ABSTRACT

NETWORK ANOMALY DETECTION USING MANAGEMENT INFORMATION BASE (MIB) NETWORK TRAFFIC VARIABLES

by
Jun Li

In this dissertation, a hierarchical, multi-tier, multiple-observation-window, network anomaly detection system (NADS) is introduced, namely, the MIB Anomaly Detection (MAD) system, which is capable of detecting and diagnosing network anomalies (including network faults and Denial of Service computer network attacks) proactively and adaptively. The MAD system utilizes statistical models and neural network classifier to detect network anomalies through monitoring the subtle changes of network traffic patterns. The process of measuring network traffic pattern is achieved by monitoring the Management Information Base (MIB) II variables, supplied by the Simple Network Management Protocol (SNMP) II. The MAD system then converted each monitored MIB variable values, collected during each observation window, into a Probability Density Function (PDF), processed them statistically, combined intelligently the result for each individual variable and derived the final decision. The MAD system has a distributed, hierarchical, multi-tier architecture, based on which it could provide the health status of each network individual element. The inter-tier communication requires low network bandwidth, thus, making it possibly utilization on capacity challenged wireless as well as wired networks.

Efficiently and accurately modeling network traffic behavior is essential for building NADS. In this work, a novel approach to statistically model network traffic measurements with high variability is introduced, that is, dividing the network traffic measurements into three different frequency segments and modeling the data in each frequency segment separately. Also in this dissertation, a new network traffic statistical model, i.e., the one-dimension hyperbolic distribution, is introduced.

NETWORK ANOMALY DETECTION USING MANAGEMENT INFORMATION BASE (MIB) NETWORK TRAFFIC VARIABLES

by
Jun Li

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

August 2004

APPROVAL PAGE

NETWORK ANOMALY DETECTION USING MANAGEMENT INFORMATION BASE (MIB) NETWORK TRAFFIC VARIABLES

Jun Li

Dr. Constantine Manikopoulos, Dissertation Advisor — Date
Associate Professor, Electrical and Computer Engineering, NJIT

Dr. Ali Akansu, Committee Member — Date
Professor, Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member — Date
Associate Professor, Electrical and Computer Engineering, NJIT

Dr. Sirin Tekinay, Committee Member — Date
Associate Professor, Electrical and Computer Engineering, NJIT

Dr. George Antoniou, Committee Member — Date
Professor, Computer Science, Montclair State University

BIOGRAPHICAL SKETCH

Author: Jun Li

Degree: Doctor of Philosophy

Date: August 2004

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering
  New Jersey Institute of Technology, Newark, NJ, 2004
- Master of Science in Electrical Engineering
  Tianjin University, Tianjin, P. R. China, 1998
- Bachelor of Science in Electrical Engineering
  Tianjin University, Tianjin, P. R. China, 1995

Major: Electrical Engineering

Publications and Presentations:

J. Li and C. Manikopoulos, "A Novel Statistical Network Model: The Hyperbolic Distribution," accepted by *IEE Proceedings on Communications*.

J. Li and C. Manikopoulos, "Denial of Service (DoS) Anomaly Intrusion Detection Using Management Information Base (MIB) Network Traffic Parameters," in preparation.

J. Li and C. Manikopoulos, "The Application of a Low Pass Filter in Early Detection of Network Anomalies and Intrusions," in preparation.

J. Li and C. Manikopoulos, "Statistical Traffic Modeling of Seasonal Variations in Proactive Network Anomaly Detection," in preparation.

J. Li and C. Manikopoulos, "Classifier Training for Anomaly Intrusion and Fault Detection in a Production Network Using Test Network Information," in preparation.

Z. Zhang, J. Li, C. N. Manikopoulos, J. Jorgenson, J. Ucles, "Neural Networks in Anomaly Intrusion Detection," *Advances in Scientific Computing, Computational Intelligence and Applications*, N. Mastorakis, V. Madlenov, B. Suter and L. J. Wang (Eds.), World Scientific and Engineering Society Press, ISBN: 960-8052-36-X, 2001.

Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles, "A Hierarchical Anomaly Network Intrusion Detection System Using Neural Network Classification," *Advances in Neural Networks and Applications*, N. Mastorakis (Ed.), World Scientific and Engineering Society Press, ISBN: 960-8052-26-2, 2001.

J. Li and C. Manikopoulos, "The Application of a Low Pass Filter in Anomaly Network Intrusion Detection," in *Proceedings of the 5th Annual IEEE Systems, Mans, and Cybernetics Information Assurance Workshop*, (West Point, NY), June 2004.

