2014

PREDICTING AND CLASSIFYING PACKET TRANSMISSION EFFICIENCY IN BIO-INSPIRED WIRELESS SENSOR NETWORKS

Ahmad Alkazzaz
Virginia Commonwealth University

PREDICTING AND CLASSIFYING PACKET TRANSMISSION EFFICIENCY IN BIO-INSPIRED WIRELESS SENSOR NETWORKS

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at Virginia Commonwealth University.

By

AHMAD ZUHAIR ALKAZZAZ

DIRECTOR: ROSALYN HOBSON HARGRAVES
ASSOCIATE PROFESSOR, ELECTRICAL AND COMPUTER ENGINEERING

Virginia Commonwealth University
Richmond, Virginia
August 2014

Acknowledgement

This thesis was completed with the assistance of a number of faculty, peers, family, and friends. First, I would like to thank Dr. Rosalyn Hobson Hargraves, my thesis advisor, for her support and guidance. Additionally, I would like to thank Dr. Ruixin Niu as well as Dr. Preetam Ghosh for laying a foundation for me to produce quality work.

Studying with such a great group of students supported my choice to remain at the School of Engineering which was one of the most memorable events of my life. Finally, I would like to thank my parents and uncle for their understanding and support.

Table of Contents

List of Tables

List of Figures

Abstract

PREDICTING AND CLASSIFYING PACKET TRANSMISSION EFFICIENCY IN BIO-INSPIRED WIRELESS SENSOR NETWORKS

By

AHMAD ZUHAIR ALKAZZAZ, M.S.

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at Virginia Commonwealth University.

Virginia Commonwealth University, 2014

Director: ROSALYN HOBSON HARGRAVES
Associate Professor, Electrical and Computer Engineering

Biological networks (specifically genetic regulatory networks) are known to be robust to various external perturbations. Bio-inspired wireless sensor networks (WSN) are known to be smart communication structures and have a have high packet transmission efficiency. In earlier work neural network models that correlate the average packet receival rates to the five topological features of the bio-inspired WSN were investigated. These features include the degree index, sink coverage, network density, hub node density, and motif index. In this thesis, an appropriate classification algorithm that works with these five features is investigated. The random forest algorithm is the best classification algorithm compared to other classification methods (APPENDIX B). In addition, a local weighted linear regression algorithm was created to predict the robustness of the network utilizing these five topological features.

Introduction

1.1 Problem statement

As many technological advances are made in the monitoring, collection, and transmission of data, it is crucial to ensure that the data is of the highest integrity. One such technology that has allowed for the successful collection of environmental data using a variety of types of sensors is a sensor mote. A sensor mote is a collection of tiny devices that includes sensors capable of capturing physical data and a transceiver that sends and receives wireless signals to and from other sensor motes. Essentially these sensors are arranged in specific terrains, together forming Wireless Sensor Networks (WSNs) that continuously monitor physical changes such as temperature, humidity, sunlight, wind speed, etc. In military applications, large scale WSNs are deployed to alert the military base of any distant foreign intrusions. Another example is the use of WSNs in the SCADA system for power plants to achieve the real time logging that would allow warnings to be given to the relevant personnel (e.g. an SMS warning message to the supervisor) when a failure occurs in the plant and also allow corrective action to be taken before the performance is severely degraded (Govt, et al. 2011). However, with the increased deployment of WSNs structural issues such as node failures and channel noise are harder to detect in a timely manner which can adversely impact the system for which they have been deployed.

Much work has been done to minimize transmission issues such as multi-path interference, channel inhomogeneity (Savarese, et al., 2002) node failures and congestion (Li, et al., 2007). However, more work has to be invested in advances that would insure minimum loss of packets resulting from end to end delays and packet multi-hop as a product of structural discrepancies. Previous works (Ghosh, et al., 2011, Kamapantula, et al., 2012) showed that particular natural

graphs operating as smart routing topologies in bio-inspired WSNs, demonstrate more efficient packet transmission rates than that of randomly deployed sensor nets.

1.2 WSN versus Gene Regulatory Network (GRN)

Biological networks specifically genetic regulatory networks are known to resist external perturbations, and have inspired the design of WSNs to maintain packet transmission efficiency. Similarities between GRNs and WSNs can be explained through the biological function of transcription, where genes process signals from neighboring nodes in the form of transcription factors and forward them to downstream nodes of a GRN. The process is similar in WSNs where sensors receive packets from neighbors based on user defined routing protocols with packet forwarding instructions to other destination points (sinks).

1.3 Purpose

There are two major objectives for this master's thesis:

1) Investigate classification methods that will classify the robustness of bio inspired wireless sensor networks.

2) Create an algorithm that can predict the average packet percentage of data received using selected topological features of a bio inspired wireless sensor network. Examples of topological features include: degree index, network density, motif index, sink coverage and hub node density. These topological features will be formally defined in Section 2.2.

1.5 Contents of the Thesis

The remainder of this thesis is organized as follows. Chapter two gives a summary of the literature reviewed of similar research and the previous studies conducted on the problem. Chapter three provides the theory supporting the Random Forest classification algorithm and the locally weighted linear regression algorithm. Chapter four explains the methodology employed during the study. Chapter five describes the result received from the study. Chapter six summarizes the conclusions and recommendations based on the results.

CHAPTER 2 LITERATURE REVIEW and PRELIMINARY WORK

2.1 Overview

GRNs exhibit a type of biological robustness as defined by Kitano (2004a) where 'robustness is a property that allows a system to maintain its functions against internal and external perturbations.' Several researchers have demonstrated the robustness of GRN's. For example, Kitano, et al. (2007) demonstrate the GRN's capability to maintain genetic signaling in the face of internal and external cell distresses. Eum, et al. (2007) have shown that there are several optimized GRN's inspired topologies that are not affected by post link failures, nodes failures or link congestion. Kamapantula, et al. (2012) demonstrate that GRN derived sensor networks can out-perform those of randomly-generated ones with respect to packet-loss rates, but will experience longer transmission delays. Due to the existence of different disruption scenarios, researchers have yet to announce a unified measure for robustness. For example, Feyessa, et al. (2011) suggest that the network efficiency, an inverse function of the magnitude of the average shortest path P Crucittia, et al. (2004), should be analyzed for single node deletions. Other work includes the assessment of connection failures (Cohen, et al., 2000) and fractional inactivation Agoston, et al. (2005). However, in this thesis, it is hypothesized that connectivity measures are not sufficient for describing the effects of disruptions, as they do not consider the network's capability to deliver its primary function (i.e. communication). Therefore, attention is given to understanding the topological features that affect the networks' transmission efficiency, i.e. their ability to deliver packets to their final destination nodes (sinks).

Many studies show that classification theory helps in the design and evaluation of WSN in many aspects, such as increasing the network life time, reducing false detection and solving deployment problems. El-Aaasser, et al. (2013) show that WSN's can be classified based upon their energy

saving approaches: 1) traffic energy based approaches, 2) topology control based approaches, or 3) reserved base approaches. Wang, et al. (2007) prove that dynamic collaborative support vector machines (SVM) have outstanding performance in reducing time delay and improving the energy efficiency of WSN. Elbhiri, et al. (2013) show that using new spectral classifications increase the life time of whole network and save WSN energy. Classification methods also reduce false detection rate in a study by Dai, et al. (2012) by using multi-variate classification and increase accurate recovery action by using Hidden Markov Model (HMMs) methods as noted by Warriach, et al. (2012). Deif, et al. (2014) show that classification methods can be used for modeling and solving the deployment problem in WSNs. In the preliminary study, it is shown that there are five properties which contributed to the robustness of wireless sensor networks by using a random forest classification method and locally weighted linear regression. The five properties are: degree index, network density, motif index, sink coverage and hub node density. This thesis continues this work in that it uses these five topological features to classify the robustness of a bio inspired WSN and create an algorithm that can predict the WSN average packet percentage of data received using these selected topological features. These features are described in depth in the next section.

2.2 WSN Features

All sample features have been determined computationally, and their concepts are described in the following subsections. The equations that characterize the functioning of five features will be discussed. Chart and figures are utilized for clarification.

A. The Degree Index

Degree Index is assigned as one of features as it combines two characteristics of relative importance to the percentage of packets received in the NS-2 simulations (SLC): (1) the degree of the sink node, and (2) the relative degrees of every other node. Hence, to calculate the Degree

Index, the ratio of the Average Nodal Degree Eq. (1) to the highest degree Eq. (2) is considered as follows:

$$K_{avg} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}, \text{ and} \quad (1)$$

$$K_{max} = Max[\sum_{m=1}^{n} A_{m1} \sum_{m=1}^{n} A_{m2}, \ldots, \sum_{m=1}^{n} A_{mn}]. \quad (2)$$

Where n is the number of nodes in the network and A is the adjacency matrix of the networks, for which $A_{ij} = 1$ for a link between nodes i and j, and $A_{ij} = 0$ otherwise.

In Figure 1, an experiment performed over 5 networks of different sizes that were simulated for packet transmission rates under varying single sink schemes was presented. As shown in the figure, the transmission rates depend heavily on the degree of the nodes selected as sinks. Though the transmission rates do not monotonically increase with respect to the degree of the sink nodes selected (as shown in networks 2 and 5), it is always the case that the highest degree node gives the best results. Based on these observations, the Degree Index is considered a suitable metric for predicting SLC. The Degree Index is denoted as:

$$\text{DI} = \frac{K_{avg}}{K_{max}}, \quad (3)$$

which can depict the tightness or looseness of the network. As DI $\rightarrow$1, it can be deduced that the network's nodes gain relative closeness to the sinks degree, from which it can be inferred that a network is tightly connected. In cases of tightly connected networks, nodes gain less significance among themselves in terms of being selected as sinks. Figure 2 shows the results of another experiment conducted on 50 networks of varying sizes and DI. Surprisingly, as DI increases, the performance of the networks decreases, which makes the benefits of the sink node selection (having highest degree) redundant.

