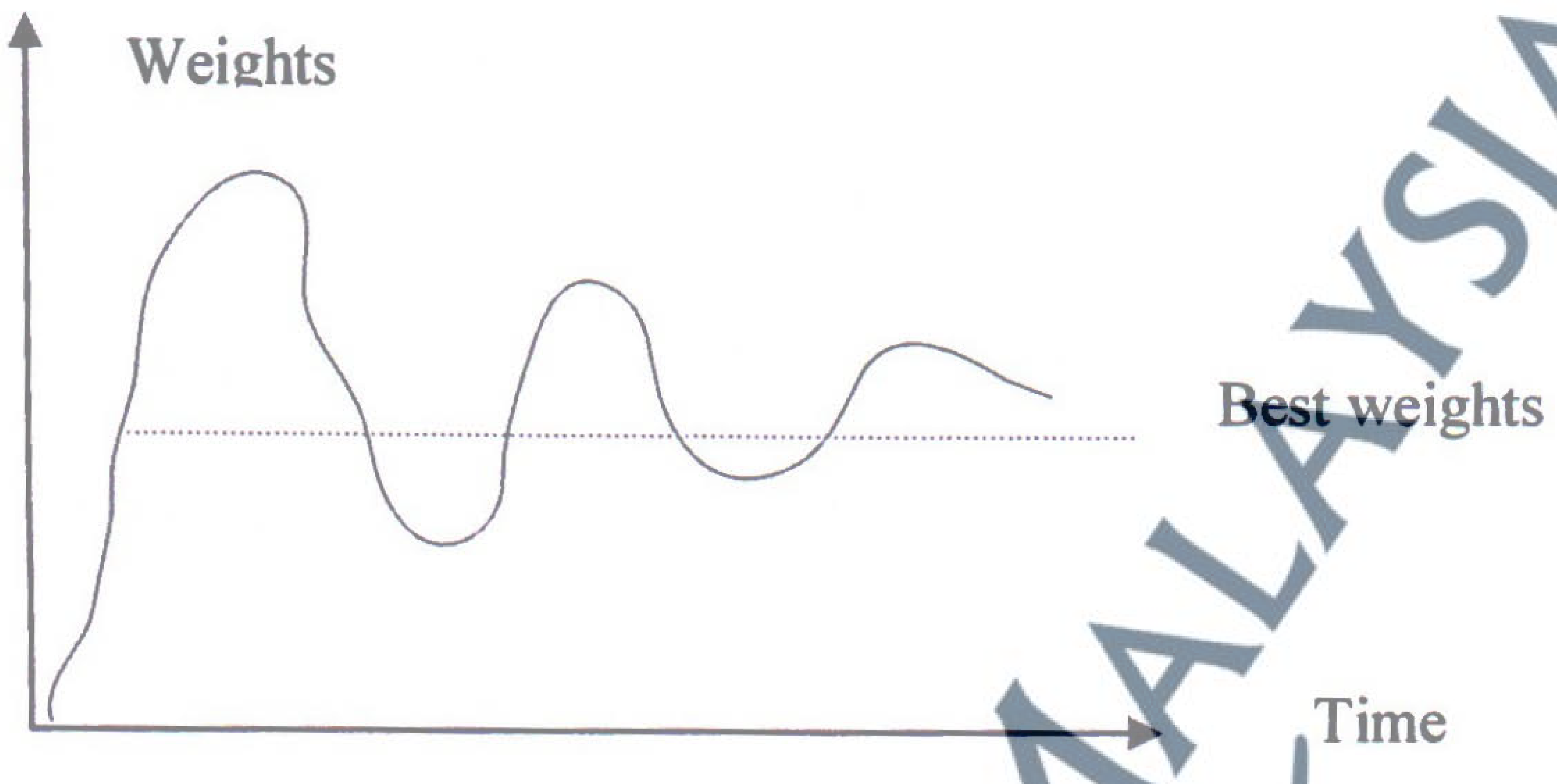
3. Hybrid

A hybrid learning paradigm (Teh & Tapan, 2008; Castelli & Trentin, 2014) practically integrates both unsupervised and supervised learning. Certain portion of the weights within the neural network is governed by unsupervised learning while others by supervised learning. This can create some issues in congruence. As such, coordination is extremely vital.

To learn, the artificial neural network compares the generated output with the desired output (Tay et al., 2007; Abdul Hamid & Abdul Rahman, 2010). The difference between the two is called the error. In this respect, learning is performed by reducing this discrepancy progressively. It is possible to minimize the error by changing the weights between the neurons in successive iterations. Lowering the error through evolution rather than revolution is important to ensure that the best solution is not accidentally bypassed.

Demonstrating the principle of incremental learning through weights updating requires the use of a graph (Figure 8). Without the loss of generalization, assume that the weights are initialized at zero. The best weights that would result to the least error between the generated output and desired output, is signified by the dotted line. Notice that the weights are concurrently updated again and again until they approach the best solution (Kumar et al., 2014). This is how the learning rule works through error correction.