J. Li and C. Manikopoulos, "On Using the Hyperbolic Distribution to Statistically Model Network LAN Traffic Time-Segments", in *Proceedings of the 7th WSEAS International Multiconference on Circuits, Systems, Communications and Computers*, (Corfu, Greece), July 2003.

B. He, J. Li and C. Manikopoulos, "Wireless Mobile Ad Hoc Network (MANET) Emulation Using the Hardware Switch Software Router (HSSR) Platform", in *Proceedings of the 7th WSEAS International Multiconference on Circuits, Systems, Communications and Computers*, (Corfu, Greece), July 2003.

J. Li and C. Manikopoulos, "Early Statistical Anomaly Intrusion Detection of DOS Attacks Using MIB Traffic Parameters" in *Proceedings of the 4th Annual IEEE Systems, Mans, and Cybernetics Information Assurance Workshop*, (West Point, NY), June 2003.

J. Li and C. Manikopoulos, "Network Fault Detection Using MIB Traffic Parameters", in *Proceedings of the 37th Annual Conference on Information Sciences and Systems*, (Baltimore, MD), Mar. 2003.

J. Li and C. Manikopoulos, "Network Fault Detection: Classifier Training Method for Anomaly Fault Detection in a Deployed Network Using Test Network Information", in *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, (Tampa, FL), Nov. 2002.

J. Li and C. Manikopoulos, "Investigation of the performance of GAFT, a novel network anomaly fault detection system", in *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, (Tampa, FL), Nov. 2002.

J. Li and C. Manikopoulos, "Simulating and Analyzing Self-similarity of Internet Round Trip Delay Using OPNET Modeler", in *Proceedings of OPNETWORK2002*, (Washington, D.C.), Aug. 2002.

J. Li and C. Manikopoulos, "Modeling Distributed Denial of Service Attacks Using OPNET Modeler", in *Proceedings of OPNETWORK2002*, (Washington, D.C.), Aug. 2002.

J. Li, S. Xu, C. Manikopoulos and S. Papavassiliou, "Anomaly Network Intrusion Detection for AD-HOC Mobile Wireless Networks", in *Proceedings of the 3rd Annual IEEE Systems, Mans, and Cybernetics Information Assurance Workshop*, (West Point, NY), June, 2002.

J. Li and C. Manikopoulos, "Anomaly Intrusion Detection for Hierarchical Network Architectures", in *Proceedings of the 36th Annual Conference on Information Sciences and Systems*, (Princeton, NJ), Mar. 2002.

J. Li, C. Manikopoulos and J. Jorgenson, "The Investigation of Internet Round Trip Delay", in *Proceedings of International Conference on Computing and Information Technologies*, (Montclair, NJ), Oct. 2001.

Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, J. Ucles, "Neural Networks in Statistical Intrusion Detection", in *Proceedings of the 5th World Multiconference on Circuits, Systems, Communications and Computers*, (Rethymnon, Greece), July 2001.

Z. Zhang, J. Li, C. Manikopoulos, J Jorgenson, J. Ucles, "HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification", in *Proceedings of the 2nd Annual IEEE Systems, Mans, and Cybernetics Information Assurance Workshop*, (West Point, NY), June 2001.

Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, J. Ucles, "A Hierarchical Anomaly Network Intrusion Detection System Using Neural Network Classification", in *Proceedings of the 2001 WSES International Conference on Neural Networks and Applications*, (Tenerife, Canary Islands), Feb. 2001.

This work is dedicated to my beloved family

ACKNOWLEDGEMENT

I would like to take this opportunity to thank my advisor Dr. Constantine Manikopoulos for offering me a chance to work under his guidance and supervision. I am deeply grateful to him for his valuable suggestions, continuous support and consistent encouragement.

Special thanks are given to Dr. Ali Akansu, Dr. Edwin Hou, Dr. Sirin Tekinay and Dr. George Antoniou for actively participating in my committee, reviewing this work and providing valuable criticism and suggestions.

My wife, Jie Shen, contributed significantly to the completion of this dissertation. She was emotionally supportive while pushing me to get things done. She did lots of work so that I could concentrate on my dissertation research. I want to thank her wholeheartedly for her support.