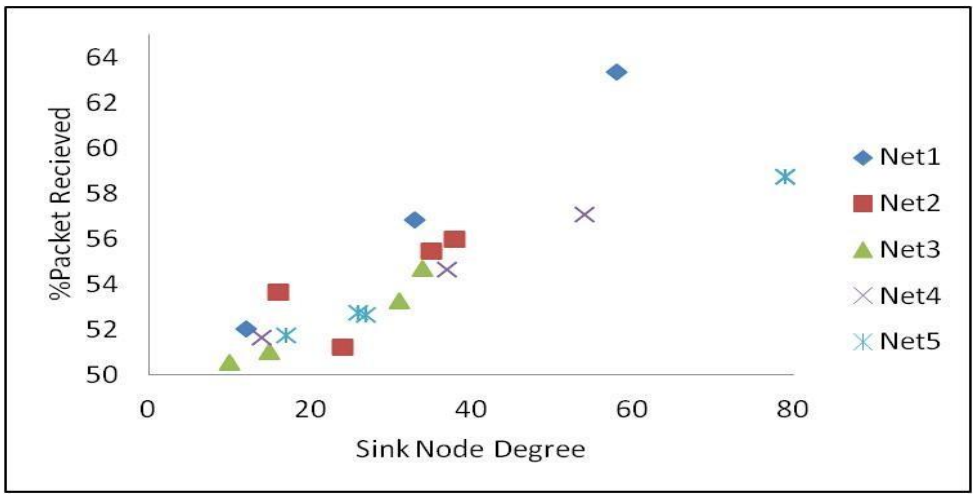

Figure 1: Scatter plot of percentage packets received vs. the degrees of the sink nodes selected for networks 1-5. Abdelzaher, et al. (2012)

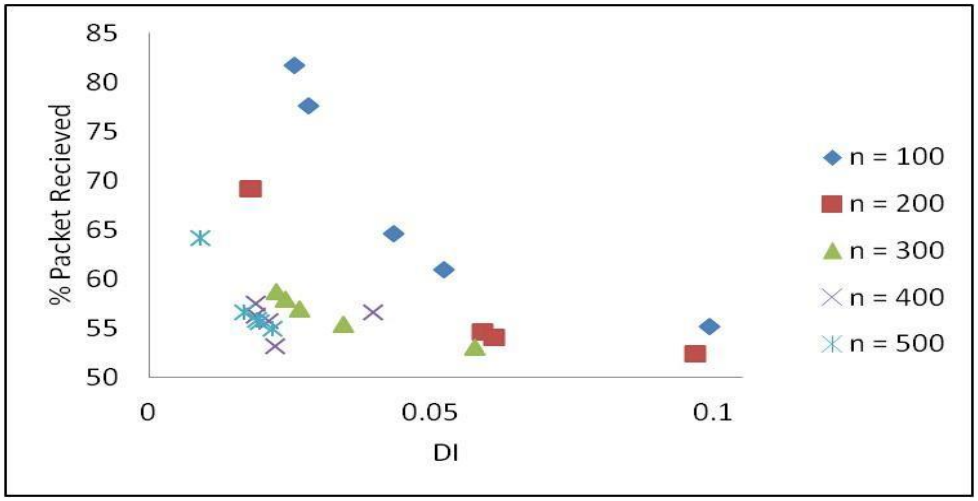

Figure 2: Scatter plot of percentage packets received vs. the Degree Index for different networks of sizes 100-500. Abdelzaher, et al. (2012)

B. The Network Density

The Network Density is traditionally a measure of the territorial occupation of a communication network, calculated as the ratio of the sum of the edge lengths to the surface area occupied by the network grid Beauguitte, et al. (2011). Since the simulations do not account for edge weights, the nodes are simplistically considered to be equidistant, having unit lengths of one.

The measure accounts for how many links occupy the adjacency matrix (A) grid and determine the Network Density as follows:

$$ND = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{i=1}^{n} A_{ij}, \quad (4)$$

where ND ranges between (n-1)/n^2 for a Star configuration (with a single hub, provided every other node is solely connected to the hub with one edge), and 1- (1/n) for a fully connected sub-graph excluding the self-loops. Table 1 shows a series of simulation results on 4 different sets of 10"density controlled" WSNs, in order to show the effects of ND on SLC. Density controlled networks are generated using the methods of switches Milo, et al. (2002) - a method of switching edges between nodes, thereby preserving the nodes in/out degrees but altering the networks final orientation. This way every set will have 10 networks having same ND and nodal degrees, but different overall network structures. Note that the performance of the networks having the same ND are comparable because the properties of the sink nodes selected after the randomizations are still preserved in the sub graphs. Results of this experiment show that there is no direct correlation between the network density and the performances of the WSNs.

Networks	Density ($* 10^{-3}$)	%Packets Received
1-10	5.4	33.33
11-20	17.9	50.00
21-30	19	52.17
31-40	25	50.00

Table 1: Average packet receipt rates for networks having different densities.

C. The Motif Index

Before dwelling into the definition of the Motif Index, it must be acknowledged that these repetitive "motif" substructures have significant contributions to WSN performance and functionality as is shown earlier in a study by Kamapantula, et al, (2012) and separately at (Hovareshti, et al, 2011), as well as affecting robustness in biological networks (Kitano.et al, 2004; Kitano, et al, 2007). Although various types of motifs have been identified previously in biological networks, the "most significant" motifs considered for this model are the Feed-Forward Loop (FFL) and the Bi-fan (BF) (Milo, et al, 2002). These two motifs significantly outnumber similar sub-structures when mined from the GRN of E. coli in comparison to other randomized networks and hence are believed to have significance in biological networks in general. Furthermore, FFLs are notable for their ability to deliver vital functions such as delay response times in genes, irreversible speed up, or create pulses (Mangan, et al, 2003). Similarly, BFs are the building blocks of dense overlapping regions, which are considered to be the backbone for GRNs, sharing global functions such as: stress response, nutrient metabolism, or bio-synthesis of key classes of cellular components (Alon, et al, 2006). Figure 3 shows the FFL and BF structures mentioned above, for which it is hypothesized that their relative abundance in the network should make an important feature to consider in our regression model. In our data sets, it was observed that the FFL counts accompany larger BF counts for any bio-inspired network considered. In order to account for both, their counts are converted into a normalized ratio of one motif abundance to the other, which also reduces the features' dimensionality by one parameter. Since directions in the simulated networks are ignored, the same conversion is applied to the Motif Index and the occurrences of quadrilaterals and triangles are considered (corresponding to BFs and FFLs respectively) in the networks as shown in Figure 4(a).

To calculate the motif counts, every pattern is considered:

NT = (< i; j > \ < j; k > \ < i; k >) 2 Rc and

NQ = (< i; j > \ < i; l > \ < k; j > \ < k; l >) 2 Rc

for every i; j; k and l < n, for the Triangle and Quadrilateral structures respectively. The hypothesis is motivated by the fact that such patterns are ideal for considering cluster formations based on the number of nodes that participate in forming them and the ones that do not Fagiolo, et al. (2007) and Barmpoutis, et al.(2010). For an undirected non-weighted network stored in an adjacency matrix A, these counts can be determined mathematically as;

$$N_T = \frac{1}{6}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k}^{n}[A_{ij} \cap A_{ik} \cap A_{jk}] \quad (5)$$

$$N_Q = \frac{1}{8}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{i}^{n}[A_{ij} \cap A_{il} \cap A_{jk} \cap A_{kl}] \quad (6).$$

An illustration of the motif patterns in the adjacency matrix is given in Figure 4(b). Note that in Eq. (5), the occurrence of triangles is divided by 6 to avoid redundancy caused by the symmetry of the triangle pattern, similarly with Eq. (6) the occurrence of quadrilaterals is divided by 8. Hence, the Motif Index is calculated as:

$$MI = \frac{N_Q}{N_T + N_Q}, \quad (7)$$

which will account for effects of the motif ratios to the packet transmission efficiency of the networks considered.

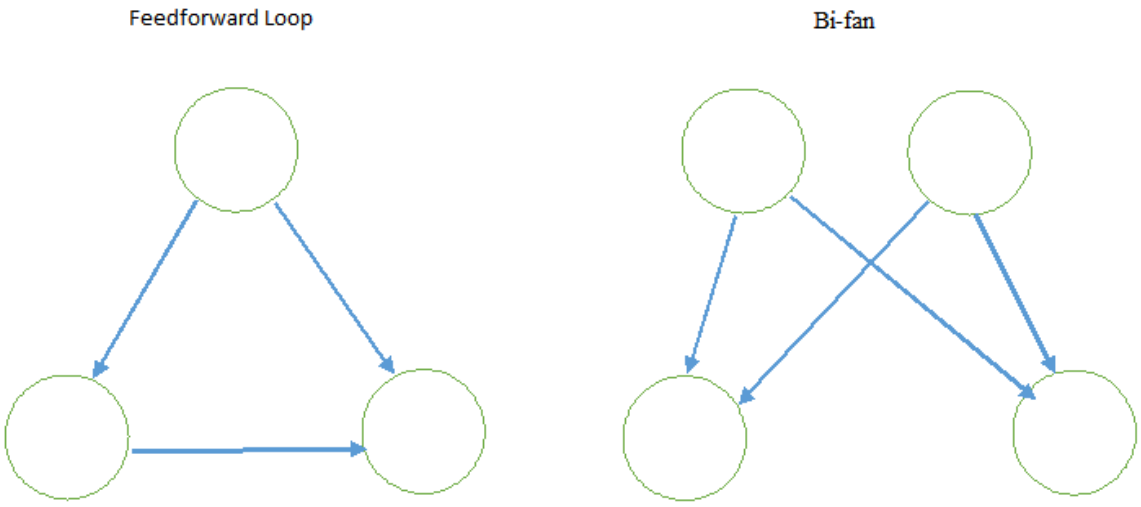

Figure 3: The FFL and BF motif structures.