Figure 8: Learning Rule Illustration (Kumar et al., 2014).



The generic algorithm for learning (Rafiq et al., 2009; Shaban et al., 2009) in artificial neural network is given in Figure 8. It contains the pattern of network which is the hidden layer that processes the input into output. The generated output Y is contrasted against the desired output D. If the discrepancy (D -Y) is higher than the threshold of error E, then the iteration begins.

ANN LEARNING ALGORITHM

INPUT= I = $I_1 \ldots I_N$

PATTERN = X = $X_1 \ldots X_N$

OUTPUT = Y

GAIN= Π = (0, 1)

D= DESIRED OUTPUT

Y= GENERATED OUTPUT

E= ERROR THRESHOLD

Y = PROCESS (I, X)

WHILE (| D – Y | > E)

FOR EACH W (X_A, X_B), W (X_A, X_B) = W (X_A, X_B) + Π (D – Y)

ENDFOR

ENDWHILE

In each iteration, the weight of the connection $W(X_A, X_B)$ between a particular node X_A and X_B is updated by increasing it proportionately to the product of the gain Π and error $(D - Y)$. The value of the gain is normally defined in a small range to ensure that the process of error minimization would operate smoothly without any abrupt modification (Liu et al., 2010). This emulates the biological nature of a neuron in adapting to changes that transpires gradually through time.

2.4.2 K-Nearest Neighbor (KNN)

Human beings tend to learn from examples (Selbing et al., 2014). They tend to emulate what is prevalent around them. When a person is in a new situation and unsure of the right way to behave, he will direct his attention to the people around him. This form of learning is quite fluid. It allows learning to occur in the most complex circumstances without the need to understand the actual mechanism of the behavior to be acquired.

Instance based learning (Zheng et al., 2010) works in a rather similar fashion as the aforementioned illustration. It does not generalize or iterate a complicated series of interconnection as an artificial neural network would. Instead, it simply explores the possible set of instances available, chooses the best one based on certain criteria and decides to follow the outcome demonstrated by the selected instance.

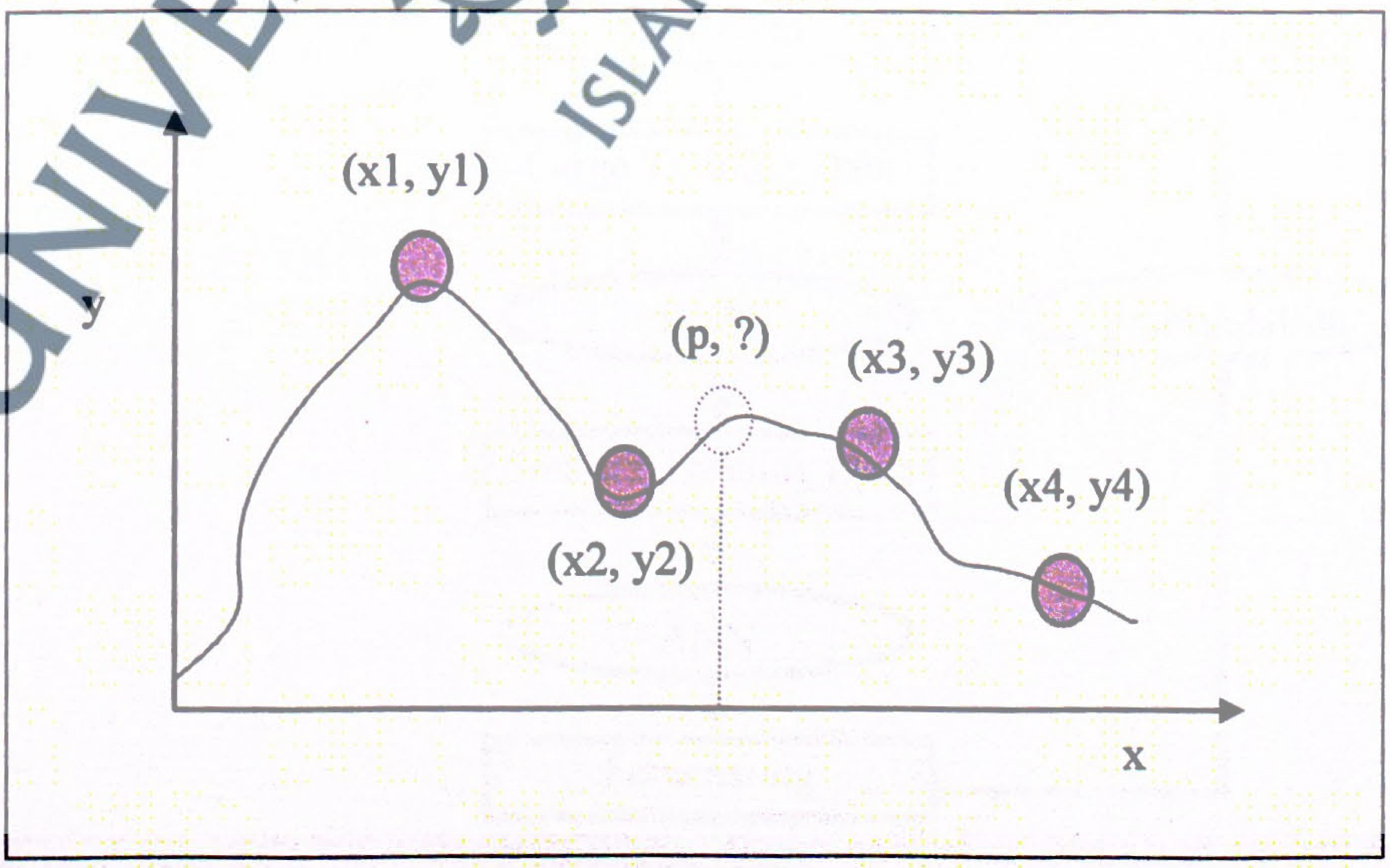
K-nearest-neighbor (KNN) is a type of instance based learning. It actively elects a number of instances k as the examples to follow from the rest that are present. This way, learning can transpire almost immediately as the examples are provided.