TABLE OF CONTENTS

Chapter		Page
1	INTRODUCTION	1
	1.1 Background	2
	1.1.1. Signature Based Network Anomaly Detection System	2
	1.1.2. Statistical Based Network Anomaly Detection System	7
	1.2 The Proposed Approach	9
	1.3 Roadmap of This Dissertation	11
2	A NOVEL STATISTICAL BASED NETWORK ANOMALY DETECTION USING MIB – MIB ANOMALY DETECTION (MAD) SYSTEM	12
	2.1 The Hierarchical Architecture of MAD	13
	2.2 Anomaly Detection Agent: The Basic Network Anomaly Detection Unit	14
	2.2.1. MIB Variable Collection	16
	2.2.2. The Statistical Model	20
	2.2.3. Neural Network Classifier	30
3	THE PERFORMANCE EVALUATION PROCESS	33
	3.1 The Evaluation Testbed Configuration	33
	3.2 Experiment Results and Discussion	35
	3.2.1. The Results for Investigating the Efficaciousness of the Partition Schemes	36
	3.2.2. The Results for Investigating the Effectiveness of the Similarity Measurement Metrics	38
4	THE GRAFTED AND RE-USE CLASSIFIER TRAINING METHODS	42

TABLE OF CONTENTS
(Continued)

Chapter		Page
	4.1 Methodology	42
	4.2 Evaluation Process	49
	4.2.1. The Testbed Configuration	49
	4.2.2. Numerical Results and Discussion	52
5	THE APPLICATION OF LOW PASS FILTERS IN NADS DESIGN	56
	5.1 Comparing Savitzky-Golay Filter with Moving Window Average Filter	59
	5.2 The Butterworth Filter	61
	5.3 Do the Low Pass Filters Alternate the Statistical Properties of Network Traffic Measurements?	63
	5.4 Can the Low Pass Filters Reduce the False Alarm Rate of NADS?	67
6	A FREQUENCY BASED NETWORK TRAFFIC STATISTICAL MODELING SCHEME	71
	6.1 Motivation	71
	6.2 A Frequency Based Network Traffic Statistical Modeling Scheme	73
	6.2.1. Modeling Low Frequency Part Signal	74
	6.2.2. Modeling Middle Frequency Part Signal	77
	6.2.3. Modeling High Frequency Part Signal	79
	6.3 Model Evaluation	85
	6.3.1. Data Collection	85
	6.3.2. Model Evaluation Results and Discussion	87
	6.4 Summary	92

TABLE OF CONTENTS
(Continued)

Chapter	Page
7 CONCLUSIONS AND FUTURE WORK.....	94
7.1 Conclusions.....	94
7.2 Future Work.....	96
REFERENCES.....	97

LIST OF TABLES

Table		Page
2.1	List of the Selected MIB Variables in the Implementation of MAD.....	20
3.1	Experimental Ethernet Improper Short Frame Anomaly..... Scenario Specification	35
3.2	The Mean Square Root Error (x100) Neural Network..... Convergence Criterion in the Emulation of the Partition Scheme	37
3.3	The Misclassification Rate in the Emulation of the Partition Scheme..... (in percentage)	37
3.4	The False Positive Rate in the Emulation of the Partition Scheme (in percentage)	37
3.5	The False Negative Rate in the Emulation of the Partition Scheme (in percentage)	38
3.6	The Mean Square Root Error (x100) Neural Network..... Convergence Criterion in the Emulation of the Effectiveness of the Similarity Measurement Metrics	40
3.7	The Misclassification Rate in the Emulation of the Effectiveness of the..... Similarity Measurement Metrics (in percentage)	40
3.8	The False Positive Rate in the Emulation of the Effectiveness of the Similarity Measurement Metrics (in percentage)	41
3.9	The False Negative Rate in the Emulation of the Effectiveness of the..... Similarity Measurement Metrics (in percentage)	41
4.1	The Production Network Configuration.....	53
4.2	The Test Simulation Network Configuration.....	53
4.3	The Misclassification Rate of the Simulation Experiments.....	54
5.1	Mean, Variance, Standard Deviation and Hurst Parameter.....	66
5.2	The Improvement of the Mean Square Root Error (x100).....	69

LIST OF TABLES
(Continued)

Table		Page
5.3	The Improvement of the Misclassification Rate (in Percentage).....	69
6.1	The K-S Distances for Modeling the Low Frequency Part Signal.....	75
6.2	The K-S Distances for Modeling the Middle Frequency Part Signal.....	79
6.3	The Averaged K-S Distance Between the CDFs of the Segmented..... High frequency Part Signal Data	82
6.4	Investigating the deviation Between the CDFs of the Segmented High..... frequency Part Signal Data	82
6.5	The Fitting Results for the High Frequency Part Signal.....	84
6.6	The Specification of the Data Traces for the Model Evaluation	87
6.7	The Evaluation Results for Modeling the Low Frequency Part Signal.....	88
6.8	The Evaluation Results for Modeling the Middle Frequency Part Signal.....	88
6.9	Investigating the Deviation Between the CDFs of the Segmented High..... frequency Part Signal Data	89
6.10	The Evaluation Results for Modeling the High Frequency Part Signal.....	90