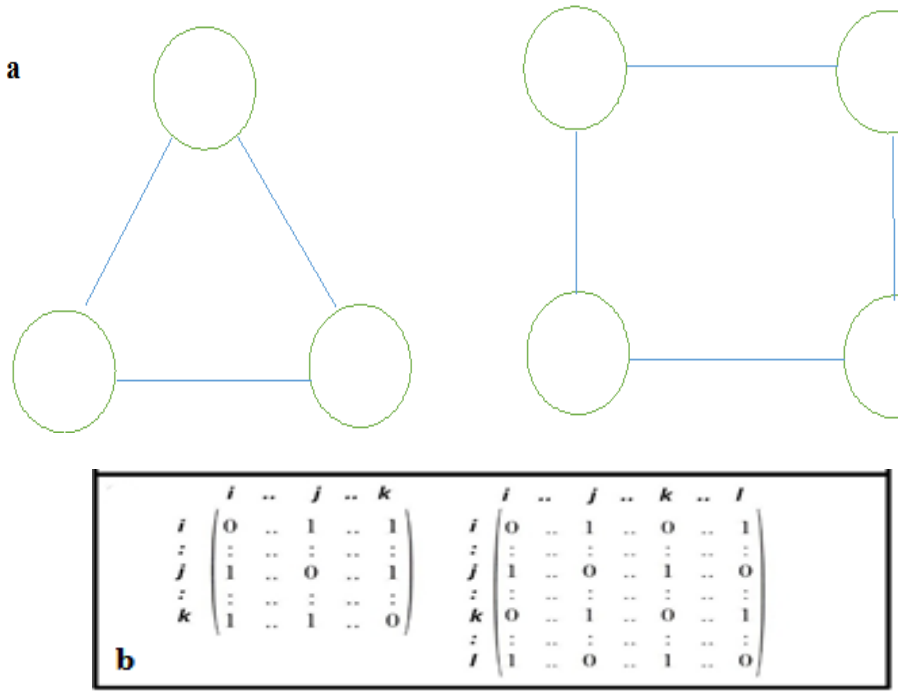

Figure 4: (a) Actual (undirected) structures considered for the Motif Index. (b) The adjacency matrix of the structures considered for the Motif Index

D. The Sink Coverage

The sink coverage measures the percentage of nodes that have a direct link to the sink node, using Kmax of Eq. (2),

$$SC=\frac{K_{max}}{n} \qquad (8)$$

When node a tries to send packets to node b through node c lying along the path dab, packets are queued at c before they get forwarded to b, which in return can be dropped if packets exceed the queue length at c. However if c did not exist in the dab path, packets will not be discarded due to multi-hops;

SC is a feature that captures such scenarios.

E. The Hub Nodes Density

The density of the hub nodes measures the territorial occupation of the adjacency matrix grid by the higher degree nodes as a fraction of the total number of edges. It is hypothesized that the hub nodes are the hot spot traffic management zones as they have more packets hopping through them. This quantity can be determined as follows:

$$HDN=\frac{1}{l_{total}}\sum_{i}^{nh}[\,2\sum_{j=gi}^{ng}A_{ij}+\sum_{j=hi}^{nh}A_{ij}], \qquad (9)$$

where *nh* is the number of hub nodes, *gi* is the index of nodes outside the set of hub nodes, *hi* is the index of the hub nodes and l_{total} is the total number of edges in the network.

2.3 Preliminary work

2.3.1 Overview

In the preliminary work different neural network models were proposed to correlate the average packet received rates to the five topological features of the bio inspired WSN described in section 2.2: degree index, network density motif index, sink coverage and hub density. In the following sections that work is discussed.

2.3.2 Neural Networks

The essence of neural network modeling emerges from the fact that any function y can be approximated using a set of weights w, and a set of features X that are related to the data using the famous formula,

$$y = F(wX + b), \qquad (10)$$

where b, the bias, represents a constant translation to the curve or the plane, and F(·) represents an nonlinear activation function. Given set of networks (data) with quantifiable network features and quantifiable performances depending on those features, a neural network can predict these performances. The process known as training is achieved by learning from the features that correspond to particular performances, for which the model tries to adjust the weights and bias to produce an approximation. Every training iteration shift from data point i to j accompanies adjustments of the weights,

$$\_\Delta w_{ij}(p + 1) = \eta(e_j y_i) + \Delta w_{ij}(p) \qquad (11)$$

Such that the new weight value for pattern p + 1 is dependent on the weight change associated with pattern p. The termination criteria for Eq. (11) depends on the error e, and the learning rate ŋ.

Many artificial neural network algorithms have been considered for predicting and enhancing the performance and speed of networks such as the Internet (Cortez, et al, 2006; Nelson, et al, 2008). While many have succeeded in predicting the Internet's traffic using neural networks enhanced by genetic algorithms (Wang,, et al, 2008), fundamental multi-layer perceptron (MLP) and radial basis functions (RBF) (Rutka., et al, 2006), others succeeded in classifying the traffic of sub-networks of the Internet using basic data of packet size, inter-arrival time and classifying the traffic over a time frame (Trivedi, et al, 2004). The famous Grey NN model (Wang, et al, 2009) integrates the strengths of multiple neural network concepts into one single neural network architecture, is known to be the most accurate in terms of predicting network traffic flow. The network flow assigns a performance value for traffic emerging from one destination of the network to another or many others, which does not describe the networks performance in a congestion scenario. Moreover, the solutions above consider particular routes to be taken for optimizing flow in a particular direction, and do not consider the effects of the topologies on rerouting the flow in other directions. Hence, the networks' performance is assumed to reflect its ability to continue flow in worst case scenarios by targeting one hot spot of the network as a destination point (or sink) and every other spot as having traffic emerging from it (i.e. packet source nodes). In WSNs, the mentioned routing scheme is known as the "flooding protocol" and it can be used to determine the networks performance using the following equation:

$$\text{Per}f: = \frac{\#Packets\ received\ by\ sink}{\#\text{packets flooded in the network}} * 100. \quad (12)$$

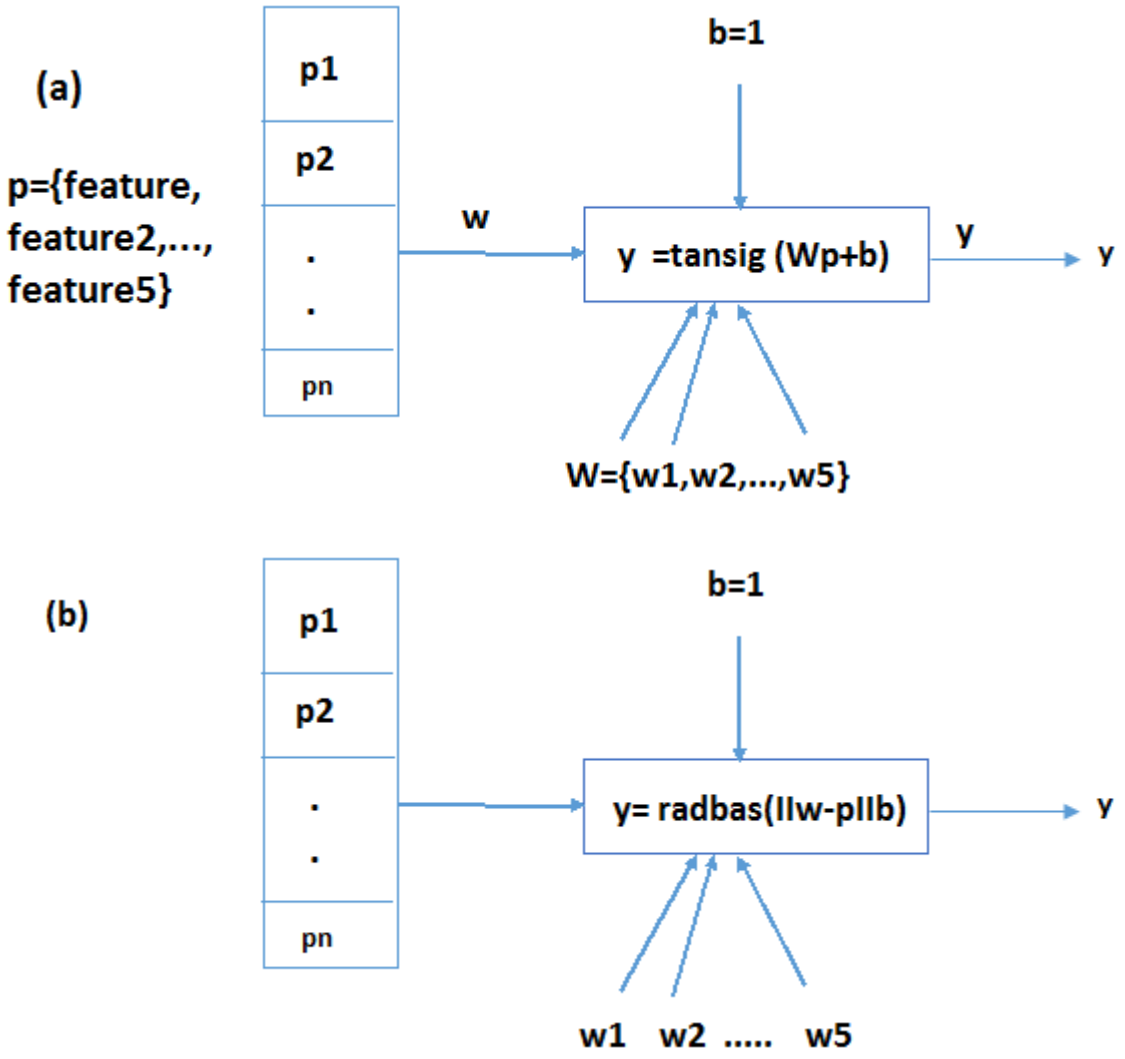

Fig.5.a Simplified representation of the Multi-layer Perceptron with a single hidden layer

Fig.5b Simplified representation of the Radial Basis Function model with a single hidden neuron.