Given its innate simplicity, KNN is used widely in various applications. For instance, in computational linguistics it is employed to classify a document into its corresponding genre (Guo et al., 2004). In medicine, it is utilized to analyze medical imaging (Mangai et al., 2013) as well as to diagnose patients with heart disease (Shouman et al., 2012). Finally for engineering, KNN can be used to monitor road networks (Guohui et al., 2010).

Once the nearest neighbors are known, the algorithm proceeds by finding the majority class of the data involved (Jiang et al., 2012). For instance, supposed that k = 3 and X_1, X_4, X_8 are the nearest neighbors with the corresponding class of $CLASS(X_1) \rightarrow C_P$, $CLASS(X_4) \rightarrow C_Q$ and $CLASS(X_8) \rightarrow C_P$. Since the class C_P is the majority, it is decided that the unknown data Y belongs to the C_P class.

The idea of exploiting KNN for forecasting (Sharma & Sharma, 2012; Imandoust & Bolandraftar, 2013) combines the notion of classification and regression. Consider the simple graph below (Figure 9). In the graph, assume that there are four available classes within the universe, namely (x1, y1), (x2, y2), (x3, y3), and (x4, y4). If this is the case, then the predicted value for the missing portion of the variable (p,?) is determined by classifying it to the nearest class and then, inferring the possible value that is related to it.