LIST OF FIGURES

Figure		Page
2.1	A sample network	13
2.2	System hierarchy	14
2.3	Anomaly detection agent	16
2.4	Case diagrams for the Interface, IP, UDP and TCP group MIB variables	19
2.5	The XLPS partition scheme	22
2.6	The XSRPS partition scheme	23
2.7	The YLPS partition scheme	23
2.8	The YSRPS partition scheme	24
2.9	An illustration of the weight function with 16 bins	27
2.10	The neural network classifier architecture	31
3.1	Schematic of the testbed network facility	34
4.1	The grafted classifier method	49
4.2	The test network	50
4.3	The production network	51
4.4	The ROC curves	55
5.1	Comparing the different characteristics of traffic burst and network anomaly	57
5.2	The frequency responses of the 2^{nd}, 4^{th}, and 8^{th} order Butterworth low pass filters	62
5.3	The frequency responses of the low pass filters under investigation	63
5.4	A sample of the network traffic measurements collected from the main router of Bergen County Library Network System	64

LIST OF FIGURES

Figure		Page
5.5	Autocorrelation	66
5.6	Cumulative density function	66
5.7	The refined architecture of ADA	67
5.8	The improvement of mean square root error	68
5.9	The improvement of misclassification rate	68
6.1	Plot of three weeks of traffic variable observations	72
6.2	Plot of one day of traffic variable observations	73
6.3	A sample of one day's traffic variable observations collected from the main router of BCCLNS	74
6.4	The original and sampled low frequency part signal data	76
6.5	The CDF plots of the original and sampled low frequency part signal data	76
6.6	The original and sampled middle frequency part signal data	78
6.7	The CDF plots of the segmented middle frequency part signal data	78
6.8	The CDF plots of the original and sampled middle frequency part signal data	79
6.9	The high frequency part signal data	81
6.10	The CDF plots of the segmented high frequency part signal data	81
6.11	The fitting results for the high frequency part signal	84
6.12	The simplified abbreviated network topology of BCCLNS	86
6.13	The CDF plots of the original and sampled low frequency part signal data for the data trace MF-1	90

LIST OF FIGURES

Figure		Page
6.14	The CDF plots of the original and sampled middle frequency part signal.... data for the data trace MF-1	91
6.15	The fitting results for the high frequency part signal for the data trace..... MF-1	91

CHAPTER 1

INTRODUCTION

In this information era, many claim that with the advent of the web and Internet, the future has arrived. The dream of an interconnected planet where physical labor becomes minimally important and knowledge creation becomes the source of value and wealth appears to be here. The Internet is becoming more and more important for the world: today, the Internet has become one of the most important carriers of information; people also start to use the Internet to receive education, shop for groceries and do banking and stock transactions. The new information and communication technologies increase the installation choices. Bill Gates believes "it will affect the world seismically, rocking us in the same way the discovery of the scientific method, the invention of printing, and the arrival of the Information Age did."

However, today's communication network is still not very reliable and safe. Network anomaly happens every once in a while. Network anomalies typically refer to circumstances when network operations deviate from normal behavior [1-2]. Network anomalies can arise due to various causes such as malfunctioning network devices, network overload, malicious denial of service (DOS) attacks and network intrusions that disrupt the normal delivery of network services; and may appear as severe network/service outages or failures, or performance degradation which may finally lead to severe failure due to untimely correction. Traditional network anomaly/fault management emphasizes detection and processing of serious service failures and alarms [3]. This method is necessary but when network alarms are captured, filtered and analyzed, service and network failures are already present. Therefore, traditional anomaly management is more

reactive in nature. In order to let network operate stably and reliably, one has to deal with those anomalies which lead to network/service performance degradation. Since network performance degradations are signatures of network anomalies and are preludes to service failures, being able to detect them early and automatically enabling timely and rapid anomaly containment and correction, through which serious network and service failures can be avoided. This approach complements the traditional anomaly management.

1.1 Background

In the past few years, network anomaly detection has become an active research field. Many approaches were introduced to the design of the network anomaly detection system (NADS). Basically, the proposed NADSs fall into two major categories according to their functionality: signature based NADS and statistical based NADS.

1.1.1 Signature Based Network Anomaly Detection System

General speaking, the signature based NADS carries out its functionality based on the knowledge/signatures obtained from the previously occurred anomalies. Many approaches were studied in the design of the signature based NADS, including Expert System, Bayesian Network, and Finite State Machine. In the remainder of this section, each approach will be reviewed and its advantages and problems will be given.