The basic architectures of the multi-layer perceptron and RBF models are depicted in Figure 5. In the preliminary study, several MLP architectures, varying in the number hidden layer neurons are considered. Additionally, different RBF architectures are considered. In addition, General Regression Neural Networks (GRNN) and Probabilistic Neural Networks (PNN) are investigated

GRNNs, implemented in the MATLAB software package, have the advantage of combining the strengths of MLPs and RBFs in one structure, as it has two transformation layers: a radbas (a radial basis layer) followed by the tansig (a hyperbolic tangent layer) of figures 5 (a)(b) - is known to perform better predictions than Multi-layer Perceptron with different layers(MLP-1,MLP-2,MLP-3) and RBFs (Horng, et al, 2012). PNNs have the exact same structure of RBFs followed by a competitive layer which assigns 1 to the pattern which is closest to the target and 0 otherwise. In this study PNN is used for classification and all other neural networks are used for regression analysis. After different neural network models were applied to correlate average packet receipt rate to the five topological features, the result as presented in table 2. The results for the classification algorithm are presented in table 3. The data was partitioned into two classes, three classes, and four classes to see which classification worked best. As noted in table two the PNN worked best when the data was only partitioned into two classes. The resulting two classes were:

Method	Layers/neurons	Goal	Spread	Data error (%)
MLP-1	5	-----	------	3.0933
MLP-2	10 - 2		------	2.3899
MLP-3	7 – 6 - 4		------	2.4537
RBF	47	2.2	2.25	2.5232
GRNN	----	---	0.3	0.92164

Table 2: Different neural network model to correlate average packet receipt rate to five topological features Where MLP-1 is a mutli-layer perceptron network with 1 hidden layer of 5 neurons, MLP-2 has two hidden layers, the first with 10 neurons and the second with 2 neurons, and MLP-3 has three hidden layers, 7 , 6, and 4 neurons respectively.

Number of classes	Percentage of data error in class1	Percentage of data error in class2	Percentage of data error in class3	Percentage of data error in class4	Probabilistic Neural network percentage error
Two classification	1.6978	1.6978	----	------	1.6978
Three classification	2.5467	4.9236	2.7165	--------	3.3956
Four Classification	5.4329	10.6961	11.035	4.5840	7.9372

Table 3: Different neural network models to correlate average packet receipt rate to five topological features with classification

CHAPTER 3 THEORY

3.1 Random Forest Algorithm

3.1.1 Overview of Random Forest

Combining classifiers is the favored focus in research on improving classification accuracy since traditional machine learning algorithms have a tendency towards low accuracy. Amit, et al (1997), researched the use of random selection to search for the best split at each node among a large amount of geometric figures. During 1996 and 1998, Dietterich, et al, (1998) advanced the Bagging algorithm, an early stage algorithm, and proposed the random split selection theory, respectively. Dietterich's theory stated that "at each node the split is randomly selected from the N best splits." The "random subspace" that Dietterich theorizes about is one that Ho also studied. His theory states that "each node that is split is randomly selected form the N best splits." In addition to studying the random subspace, Ho, et al (1998) has also studied the methodology behind the random subspace. The method offers that each tree grows by a random selection of a "subset of features." While Ho utilized Dietterich's work, Breiman utilized the ideas of Amit, et al, (1997), whose output in the original training set led to his creation of new training sets by "randomizing the outputs in the original training set."

A combination machine learning algorithm is a random forest. Random forests, or RFs, are determined by combining with a series of tree classifiers, giving each tree a unit vote for the most popular class, and then combining those results to get the final sort result. RF is one of the most popular and reliable research methods for gathering data. High classification accuracy, the toleration of outliers and noise, and lack of over-fitting characterize RF.

3.1.2 Principle of Operation

A random forest is defined as a "collection of tree-structure classifiers" denoted by the equation $\{h(x, \Theta_k), k=1...\}$. As mentioned above, each tree casts a vote for the most popular class. Within the equation, the$\{\Theta_k\}$ are "independent identically distributed random vectors" and the votes are identified at input x. A training sample set and a random variable anchor the planting of a tree in Breiman's RF model. The random variable is equivalent to the kth tree and is identified as Θ_k. Elements between these two random variables are identified as a classifier $h(x, \Theta_k)$. Again, x is the input vector. Running the equation k times gives a classifier sequence of $\{h_1(x), h_2(x) \ldots h_k(x)\}$. That sequence is used to establish multiple classification model systems. Ultimately, ordinary majority drowns the system and the "decision function" is denoted as H(x) = avg $\max \sum_{i=1}^{k} I(h_i(x) = Y)$. In this equation, H(x) is a combination of the classification model. h_i is a single decision tree model, Y is the output variable, and I(.) is the indication function. Any input variable gives the decision tree the opportunity to vote for the best classification result.

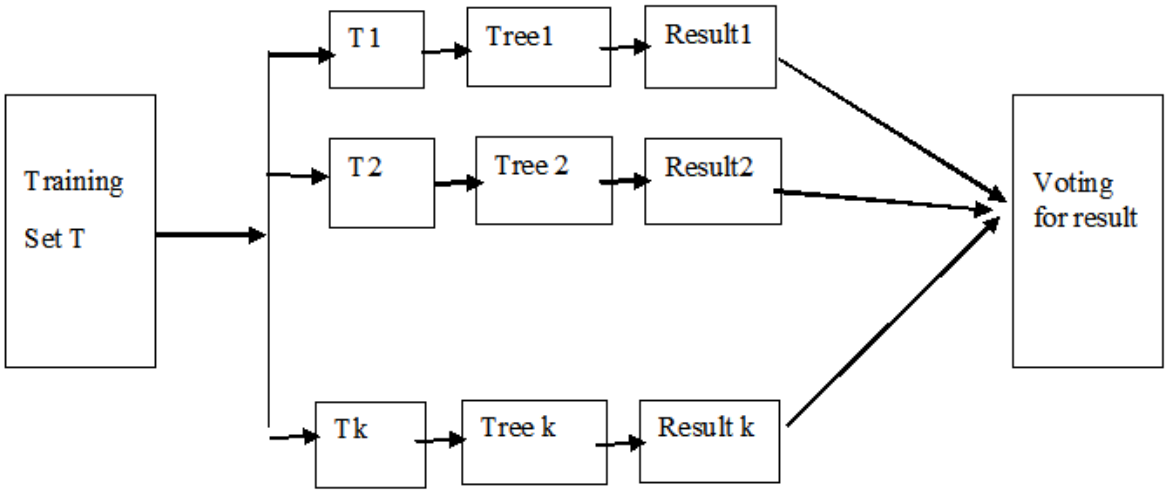

Figure 6: Random forest schematic

3.1.3 Out-of-Bag Estimation

Bagging methods impact the use of random feature selection and are used when the tree is begun on a new training set. That new training set is taken from the original via bagging methods. Whether to enhance accuracy when using random features, or to bring out data that is utilized to give continuing estimates of the error of the classifier (PE) of RF, alongside estimates of strength and correlation, bagging methods work well with random feature selection. Out-of-bag (OOB) data, which uses the OOB estimation algorithm, is used to estimate the performance of classification:

> Given an original training set T with N samples, the kth training set is drawn from T with replacement by bagging, every Tk contains N samples. Then the probability of each sample cannot contain $(1 - 1/N)^N$, when N is large enough, $(1 - 1/N)^N$converges to e^{-1}. In other words, 36.8% samples of the T is not contained in Tk. (Liu, et al, 2011)

There is an OOB estimate for error for each tree and that estimate of generalization error of RF is equivalent to the average of estimations "of all tree error for every tree contained in the RF." Breiman, Tibshirani, Wolpert and Macready all proposed OOB data as being useful. Tibshirani, identified its use in estimates of generalization error while Breiman proved it to be accurate as using a test set of the same size as the training set. If one were to compare cross-validation and OOB data, the OOB estimate would be determined as unbiased and faster in its calculations. Since the OOB estimate is equivalent to the test set OOB removes the need for such a set and is good for strength and correlation estimates as well. Using OOB allows researchers to further determine classification accuracy and how to improve it.

3.2 Locally weighted linear regression algorithm

A locally weighted linear regression algorithm does two things: (1) Fit θ to minimize $\sum_i (y^i - \theta^t x^i)^2$ and (2) Output $\theta^T X$:

The $w^{(i)}$'s are non-negative valued weights. Intuitively, if $w^{(i)}$ is large for a particular value of i, then in picking θ, the value of $(y^i - \theta^t x^i)^2$ should be minimized. If $w^{(i)}$ is small, then the $(y^i - \theta^t x^i)^2$ error term will be pretty much ignored in the fit. A fairly standard choice for the weights is If x is vector-valued, this is generalized to be $w^{(i)}$ = exp(−$(x^{(i)}$−x)^T $(x^{(i)}$−x)/(2τ^2)), or $w^{(i)}$= exp(($x^{(i)}$ − x)^T $\sum^{-1}$($x^{(i)}$ − x)/2), for an appropriate choice of τ or∑.

$w^{(i)}$ = exp$\frac{x^{(i)}-\text{x})2}{2\tau^2}$ (13)

If |$x^{(i)}$ − x| is small, then $w^{(i)}$is close to 1; and if |$x^{(i)}$ − x| is large, then $w^{(i)}$ is small. This shows that weights are dependent on the point x where x is being evaluated. One chooses θ to give a higher weight to the (errors on) training examples. These are close to the query point x. The parameter τ determines how quickly the weight of a training example falls off with distance of it's $x^{(i)}$ from the query point x. τ is called the bandwidth parameter.