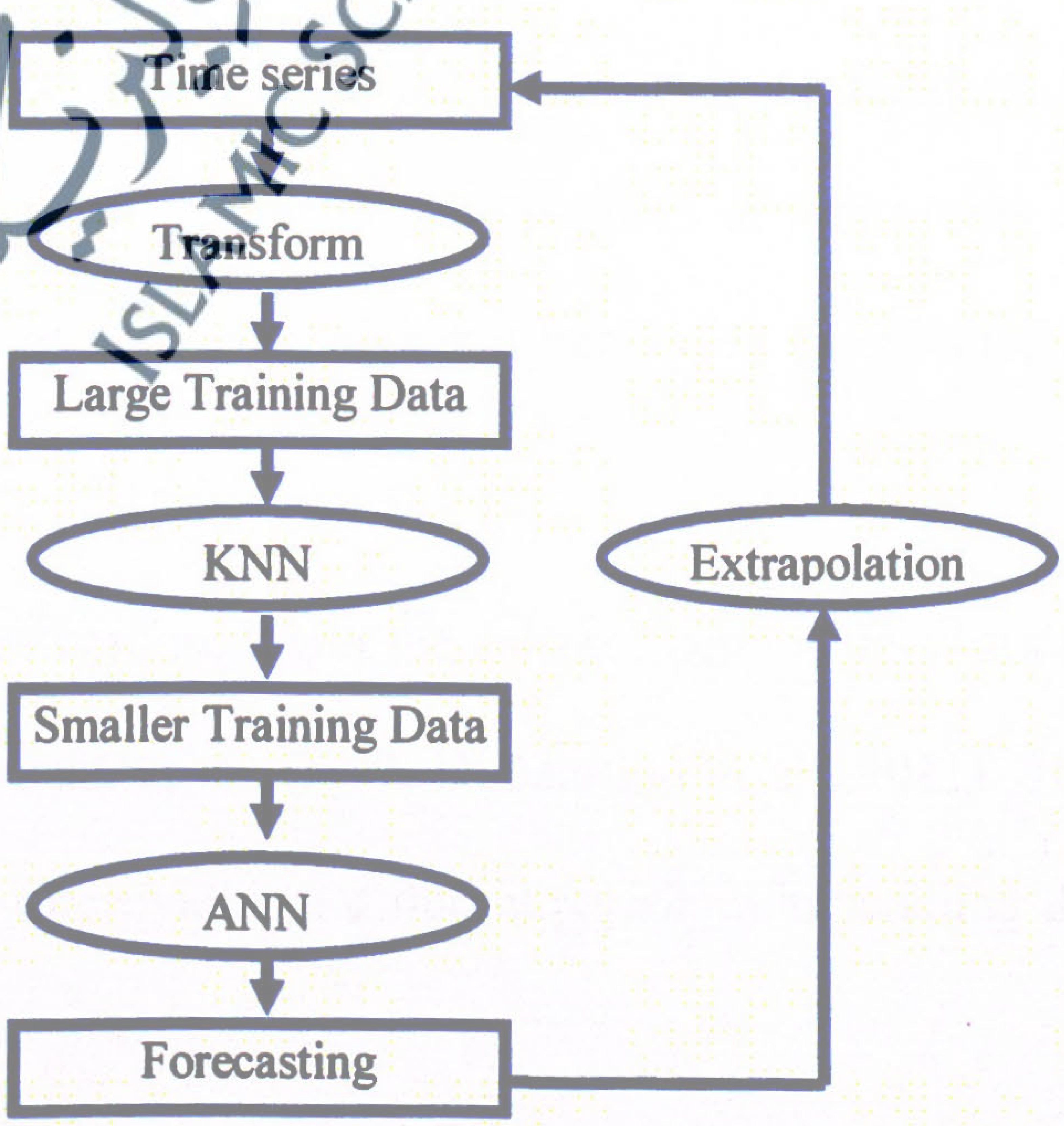
Figure 9: Example of KNN in Forecasting



It is quite clear that (p,?) lies between (x2, y2) and (x3, y3). As such, the use of KNN in this situation is to decide which class is more conducive for the classification. Since it is nearer to (x2, y2), it will be classified to (x2, y2). Here, the predicted value for (p,?) would then be (p, y2) instead of (p, y2). This is merely a simplified illustration of applying KNN for prediction. The actual application of KNN for prediction may be more complex (Ren & Suganthan, 2014).

To enhance the performance of forecasting, KNN can be coupled with other machine learning approaches such as artificial neural network (Tongal, 2013) and support vector machine (Hao et al., 2006). It is found to be temporally synergistic with artificial neural network (ANN) in forming a variant known as the k nearest neighbour based local linear wavelet neural network (Lina et al., 2013). This particular method, although bearing comparative accuracy to the original one, offers a significantly faster computational complexity as a whole. Below is a procedural overview of the synergy (Figure 10). Here, it can be seen that KNN minimizes the training data required by ANN, which eventually shortens processing time.

Figure 10: Synergy of KNN and ANN in Forecasting



Unlike time series approaches that are rather sensitive to noise, the k nearest neighbour method is reasonably resistant to anomalous data (Maier et al., 2009; Weinberger & Saul, 2009). It can accommodate noise well by measuring the closest distance to a pivotal set of anchor data that serves as the reference point of decision making. As such, although a multitude of noise may attempt to interfere, as long as the axial of reasoning is undisturbed, the method would still be able to maintain its accuracy.

2.4.3 Fuzzy Time Series (FTS)

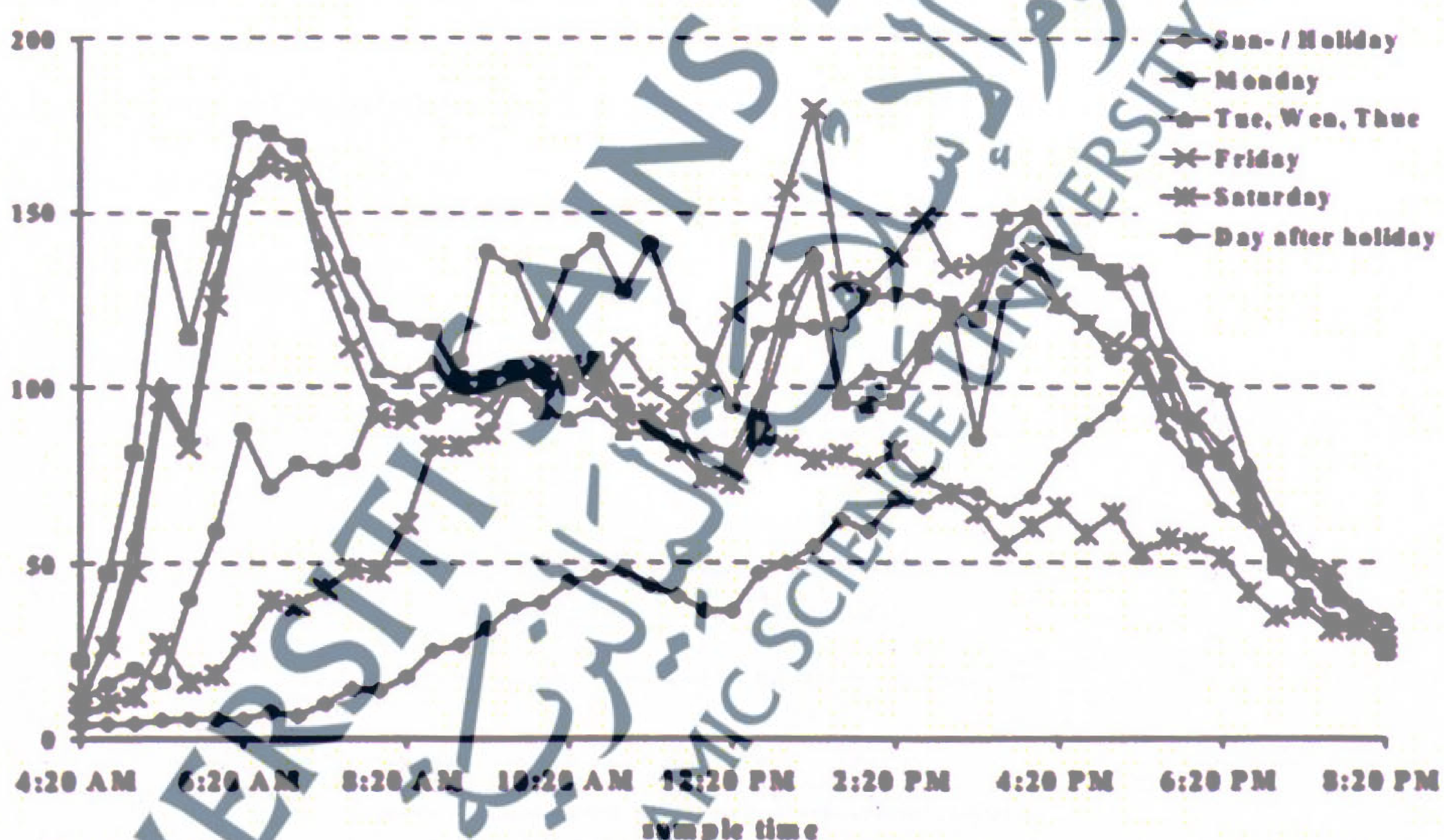
The main idea of time series in forecasting (Contreras et al., 2003) is to exploit a collection of data from past events to predict future events. It organizes data based on a single representation of time. In simple terms, events are all attached to one continuum of time. This can be quite problematic because patterns of events are not dispersed equally through time. Instead, they can overlap with one another through a multifaceted duration of temporal progress.