1.1.1.1 Expert System. The IMPACT system, proposed by Jakobson *et al.* [4], is a typical example of a signature based NADS using Expert System. The IMPACT system employs an exhaustive database containing knowledge of previously occurred network anomalies as "if-then" rules. The "if part" contains a symptom. This symptom is tested against the current network situation and if the symptom is met, the "then-part" is

executed, possibly activating other rules. When the symptoms of all rules of an anomaly are activated, the anomaly is identified. Thus, the identification of anomalies heavily relies on the symptoms that are specific to a particular manifestation of an anomaly. In [5], a computationally based Expert System was proposed to manage anomaly propagation in internetworks using the Fuzzy Cognitive Maps (FCM). The dynamic features of FCM are exploited to characterize the time-varying aspects of network anomalies, while its graphical features were used as a framework for representing the distributed properties of anomaly propagation.

Expert System is a good approach for diagnosing anomalies, as it appears relatively easy to implement and can provide accurate diagnosis. However, there are potential flaws after some observations. When a potential issue arises, there is a strict definition of what constitutes an anomaly from the diagnosis system point of view; either a potential anomaly matches a rule of what an anomaly should be or it is defined as not being an anomaly whatsoever. This means that the system cannot intelligently indicate what could be a potential new anomaly arising. Another flaw derived from this is the human intervention involved to create a new anomaly profile. To create a new profile, the administrator will have to detect and pinpoint the anomaly "by hand" and insert its characteristics and properties into the knowledge database in the form of a rule. This discrepancy makes these unscalable and very sensitive to "noise" data. Finally, a minor flaw detected within the system is the potential flagging of duplicate anomalies: two separate symptoms of a single anomaly flagged by two individual rules, thus creating two anomaly indications.

1.1.1.2 Bayesian Network. Bayesian Network (BN) is another solution for implementing a signature based NADS due to their ability to handle uncertainty and represent cause and effect relationships. A BN is a representation consisting of nodes representing uncertain variables that are connected by arcs that represent cause and effect dependencies among the nodes. The information known about one node (i.e. effect node) depends on the information of its predecessor nodes that represent its causes. The relationship is expressed by a probability distribution of each effect node, based on the possible values of its predecessor nodes' variables. Notes that an effect node can also lead into other nodes, where it then plays the role of a cause node. An important advantage that BSs offer is the avoidance of building huge joint probability distribution tables that include permutations of all the nodes in the network.

The BNs can represent deep knowledge by modeling the functionality of the transmission network in terms of cause and effect relationships between element and network behavior and anomalies. Also they can provide guidance in anomaly diagnosis. Calculations over the same BN can determine both the precedence of anomaly alarms and the areas that need further clarification in order to provide a finer grained diagnosis. Several approaches were studied to design a signature based NADS using BNs. A typical example can be found in [6], where Hood and Ji proposed an anomaly detection scheme based on autoregressive models and BNs. In their work, a simple BN was constructed to derive the network node level anomaly detection result from combining the network performance variable level anomaly indicators. In [7], Huard and Lazar used a more general BN model with multiple root nodes as the candidate anomalies. They also presented a dynamic programming (DP) formulation for the network troubleshooting

problem. In [8], an anomaly detection framework was proposed based on a BN with multiple root nodes chosen as the knowledge representation scheme. In their work, multiple anomalies could be handled concurrently and the anomaly diagnosis procedure was formulated as a partially observable Markov decision processes.

The development of a diagnostic BN requires a deep understanding of the network configurations in a domain, provided by domain experts. Also an anomaly detection scheme based on BNs needs to maintain the possibility distribution that expresses the relationship of the cause and effect used to establish the BN for each network anomaly. Similar to the problem met in the Expert System, obtaining such information relies heavily on network experts and it should be frequently updated in fast evolving network environments. These constraints badly limit the capability of an anomaly detection scheme based on BNs to operate in a new network environment.