3.3 K-means Algorithm

One of the foundational learning algorithms that solves the clustering problem is K-means (MacQueen, 1967), which is a simple way to identify a data set through a particular number of clusters (assume k clusters) fixed a priori. The central idea is to define a singular k centroid for each cluster. Different locations cause different results, so the centroids should be situated in an

astute way. Therefore, placing the centroids as far away from each other as possible is the better decision and first step. Following the placement of the centroids, each point belonging to a given data set must be aligned with the nearest centroid. If no point is pending, there is no need to conduct this step; it is already completed. Early groupage is done, so the next step is to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. A new binding must be done between the same data set points and the nearest new centroid after we have their k new centroids. The loop is generated. We may find that the k centroids move step by step until there are no more changes as a result of the loop. Simply, the centroids do not move anymore. This algorithm aims at minimizing a squared error function, or *objective function*. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \quad (14)$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j, is an indicator of the distance of the *n* data points from their respective cluster centers.

CHAPTER 4 METHODOLOGY

4.1 WSN Data

WSN data was recorded at Virginia Commonwealth University computer science department. The data can be segmented into two deferent sets of values; the target values of the NS-2 simulations and the graph features of the GRNs inspired WSN. For training, 590 different GRNs of sizes $50 \leq n \leq 1477$were extracted from the bacterium GRN of E. coli using the Gene Net Weaver tool (Schaffter, et al, 2011). Similarly for testing, 118 deferent GRNs were considered. The target values were generated via NS-2 software and the features were determined computationally using Java or any other programming language.

4.2 Data Pre-Processing

In order to improve the performance of the classification algorithm outliers in the data set were identified. Determining the outliers or more specifically, determining upper and lower quartile values (IQR), is a fairly simple method and can also be used to reveal the interquartile range. The lower quartile value, or LQ, is the "value that 25 % of the data set is equal to or less than" while the upper quartile value, or UQ, is the "value that 75% of the data set is equal to or less than." From these numbers, one can identify suspect outliers as being 1.5*IQR greater than the upper quartile or 1.5*IQR less than the lower quartile. R software can be used as a free software environment for statistical computing and graphics because it gathers and computes information on a variety of UNIX platforms including Windows and MacOS. (Result in Appendix A**)**

4.3 Clustering method

The K-means method was efficiently used for clustering data. Two cluster centroids points were used for two classifications. Three cluster centroids points were used for three classifications. Finally, four cluster centroids points were used for four classifications. The standard deviation was used as a cut off points in each class. The results are shown in the table below.

	Four Classification	Three Classification	Two Classification
Class1 cluster centroids	52.974	53.1527	53.9523
Class1 standard deviation	+/-0.3369	+/-0.4409	+/-0.8532
Class1 number of network	83	109	239
Class2 cluster centroids	53.7231	56.0101	59.3984
Class2 standard deviation	+/-0.1519	+/-1.6734	+/-3.1468
Class2 number of network	26	266	216
Class3 cluster centroids	56.2506	62.9044	-
Class3 standard deviation	+/-1.9189	+/-2.0703	-
Class3 number of network	281	80	-
Class4 cluster centroids	63.4558	-	-
Class4 standard deviation	+/-1.9058	-	-
Class4 number of network	65	-	-
Random Forest accuracy	94.7253%	90.7692%	94.0659%

Table4: Clustering data by using K-means method and the Random Forest accuracy result

4.4 Classification Method

The random forest algorithm was used to classify the robustness of wireless sensor networks. The five features used to characterize the wireless sensor network are: degree index, network density, motif index, sink coverage and hub node density. The percentage of data received was

1. divided into 4 classes using the following rules:

- Class 1 - between 52.33629 and 53.49696% of data received.
- Class 2 - between 53.52004 and 53.99475%of data received.
- Class 3 - between 54.00187 and 60.96491% of data received.
- Class 4 - between 61.00629 and 67.2956% of data received.

2. divided into 3 classes using the following rules:

- Class 1 - between 52.33628922 and 53.99474869% of data received.
- Class 2 - between 54.00187441and 54.74349965%of data received.
- Class 3 - between 54.77375566and 67.20257235% of data received.

3. divided into 2 classes using the following rules:

- Class 1 - between 52.33628922 and 55.49102429% of data received.
- Class 2 - between 55.52884615and 67.29559748%of data received.

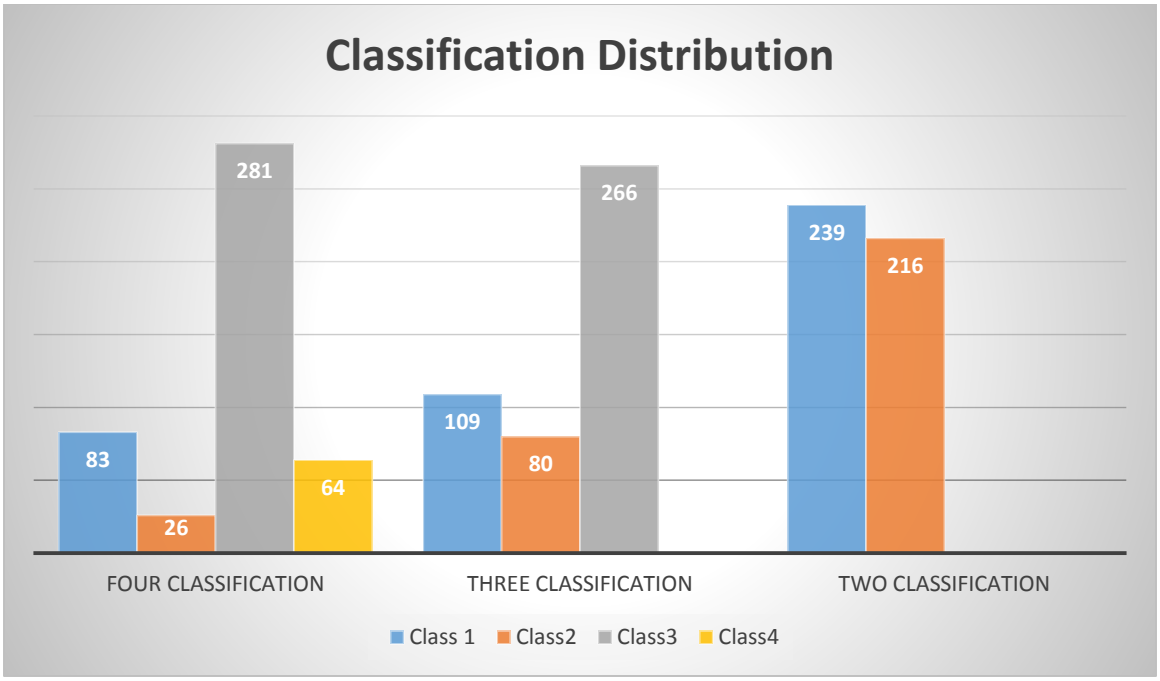

Figure 7: Number of instances for each class of the different classification system

4.5 Prediction Method

A weighted linear regression algorithm was used to predict the percentage of data received using MATLAB code. The first method used the five features described in section 2.2 as input to the weighted linear regression algorithm and produced an output (predicted) value for the percentage of packets received. The second method utilized a cascade architecture where first the data was subdivided into two classes and then the output from the classification algorithm was then sent to the weighted linear regression algorithm. Third the data was subdivided into three classes and then the output from the classification algorithm was then sent to the weighted linear regression algorithm. Finally the data was subdivided into four classes and then the output from the classification algorithm was then sent to the weighted linear regression algorithm.

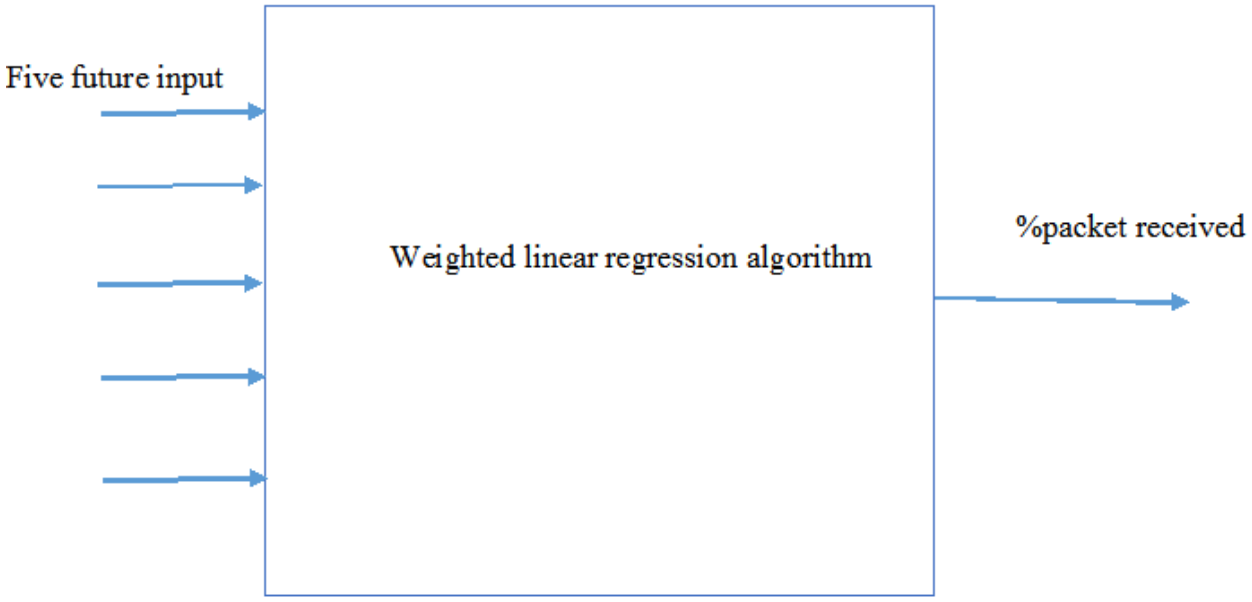

Figure: 8 System used to predict the percentage of data received without prior classification