As a way of solving the challenge mentioned, fuzzy time series (Duru & Bulut, 2014; Lee & Hong, 2015) had emerged. It provides a multilayered conceptualization of time of which different patterns can be perceived individually and collectively. Time and the corresponding events are decomposed based on their dominant features. In effect, it allows greater versatility to forecasting whereby the nuances of past events are analyzed more intricately in the effort of achieving higher accuracy on the prediction of future events.

Fuzzy time series should not be confused with the enhancement of time series with fuzzy sets or fuzzy transform (Stepnicka et al., 2009; Di Martino et al., 2011). The former is mostly concerned with the act of providing a different paradigm in handling the divergent trait of events within time. This implies that it will no longer retain the classical notion of time as suggested by time series. However, the latter still maintains the elemental notion of time series but offers a more enhanced mode of processing that exploits fuzzy logic.

In the illustration (Figure 11), the traffic flows of different temporal subclasses are derived from a single time-series. Subclassing can be performed through a number of means such as clustering (Zhang & Zhu, 2012) or expert definition. There are six subclasses in total. They are Sun-Holiday, Monday, Tuesday-Wednesday-Thursday, Friday, Saturday and finally Day-after-holiday. Decomposing a single time-series into its subclasses can improve the versatility of forecasting.

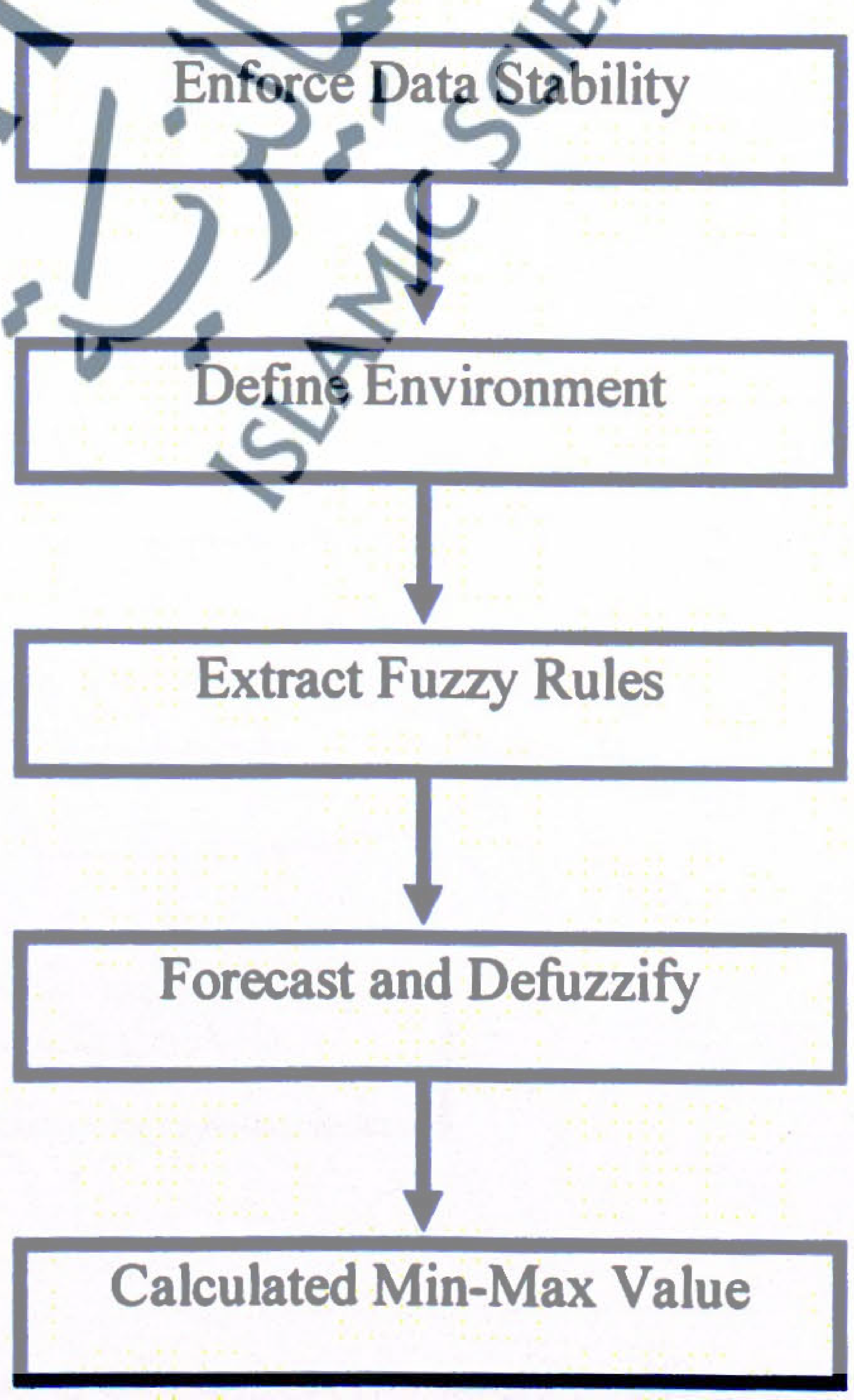
Figure 11: Fuzzy Model of Traffic Flows (Warg, 1997)



Traditional time-series would portray the entire collected data as a single class entity (Christodoulos et al., 2010), which may not be entirely commendable as the varying characteristics of each subset of data will be eventually shadowed by its general attributes. This can be quite detrimental to forecasting because it desensitizes the implication of salient patterns within the congregation. Consequently, it reduces the accuracy of forecasting as a whole.

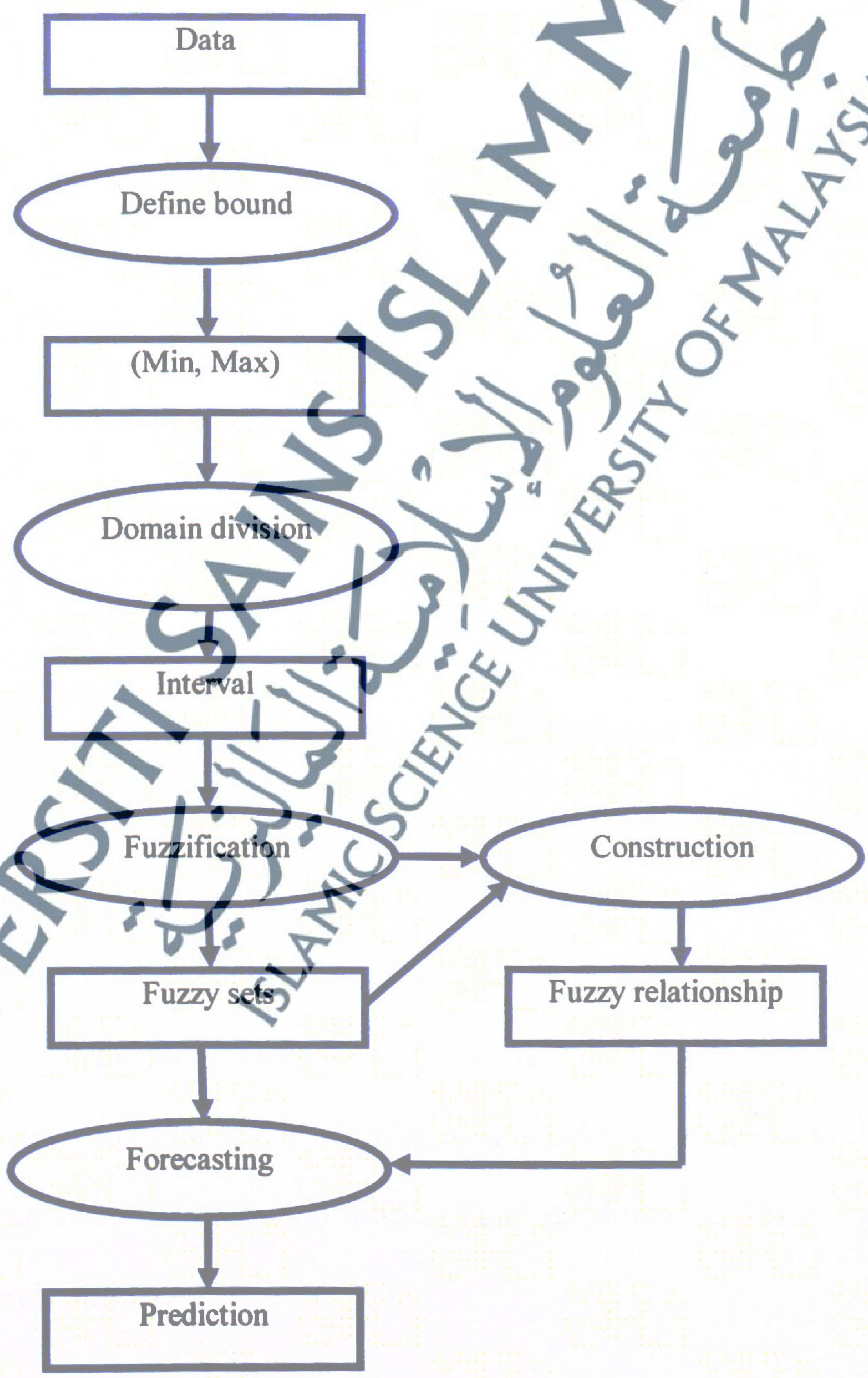
A demonstrative process of applying fuzzy time series in forecasting (Chou, 2011) is given below (Figure 12). It begins by enforcing stability on the data set. If the data is unstable, it is amended recursively until the requirement is fulfilled. Then, the environment is properly defined. This involves the act of explicitly enumerating all the attributes and values that will be used in the calculation, which includes the fuzzy sets (Gangwara & Kumar, 2014) employed in the overall reasoning. Once the environment is well defined, the procedure of extracting fuzzy rules is executed. The rules dictate which fuzzy set will be the dominating factor in a particular situation. Having these rules, forecasting becomes a possibility. The result of forecasting is then defuzzified to imbue sense making. Finally, the long term predictive value interval of the forecasting can be deduced. It consists of minimum and maximum value of the entire forecast.