1.1.1.3 Finite State Machine. An anomaly detection system based on finite state machine (FSM) [9] models the network, and its behavior when an anomaly occurs, as a FSM. It consists in a set of states, with transitions between states dictated by input events such as anomaly alarms coming from the network. Each of the states defines either the correct functioning of the network or a failure scenario. Therefore, only the failure scenarios considered in the FSM can be identified. In [10-11], Bouloutas *et al.* proposed a signature based NADS using the FSM model. The system consists of two FSMs, G and A, one observing the behavior of the other. The behavior of G is described by G_b before the anomaly, and by G_a after the anomaly. The signals from G, after processed by a maximal filter (i.e. the smallest set of symbols is passed), is fed into the observer A which can detect that an anomaly took place. Wang and Schwartz [12] refined this detection scheme by

constructing multiple independent FSM observers and proposing a fast real-time anomaly detection mechanism to eliminate synchronization problems arising in the original FSM decomposition method. Another work using FSMs in the design of a signature based NADS was presented in [13]. First, the right FSM was constructed for each anomaly. And then, given the anomaly model, the FSM correlates alarms "on-line" and identifies the anomalies at the origin of the observed alarms. The advantage of this work is that an extension of the Viterbi algorithm was used to deal with the corrupted data, thus enhancing the system's capability to handle the alarm symbols with deletions, additions and changes.

General speaking, the advantage of the FSM model is that state machines are designed with the intention of not just detecting an anomaly but also possibly identifying and diagnosing the problem. The disadvantage of using FSM in network anomaly detection lies in that the requirement that a FSM should be established for each existing anomaly makes this approach of no capability to detect unknown anomalies. This is also complicated by the fact that not all anomalies can be captured by a finite sequence of alarms of reasonable length. Furthermore, the already established FSMs should be updated each time the network configurations (including hardware and software) are changed.

1.1.1.4 Summary. It is well known that a single network anomaly may generate several alarms at different segments within a network domain [14-15]. To identify the anomaly at the original place, a network manager should check the alarms one by one. This is impractical, and may soon be impossible, since networks and their dynamics are getting ever more complex. The main advantage of the signature based NADS is its capability of correlating several alarms generated by a single anomaly and providing the network manager more accurate information on the type and location of the anomaly, thus making

the task of solving the problem much easier. However, to correlate the alarms, the signature based NADS requires the specification of the anomaly and the detailed network configurations (both hardware and software). Obviously, these requirements badly limit the performance of these approaches since it is not feasible to specify all possible anomalies; and much more importantly, it is almost impossible to specify every detail of the network configurations in a large scaled network. In addition, changes in network configuration, either the hardware or the software, can change the types and nature of anomalies that may occur, making modeling anomaly more difficult and in many cases impractical.

1.1.2 Statistical Based Network Anomaly Detection System

Typically, statistical approaches generate statistical measures to determine how far the observed behavior deviates from the previously measured one. Activity measures, like the consumed CPU time, the consumed network bandwidth and the number of service invocations, are typically taken as measures. Usually several of these measures are included in a profile. Some systems merge the currently measured profile with the stored one, while others keep the profile constant for a certain amount of time.

In [16-17], the authors proposed and implemented a real-time statistical based anomaly detection system which used to identify anomaly condition in a transaction-orientated wide area network. In their work, they attempt to deal with the variability in the network traffic behavior. A normal traffic threshold is built from historical data. These thresholds are then categorized by time of day, day of week and special days, such as weekend and holidays. When newly measured data fails to fit within some confidential interval of the threshold, an anomaly alarm is generated. The challenge in this case

concerns the threshold themselves: what proper confidential interval of the threshold to set for each measured network performance variable and what are their corresponding values. This is complicated by the well-known fact that performance variables (such as utilization) of networks undergo cyclic evolution and temporal fluctuation [18-19]. However, improper settings of the thresholds may render high false alarm rate of the anomaly detection scheme. These challenges are being tackled in [20-21].

Another typical example of the statistical based NADS design was presented in [22-23]. In this system, the generalized likelihood ratio test and duration filter were employed to measure the deviations of the behavior of six network performance variables in two successive observation windows. However, identifying anomalies as significant deviation between the measurements of the performance variables in two adjacent observation windows may raise the false alarm rate, especially in a network environment with highly variable traffic behavior. In this design, there is an underlying assumption that the detection scheme considers the data in the first one of the two adjacent observation windows as the normal profile. Thus, if a detection error occurs, this error will be propagated in the future detection results.