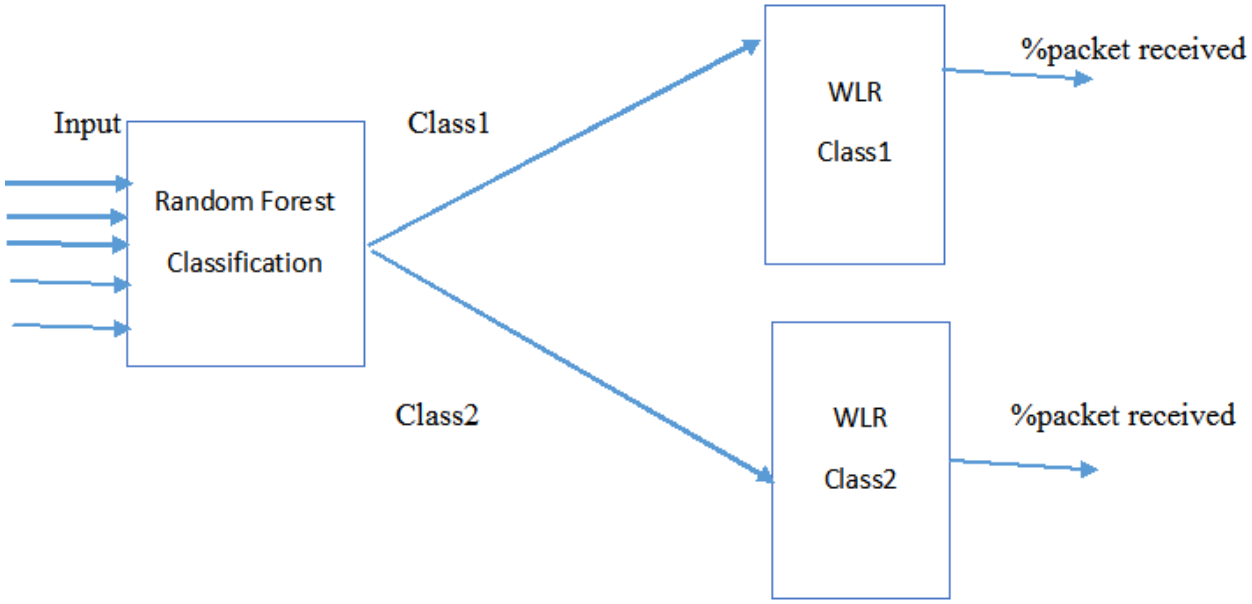

Figure: 9 System used to predict the percentage of data received with the data presorted into two classes

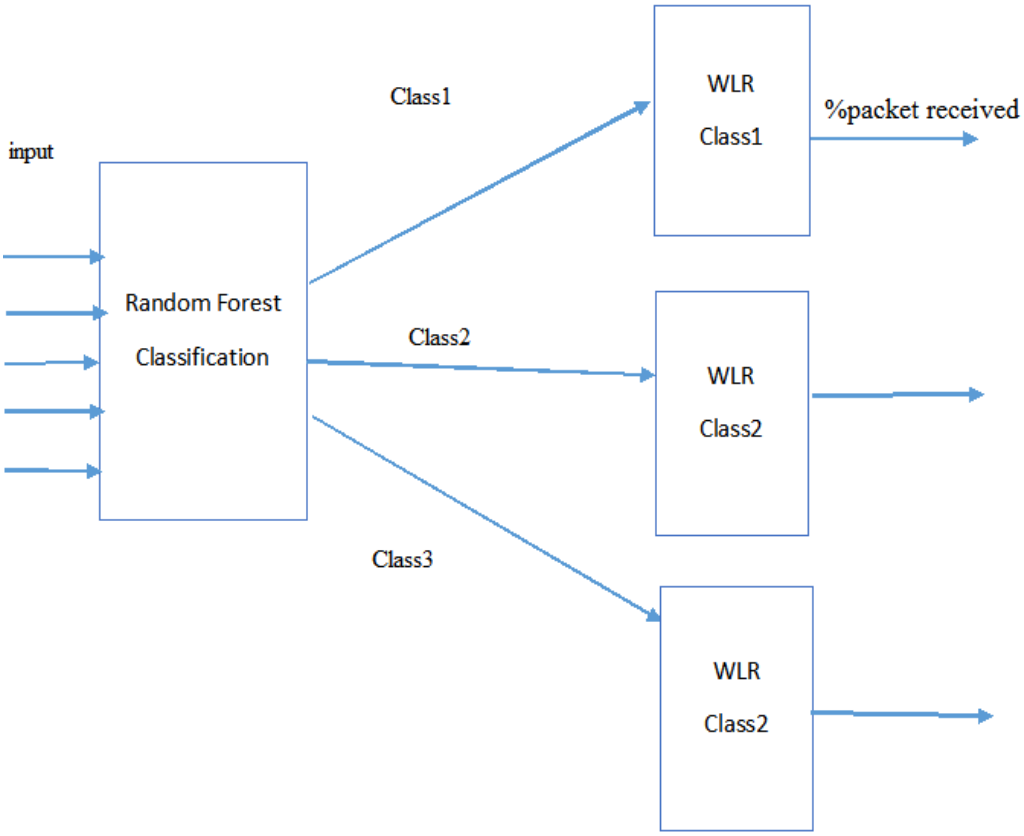

Figure: 10 System used to predict the percentage of data received with the data presorted into three classes

Chapter 5 Classification and Prediction result

5.1 Count Outlier

R software was used to determine the upper and the lower quartile value. The interquartile range (IQR) was found by subtracting the lower quartile value from the upper value. Points that were less than the lower cutoff point (QL-1.5*IQR) and greater than the upper cutoff point (QU+1.5*IQR) were eliminated. The results are shown in the table below.

	QUF1	QLF1	QUF2	QLF2	QUF3
Wclass	0.064399	0.016638	0.032355	0.005329	0.9841
Class1	0.01676	0.01375	0.005116	0.003521	0.9845
Class2	0.068472	0.017937	0.025687	0.005889	0.9861
Class3	0.066563	0.035705	0.0359	0.01547	0.9807
Class4	0.061765	0.030638	0.047755	0.025263	0.9731
	QLF3	QUF4	QLF4	QUF5	QLF5
Wclass	0.9051	0.65	0.2966	0.4494	0.2532
Class1	0.9826	0.2966	0.24	0.3602	0.2357
Class2	0.9123	0.4257	0.2975	0.422	0.24212
Class3	0.9196	0.636	0.4313	0.42699	0.29491
Class4	0.891	0.8718	0.68	0.5047	0.2842

Table 5: R software result to count the upper and the lower quartile value
Q=quartile value, U=upper, L=Lower, F1=The Degree Index, F2= The Network Density

F3= the Motif Index, F4= the Sink Coverage, F5= the Hub Nodes Density

, Wclass=without classification

5.2Random Forest Result

WEKA software is used to calculate the classification algorithm. The classification algorithm that worked best was with 83 instances in class 1, 26 in class 2, 281 in class 3, and 64

in class 4. The random forest classification was utilized on the five features. The random forest classification was applied on the 5 features for the training data the results are shown in Table 5

Correctly Classified Instance			431	94.73%		
Incorrectly Classified Instances			24	5.2747		
Kappa statistic			0.9042			
Mean absolute error			0.051			
Root mean squared error			0.1575			
Relative absolute error			18.0971	%		
Root relative aquared error			42.0153	%		
Total number of instance			455			
Detailed Accuarcy By Class						
	ROC Area	Class				
	0.988	1				
	0.731	2				
	0.979	3				
	0.846	4				
Weight AV	0.947					
	Confusion Matrix					
a	b	c	d	Classified as		
82	0	1	0	|a=1		
3	19	4	0	|b=2		
2	0	275	4	|c=3		
0	0	10	55	|d=4		

Table: 6 Shows the random forest algorithm result

The confusion matrix reveals a few distinct ideas about the classes: 275 instances or Class 3 were correctly classified; two instances were misclassified in class 1 and four instances were misclassified into class 4. Class 1 follows and holds 82 instances were classified correctly while only one instance was misclassified into class 3. Finally class 4 and class 2 have 10 and seven misclassification respectively. The conclusions that can be drawn from the figures laid out by the

confusion matrix, specifically table 5, include the indication that the accuracy of the model cannot be used for "assessing the usefulness of classification models built using unbalanced datasets." The predictive performance of the model is shown in the right hand classifier output frame. Within that output frame, the confusion model is presented at the bottom of the classifier output window. The ten-fold cross validation method is also stipulated as a default. The accuracy of the model is thus very high at 94.7253%. From this, one can conclude that the "Kappa statistic" is the better choice for a good result. In cases of now relation, it is valued at zero, while it gets closer to one as the relationship between the class label and attributes of instances gets stronger. Additionally, "ROC area" is a useful determinant of statistical characteristics; the value greater than 0.9 gestures at the strength of "statistical dependence."