Figure 12: Process of Long Term Predictive Value Interval (Chou, 2011)



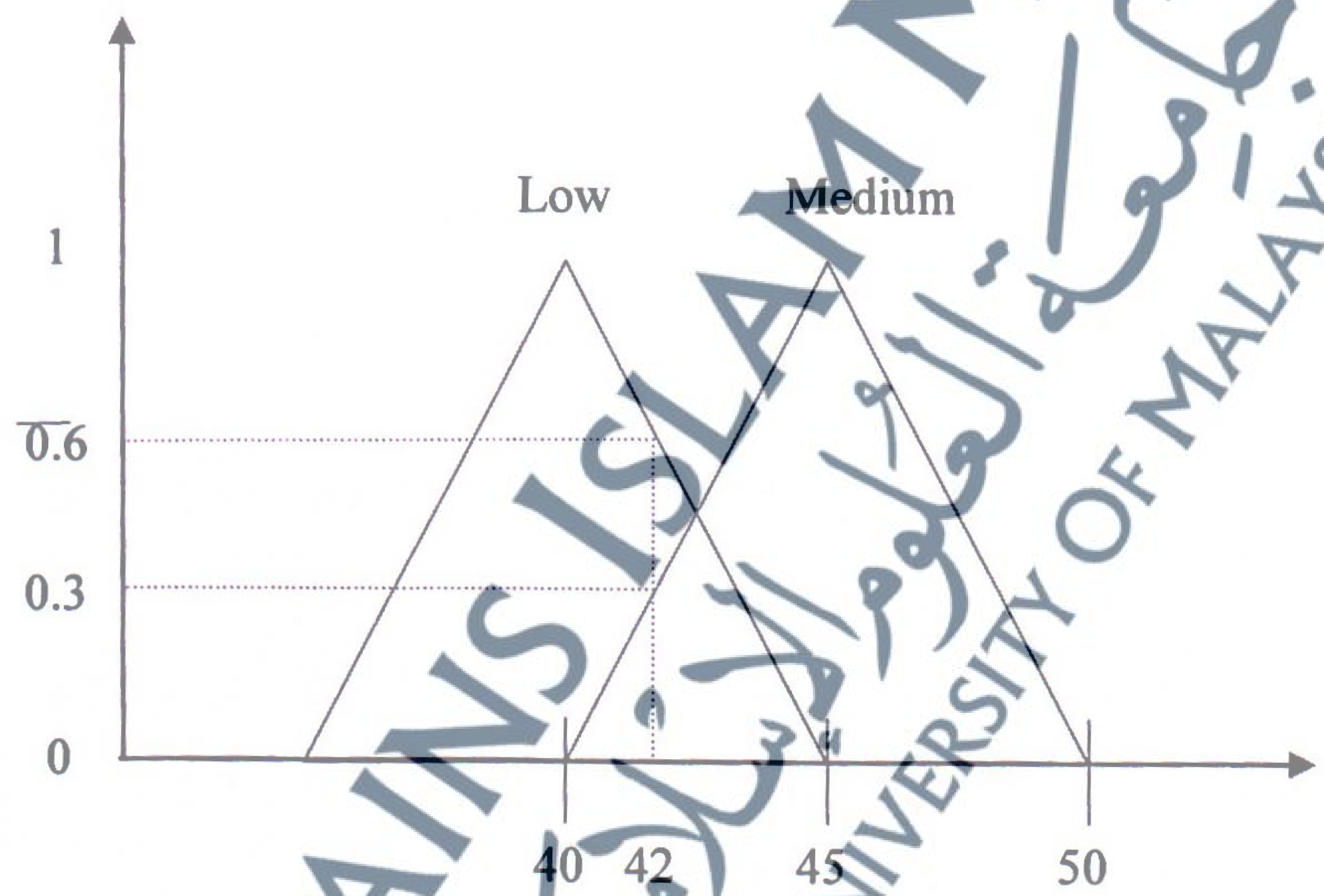
Utilizing fuzzy time series for traffic forecasting (Wen-ge & Yan, 2009) involves a rather different process from the one mentioned before. This is shown below (Figure 13). It begins by defining the boundary of the data in terms of the minimum and maximum values. Afterwards, the temporal domain is divided into a series of intervals that is customarily periodic.