General speaking, the main advantages of the statistical based NADS over the signature based NADS include: 1) its operation doesn't need the specifications of the anomalies to be detected and the detailed configurations of the network; 2) it has the capability to detect unknown anomalies. However, there are three major difficulties encountered by the statistical based anomaly detection scheme:

a) The statistical based NADS can only alert the occurrence of network anomalies, but it can't provide further information or clues for pinpointing the location of the anomaly, thus complicating the task of solving the problem;

b) Obviously, the operation of the statistical based network anomaly detection scheme relies heavily on accurate normal profiles. Indeed, the more accurately the network performance parameters can be modeled, the better the anomaly detection system will perform [24]. However, the accurate definitions of normal behavior for measured network performance parameters are challenging tasks, dependent on several network specific factors, such as the dynamics of the network configurations, the types of applications running on the network and the statistically averaged user service command pattern for a particular service. Actually, accurate modeling of normal network behavior is still an active field of research, especially the online modeling of network traffic [25-27]. For most measured network data, there are no analytically simple models that can be used to learn the normal behavior.

c) Recent measurements of local-area and wide-area network traffic demonstrate that they exhibit fractal behavior [28-29] and high burstiness [30]. The highly variable nature of traffic behavior may affect the performance of an anomaly detection system by raising the false alarm rate even though an accurate traffic normal profile can be achieved.

Since the advantages and disadvantages of the signature based and statistical based network anomaly detection schemes have been demonstrated, an attempt can be made to find an approach which will combine the advantages of signature based and statistical based network anomaly detection techniques.

1.2 The Proposed Approach

In this dissertation, a novel design of a statistical based network anomaly detection system, namely MIB Anomaly Detection (MAD) system, is proposed. The advantages of MAD over other NADSs presented in the previous section can be summarized as follows:

- The MAD system is of a hierarchical, multi-tier architecture, with which MAD can provide the operating status of each device affiliated to the network domain. In case that an anomaly occurs, even though several alarms may be rendered at different segments within the network domain, using this information the network manager can track the original location of the network anomaly very easily. Not like the signature based anomaly detection schemes, tracking the location of the anomaly in such a way doesn't require the specification of the anomaly and the detailed network configurations.

- The MAD system detects network anomalies using only network performance parameters that are provided by the Management Information Base (MIB) variables. Currently simple network management protocol (SNMP) is already installed in most computer networks, thus the collection of local information required by anomaly detection system needs little in additional resources. Also the standardized representation of the data collected in each node facilitates data exchange between nodes. The potential use of MIB variables in network anomaly detection has been explored previously in [22-23] by Thottan and Ji. The drawback of their work has been discussed in Subsection 1.1.2. In this dissertation, a simpler, more efficient anomaly detection algorithm has been developed which takes advantage of the statistical differences of the MIB variable measurements before and after an anomaly occurs.

- As discussed previously, most earlier statistical based NADSs simply measured the means and variances of the monitored network performance variables and detected whether certain thresholds were exceeded; in nonstationary systems that often do not follow the normal distribution, such systems generate incorrect decisions. To overcome some of this problem, MAD system presented the measured MIB variable data in probability density function (PDF) format rather than isolated sample values, and used statistical model (also mentioned as goodness-of-fit statistical tests [31-32]) and a neural network classifier [33] to identify network anomalies. Our simulation results demonstrated that MAD could reliably detect the anomalies with traffic intensity as low as 1% of the typical background traffic. These results show promise for the use of MAD in detecting the anomalies in their early stages before they develop into serious failures.

- As mentioned in last section, one challenge that is faced in the design of a statistical based NADS is that the detection scheme may encounter difficulties in distinguishing network anomalies from traffic bursts due to their similar characteristics, which may lead to high false alarm rate. To enhance the anomaly detection capability of MAD and reduce its false alarm rate, low pass filters are introduced to reduce the burstiness in the measurements of the MIB variables.

- As mentioned previously, to achieve reliable detection results, the statistical based NADS needs to maintain an accurate and efficient statistical model for each parameter monitored. This task is challenged by the well-known fact that performance parameters of networks exhibit high variability and cyclic evolution [19]. In this dissertation, a novel approach is introduced to statistically model the network traffic measurements with high variability, i.e., dividing the network traffic measurements into three different frequency segments and modeling the data in each segment separately. Also a new network traffic statistical model, i.e., the one-dimension hyperbolic distribution, is introduced.

1.3 Roadmap of This Dissertation

The whole dissertation is presented in six chapters:

1) Introduction;
2) A Novel Statistical Based Network Anomaly Detection Based on MIB – MIB Anomaly Detection (MAD) system;
3) The Performance Evaluation Process;
4) The Grafted and Re-use Classifier Training Methods;
5) The Application of Low Pass Filters in NADS Design;
6) A Frequency Based Network Traffic Statistical Modeling Scheme;
7) Conclusions and Future Work.