Classification method	CCI	ROC	KS
trees.J48 Ross Quinlan (1993).	94.7253	0.945	0.9049
trees.J48graft Geoff .W(1999)	94.7352	0.944	0.9047
Random Forest	94.7253	0.984	0.9048
Meta Decorate Melville, et al.(2003)	94.2857	0.972	0.8962
meta.OCC Eibe et, al.(2001)	94.2857	0.947	0.897
meta.RF Juan, et, al.(2006)	94.0659	0.989	0.8928
meta.NDDNBND Lin(2005)	93.6264	0.948	0.8844
meta.END Eibe(2004)	93.4066	0.957	0.8811
meta.LB J.Friedman, et al.(1998)	93.4066	0.932	0.98

Table 7: The best ten results of different classifiers by using WEKA software

5.3Data Prediction Result

A Locally weighted liner regression algorithm was used for predicting percentage of data received. MATLAB code was used to run this theory. The result showed that the total percentage of the data received error without classification was 2.137457, for two classes was 1.269, for three classes was 1.025, for four classes was 1.121. The result is shown in the table below:

	class31	class32	class33	no class	class41	class42	class43	class44	class21	class22
% Error	0.196665	0.414582	2.465024	2.13946	0.2022	0.107472	2.09651	2.07745	2.22E-10	2.53834

Table 8: Error percentage by using locally weighted linear regression methods with and without classification. Column headings are as followsClass31=Three classes - class1, Class32=Three classes - class2, Class33=Three classes - class3, no class= predict the data without subdivided classes, Class41=four classes - class1, Class42=four classes - class2, Class43=four classes - class3, Class21=Two classes - class1, Class22=Two classes - class2

CHAPTER 6 CONCLUSION AND RECOMONDATIONS

6.1 Conclusion

In this thesis, a random forest algorithm was presented as the best classification method in order to classify the robustness of wireless sensor networks that were derived using gene regulatory network topologies. The five features used to characterize the wireless sensor network are: degree index, network density, motif index, sink coverage and hub node density. A random forest algorithm accurately classified the data into four classes. A locally weighted linear regression algorithm was proposed to predict the average percentage of data received from the five topological features of such bio inspired wireless sensor networks.

By comparing this work with the previous work, it was found that locally weighted linear regression algorithms work better than neural networks for predicting data in all cases except for GRNN neural networks. The second observation is that the best predictions for both neural networks and weighted linear regression algorithms with classifications occurred by using locally weighted linear regression with three classes.

This work is important in its contribution to the estimation of packet transmission efficiency in any WSN application. Furthermore this work may be relevant to other studies of data transmission networks. Finally, the research provides a theoretical model that predicts network robustness based on the five identified topological features.

6. Recommendations and future study

Recommendation and future study considerations based on information gathered during the study are as follows:

1. A lot of data is eliminated by using statistical software; therefore the use of classification and prediction data was not too aggressive. The changes in the value of upper and lower quartile makes the system resist outside concerns.
2. Different results for the same network occur when the weighted linear regression algorithm, the value of the bandwidth parameter (τ) is initialized at random. Before performing a deep dive analysis, it is recommended that a network be run multiple times with various values of bandwidth parameters to observe performance.
3. Despite the random forest providing an accurate classification method for WSN's data, the classification results are not computationally as efficient as other methods.
4. Simulations performed are directly connected to the data in this research. Simulations must be applied in real time to determine the performance of networks in considering the percentage of data received.
5. Four classes were chosen in this work; a change in the number of classes lead to different results. Trying different numbers of classes in future work will be useful in creating more accurate classifiers and predictors for this problem domain.

References

Abdelzaher, A., Bhanu K. Kamapantula, Ghosh, P., &Sajal K., (2012). Empirical prediction of packet transmission efficiency in bio-inspired Wireless Sensor Networks, 705-710.

Aha, D., D. Kibler (1991). Instance-based learning algorithms. *Machine Learning*. 6:37-66.

Alon, U., (2006). An Introduction to Systems Biology Design Principles of Biological Circuits.

Amit, Y., & Geman, D., (1997). Shape quantization and recognition with randomized trees. *Neural Computation, 9*, 1545–1588.

Agoston, V., & P, Cserkmely., (2005).A transcriptional regulatory network as an example. Phys Rev E Stat Nonlin Soft Matter Phys.

Barmpoutis, D. & Murray, R. M. (2010). Networks with the smallest average distance and the largest average clustering. *Phys*.

Bauer, E., (1999).An empirical comparison of voting classification algorithms: bagging, Boosting, and variants. *Machine Learning, 36*, 105–142.

Breiman, L., (1996). Bagging predictors. *Machine Learning, 24*, 123–140.

Breiman, L., (2001). Random Forests. Machine Learning, 45(1).

Breiman, L., (2010). Out-of-bag estimation [EB/OL].
http://stat.berkeley.edu/pub/users/Breiman/OOBestimation.ps

Breiman, L., (1999). Predition games and arcing algorithms. *Neural Computation* 11, 1493–1517.

Beauguitte, L. & Ducruet, C., (2011). *Scale-free and small-world networks in geographical research: A critical examination.* 17th European Colloquium on Theoretical and Quantitative Geography.

Breiman, L., & Jerome, H., (1984). Classification and Regression Trees. Wadsworth International Group, Belmont, California.

Cleary. J. G., Leonard E. (1995).*An Instance-based Learner Using an Entropic Distance Measure*.In 12th International Conference on Machine Learning, 108-114.

Chen, J., & Chun Shao, S., (2012).Data preprocessing using hybrid general regression neural networks and particle swarm optimization for remote terminal units. *International Journal of Control*.

Crucittia, P., & Rapisardab, A., (2004). Error and attack tolerance of complex networks. *Statistical Mechanics and its Applications*, 340,388–394.

Cortez, P., & Sousa, P., (2006). *Internet traffic forecasting using neural networks*. In Proceedings of the IEEE 2006 International Joint Conference on Neural Networks, 4942–4949.

Cohen, R., & Havlin, S., (2000). Resilience of the internet to random breakdowns. *Phys. Rev*, 85.

Cohen, W. W.: Fast Effective Rule Induction. In: Twelfth International Conference on Machine Learning, 115-123, 1995.

Dai, H. & Huabo, L., (2012). *A Multivariate Classification Algorithm for Malicious Node Detection in Large-scale WSNs*, 2012 IEEE 11th International Conference.

Deif, D. S., & Gadallah, Y., (2014). Classification of wireless sensor networks. *Deployment Techniques*, 834 – 855.

Demiroz, G., & Guvenir, A. (1997). *Classification by voting feature intervals*.European Conference on Machine Learning, 85-92.

Dietterich, T., (1998). An experimental comparison of three methods for constructing ensembles of decision trees. *Bagging, Boosting and Randomization. Machine Learning*, 1–22.

Duda, R., Peter Hart (1973). Pattern Classification and Scene Analysis. Wiley, New York.

EI-Aaasser, M., & Ashour M. (2013). *Energy Aware Classification for Wireless SensorNetworks Routing*, 66 – 71.

Eum, S., & Murata, M., (2007). Toward bio-inspired networkrobustness - step 1. Modularity. *Bio-Inspired Models of Network,Information and Computing Systems*, 84–87.

Feyessa, T. & Bikdash, M., (2011).*Measuring nodal contribution to global network robustness*. In Southeastcon, pages 131 –135.

Fagiolo, G., 2007) Clustering in complex directed networks. *Physical Review E*, 76(2), 026107.

Frank, E., & Ian H., (1998) *Generating Accurate Rule Sets Without Global Optimization*.International Conference on Machine Learning, 144-151.

Frank, E., & Wang, Y., (1998). Using model trees for classification. *Machine Learning*, 32, 63-76.

Frank, E., Hall, M., (2003).*Locally Weighted Naive Bayes*. In: 19th Conference in Uncertainty in Artificial Intelligence, 249-256.

Frank, E., Kramer, S., (2004).*Ensembles of nested dichotomies for multi-class problems*. In: Twenty-first International Conference on Machine Learning.

Friedman, J., & Hastie, T., (1998). Additive Logistic Regression: a Statistical View of Boosting. Stanford University.

Freund, Y., Robert, E., (1996). Experiments with a new boosting algorithm. *Thirteenth International Conference on Machine Learning*, 148-156.

Freund, Y., Shapire, R., (1996) *Experiments with a new boosting Algorithm*. Proceedings of the Thirteenth International Conference, pp, 148–156.

Ghosh, P., & Chaitankar, V., (2011). *Principles of genomic robustness inspire fault-tolerant wsn topologies*: A network science based case study. In Pervasive Computing and Communications Workshops, IEEE Intl Conf, 160–65.

Govt., (2011). Study of wireless sensor network in SCADA system for power plant. *Smart Sensors and Ad Hoc Networks*, 2248-9738.

Holmes, G., & Pfahringer, B., (2001). Multiclass alternating decision trees. *ECML*, 161-172.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 832-844.

URL http://citeseer.ist.psu.edu/ho98random.html

Ho, T.K. (1998). The random subspace method for constructing decision forests. *On Pattern Analysis and Machine Intelligence*, 20(8), 832–844.

Ho, T.K., (1995).*Random: Decision Forests*. Proceeding of the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada, 14-18, 278–282.

Holte, R.C. (1993). Very simple classification rules perform well on most commonly used datasets. Machine Learning. 11:63-91.

Hovareshti, P., H. Chen, and J. S. Baras. Motif-based topology design for effective performance by networks of mobile autonomous vehicles. Proc. of 8th Intl Conf on Complex Systems (ICCS), pages 571–573, 2011.

John, G. H., Langley, Pat.,(1995).*Estimating Continuous Distributions in Bayesian Classifiers*.Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345,

Kitano, H., (2004).Biological robustness. *Nat Rev Genet*, 826–837.

Kitano, H., (2007) towards a theory of biological robustness. *Mol Syst Biol*, 3, 137.

Kuncheva, L. I. (2004). Combining Pattern Classifiers: Methods and Algorithms. John Wiley and Sons, Inc.

Kohavi, R. (1995). Wrappers for Performance Enhancement and Oblivious Decision Graphs. Department *of Computer Science, Stanford University*

Kohavi, R., (1995). *The Power of Decision Tables. In: 8th European Conference on Machine Learning*, 174-189.