Figure 13: Fuzzy Time Series Traffic Forecasting (Wen-ge & Yan, 2009)



Afterwards, the fuzzification process is enforced of which the variables with crisp values are transformed into fuzzy sets (Fouldsa et al., 2013). The transformation associates a value to a certain fuzzy set with a probabilistic determinant. To demonstrate the idea, the crisp value of 42 in the illustration (Figure 14) can belong to two fuzzy sets. It is connected to the first fuzzy set of "low" by the probability of 0.6 and simultaneously, to the fuzzy set of medium by the probability of 0.3.

Figure 14: Sample Fuzzification (Fouldsa et al., 2013).



The fuzzification process transpires along with the construction process. It entails the building of a structure that enumerates the possible fuzzy relationships (Hussain, 2010) between the variables in the model. Once the relationships are properly established, the forecasting can be performed. The performance of fuzzy time series in terms of vehicle traffic forecasting is found to be comparatively superior to neural network (Wen-ge & Yan, 2009). However, this should not be taken as a conclusive assertion to the general effort of forecasting since transportation traffic (Yufei & Moutarde, 2013) and network traffic are not entirely similar.

Practically speaking, a rather important characteristic of fuzzy relations is how transitional logic (Tsaur, 2012) is conducted. It is not based on the common matrix multiplication but instead, the max - min composition. This ensures that the logic transpired within the approach would not be erroneous (Guo et al., 2008; Min & Guo, 2010), which is rather crucial to ensure the soundness of the outcome. Observe the demonstrative case (Ross, 1995) that is depicted below (Figure 15).

Figure 15: Illustration of Multiplication for Fuzzy Relationship (Ross, 1995)

A = {a1, a2, a3}   B = {b1, b2, b3, b4}   C = {c1, c2, c3}

R	b1	b2	b3	b4
a1	0.1	0.2	0.0	1.0
a2	0.3	0.3	0.0	0.2
a3	0.8	0.9	1.0	0.4

S	c1	c2	c3
b1	0.9	0.0	0.3
b2	0.2	1.0	0.8
b3	0.8	0.0	0.7
b4	0.4	0.2	0.3

Min				
	0.1	0.2	0.0	1.0
	0.9	0.2	0.8	0.4
	0.1	0.2	0.0	0.4

Max	0.4

R•S	c1	c2	c3
a1	0.4	0.2	0.3
a2	0.3	0.3	0.3
a3	0.8	0.9	0.8

R <= A x B   S <= B x C

One of the greatest advantages of employing fuzzy time series in forecasting is the flexibility of dealing with non-stationary data (Wong et al., 2010). Non-stationary data (Strijbosch et al., 2011) is the type of data with distribution that changes through a temporal episode. This implies that the mean and variance can also suffer discrepancy on a dynamic basis, which makes it inherently unpredictable and therefore, difficult to forecast.

Despite the advantages of fuzzy time series, it does have a serious drawback when it comes to justifying the rationale of defining the fuzzy sets, relationships, rules etc. The ones employed to construct fuzzy relationships are normally designed without a solid theoretical basis (Duru & Yoshida, 2011). For instance, observe the artifact (Wen-ge & Yan, 2009).

With regard to this issue, fuzzy time series is inevitably coupled with a panoply of different technological approaches such as genetic algorithm (Leu & Chiu, 2011), neural network (Leite et al., 2012 ; Rahimi et al., 2012) clustering (M¨oller-Levet et al., 2003), and even ARMA (Egrioglu et al., 2013). They assist in the effort of justifying a particular recommendation that dominates the decision making process of design within the fuzzy time series.