Kohavi, R., (1996). *Scaling Up the Accuracy of Naive-Bayes Classifiers*. A Decision-Tree Hybrid. In: Second International Conference on Knoledge Discovery and Data Mining, 202-207.

Landwehr, N., & Hall, M., (2005). Logistic Model Trees. *Machine Learning*, 95(1-2), 161-205.

Li J. & Prasant M., (2007).Analytical modeling and mitigation techniques for the energy whole problem in sensor networks. *Pervasive and Mobile Computing*, 3:233 – 254.

Mangan, S. & Alon, U., (2003).Structure and function of the feed-forward loop network motif. *Proc. Natl. Acad. Sci. USA.*

Martin, B., (1995). Instance-Based learning: Nearest Neighbor with Generalization. Hamilton, New Zealand.

McCallum, A., & Nigam, K., (1998). *A Comparison of Event Models for Naive Bayes Text Classification, Learning for Text Categorization*. Workshop conducted at the annual meeting of AAAI.

Melville P., R. J. Mooney: Constructing Diverse Classifier Ensembles Using Artificial Training Examples. In: Eighteenth International Joint Conference on Artificial Intelligence, 505-510, 2003.

Milo, R., & Alon, U. (2002).Network motifs. *Simple building blocks of complex networks. Science*, 25,824–827.

Nelson, P., & Jes´us, G., (2008). Study of the application of neural networks in internet traffic engineering.

Quinlan, R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

Rennie, J. D., (2003).Lawrence shih, jaime teevan. *Tackling the Poor Assumptions of Naive Bayes Text Classifiers*, ICML, 616-623.

Rutka, G., (2006). Neural network models for internet traffic prediction, (4(68)), 5558.
Rodriguez. J. J., & Carlos J., (2006). A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619-1630.

URL http://doi.ieeecomputersociety.org/10.1109/TPAMI.2006.211

Savarese, C., & Jan, M., (2002). *Robust positioning algorithms for distributed ad-hoc wireless sensor networks*. ATEC ’02 Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference, 317 – 327.

Scha_ter, T., & Floreano, D., (2011). Genenetweaver: in silico benchmark generation and performance pro_ling of network inference methods. *Bioinformatics*, 27(5), 2263.

Shi, H. (2007). Best-first decision tree learning. Hamilton, NZ. DecisionStump.

Seewald, A. K., (2002). *How to make stacking better and faster while also taking care of an unknown weakness*. Paper presented at the Nineteenth International Conference on Machine Learning, 554-561.

Seewald, A.K., Fuernkranz, J., (2001). *An Evaluation of Grading Classifiers*. Paper presented at the Advances in Intelligent Data Analysis: 4th International Conference, Berlin/Heidelberg/New York/Tokyo, 115-124.

Tibshirani, R., (1996). *Bias, Variance, and Prediction Error for Classification Rules, Technical Report*, Retrieved from Statistics Department, University of Toronto.

Ting, K. M., & Witten, I., (1997). *Stacking Bagged and Dagged Models*. Fourteenth international Conference on Machine Learning, San Francisco, CA, 367-375.

Trivedi, C., & Mo-Yuen, Chow. (2002).Classification of telnet traffic using artificial neural networks.

Warriach, E.U., &Tei, K., (2012). *A Machine Learning Approach for Identifying and Classifying Faults in Wireless Sensor Networks*.

Wang, C., & Linlin, Z., (2008).*An internet traffic forecasting model adopting radical based on function neural network optimized by genetic algorithm*. In Proceedings of the First International Workshop on Knowledge Discovery and Data Mining, WKDD '08, 367–370.

Wang, F., & Hongbin, X. (2008) Network traffic prediction based on grey neural network integrated model. *CSSE*, 915–918.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5, 241-259.

Wang, X., & Ma J., (2007) Collaborative Peer-to-Peer Training and Target Classification in Wireless Sensor Networks. 208 – 213.

Webb, G. I., (2000). A technique for combining boosting and wagging. *Machine Learning*, Vol.40 (No.2).

Wolpert, D.H., Macready, W.G.: An Efficient Method to Estimate Bagging's Generalization Error. Machine Learning (1997).

Zheng, Z., G. Webb (2000). Lazy Learning of Bayesian Rules. *Machine Learning*, 4(1), 53-84.

APPENDIX A: Calculate interquartile range (IQR), upper cutoff and lower cutoff point

Without classification	interquartile range (IQR)	Upper cutoff point	Lower cutoff point
Feature1	0.047761	0.11216	-0.031123
Feature2	0.027026	0.059381	-0.021697
Feature3	0.079	1.0631	0.8261
Feature4	0.3534	1.0034	-0.0568
Feature5	0.1962	0.6456	0.057

Four classes: class1	interquartile range (IQR)	Upper cutoff point	Lower cutoff point
Feature1	0.00301	0.021275	0.009235
Feature2	0.001595	0.0075085	0.0011285
Feature3	0.0019	0.98735	0.97975
Feature4	0.0566	0.3815	0.1551
Feature5	0.1245	0.54695	0.04895

Four classes: class2	interquartile range (IQR)	Upper cutoff point	Lower cutoff point
Feature1	0.050535	0.1442795	-0.0578685
Feature2	0.019798	0.055384	-0.023808
Feature3	0.0738	1.0968	0.8016
Feature4	0.1282	0.618	0.1052
Feature5	0.17988	0.69182	-0.0277

Four classes: class3	interquartile range (IQR)	Upper cutoff point	Lower cutoff point
Feature1	0.030858	0.11285	-0.010587
Feature2	0.02043	0.066545	-0.015175
Feature3	0.0611	1.07235	0.82795
Feature4	0.1282	0.8407	0.2266
Feature5	0.13208	0.62511	0.09679

Four classes: class4	interquartile range (IQR)	Upper cutoff point	Lower cutoff point
Feature1	0.031127	0.1084555	-0.016
Feature2	0.022492	0.081493	-0.008475
Feature3	0.0821	1.09625	0.76785
Feature4	0.1918	1.1595	0.3923
Feature5	0.2205	0.83545	-0.04

APPENDIX B: Result of different classifier methods by using WEKA software

classsification method		CCI	ROC	KS
BFTree.	Hijian Shi (2007)	90.5495	0.986	0.8288
trees.FT	Joao Gama (2004).	92.7473	0.987	0.871
trees.J48	Ross Quinlan (1993).	94.7253	0.945	0.9049
trees.J48graft	Geoff .W(1999)	94.7253	0.944	0.9047
trees.LADTree	Geoffrey, et al.(2001)	90.955	0.964	0.8318
trees.LMT	Niels et, al.(2005)	92.3077	0.966	0.8624
treesNBTree	Ron Kohavi (1996)	92.967	0.958	0.875
Random Forest		94.7253	0.984	0.9048
Random Tree		93.1868	0.932	0.8773
SimpleCart	Leo Breiman (1984)	91.8681	0.936	0.8521
ComplementNaiveBayes	Jason D (2003)	51.681	0.712	0.356
DMNBtext	Jiang Su,Harry et, al.(2008)	61.7582	0.501	0
NaiveBayes	George et, al.(1995)	87.4725	0.969	0.7875
NaiveBayesMultinomial	Andrew, et al. (1998)	61.7582	0.739	0
NaïveBMUpdateable	Andrew, et al.(1998)	87.4725	0.969	0.7875
NaiveBMBSimple	Richard, et al.(1973)	87.9121	0.97	0.7949
NaiveBUpdateab	Gorge, et al.(1995)	87.4725	0.969	0.7875
SMO	J.Platt, et al.(1998)	86.3736	0.921	0.7583
IB1	D.Aha,et al.(1991)	91.4286	0.923	0.8469
IBK	D.Aha,et al.(1991)	91.4286	0.923	0.8469
Kstar	John, et al.(1995)	92.0879	0.983	0.8587
LWL	Eibe, et al.(2003)	74.9451	0.903	0.558
meta.AdaBoost	Yoav, et al.(1996)	70.989	0.827	0.4463

meta.CVR E et al. (1998)	92.7473	0.984	0.8672
meta.CVPS R, et al.(1995)	61.7582	0.486	0.2818
meta.D Ting, et al.(1997)	63.7363	0.856	0.0745
metaDecorate P, et al.(2003)	94.2857	0.972	0.8962
meta.END Eibe(2004)	93.4066	0.957	0.8811
meta.Filteredclassifier	88.1319	0.941	0.7809
meta.Grad A.K, et al.(2001)	61.7582	0.5	0
meta.LB J.F et al.(1998)	93.4066	0.932	0.98
meta.MBAB Geoffrey(2000)	72.5257	0.647	0.815
meta.NDDNBND Lin(2005)	93.6264	0.948	0.8844
meta.OCC Eibe et, al.(2001)	94.2857	0.947	0.897
metaRILB Eibe, et, al.(2002)	61.7582	0.486	0
meta.RSS Tin, et, al.(1998)	91.6484	0.975	0.8462
meta.RF Juan, et, al.(2006)	94.0659	0.989	0.8928
meta.Stacking David, et al.(92)	61.7582	0.486	0
meta.StackingC A.K(2002)	61.7582	0.486	0
meta.Vote Ludila(2004)	61.7582	0.486	0
miscVFI G.D et, al.(1997)	71.4286	0.901	0.5851
rule.DecisionTable Ron(1995)	86.3736	0.927	0.7513
rulesJRip William(1995)	91.2088	0.921	0.8397
rulesNNge Brent(1995)	92.0879	0.923	0.8578
rules.OneR R.C(1993)	75.6044	0.767	0.5599
rules.PART Eibe et, al.(1998)	91.4286	0.959	0.8451

ROC= Receiver Operating Characteristic, CCI=Correctly Classified Instance KS= Kappa statistic

FINSHED