Chapter 2 presents the implementation of the proposed MAD system, where each system component is elaborated and corresponding algorithms are presented in detail. A systemic performance evaluation process for MAD is shown in Chapter 3. In Chapter 4, two approaches for adequately training the neural network classifier in a new production network environment are presented in detail, namely the re-use and the grafted classifier methods. In Chapter 5, the employment of low pass filters in the NADS to smooth the burstiness in network traffic measurements and thus reduce the false alarms is investigated. In Chapter 6, a frequency based network traffic statistical modeling scheme is presented. Finally, Chapter 7 evaluates the work of this dissertation as a whole, concludes and outlines further research possibilities.

CHAPTER 2

A NOVEL STATISTICAL BASED NETWORK ANOMALY DETECTION USING MIB – MIB ANOMALY DETECTION (MAD) SYSTEM

In this chapter, a hierarchical, multi-tier, multiple-observation-window, statistical based Network Anomaly Detection System (NADS) using only Management Information Base (MIB) II supplied traffic related variables is introduced, as carried out by the MIB Anomaly Detection (MAD) system which is capable of detecting and diagnosing the anomalies (network/service anomaly or performance degradations) proactively and adaptively. The MAD system utilizes statistical models and neural network classifier to identify network anomaly through detecting the subtle changes in network traffic. It monitors many MIB variables simultaneously, analyzes statistically their performance, combines intelligently the individual decisions and derives an integrated result of service compliance. This provides variable specific decisions on service compliance as well as combined-variable decisions employing a neural network classifier, all based on calculations using PDFs rather than individual or averaged sampled values, to detect network/service anomalies with low false alarm rates. The MAD system has a distributed, hierarchical, multi-tier architecture, based on which MAD could provide the health status of each network individual element. Using this information, the network manager may locate the network anomaly much more easily. The inter-tier communication requires low network bandwidth, thus, making it possibly utilization on capacity challenged wireless as well as wired networks.

2.1 The Hierarchical Architecture of MAD

MAD is, in general, a distributed application, deployed hierarchically over several tiers, with each tier containing several Anomaly Detection Agents (ADAs). ADAs are the basic anomaly detection components that monitor the activities of the network to which they are attached. Different tiers correspond to different network scopes that are monitored by the agents affiliated to them.

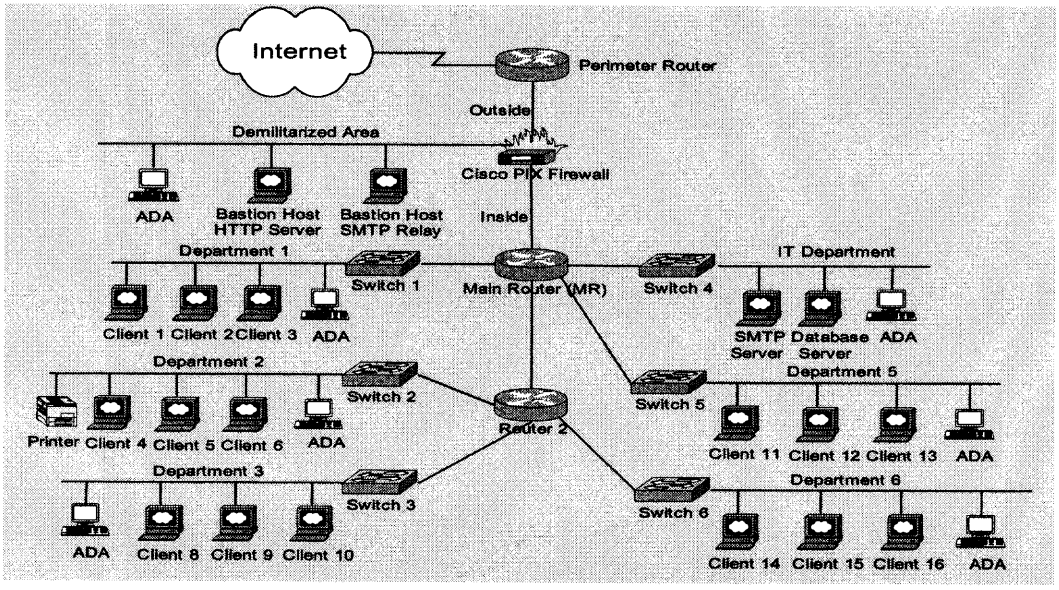

Figure 2.1 A sample network.

For the sample network shown in Figure 2.1, the anomaly detection system can be divided into 2 tiers, that is, Tier1 ADA and Tier2 ADA. A Tier1 ADA monitors the activities of each network device, including workstation, server, switch and each interface of the router within a departmental LAN, through measuring its MIB variables. Through statistically analyzing the MIB variable data collected from each network device, the Tier1 ADA derives a Network Device Status Indicator (NDSI) that presents the operating status of the monitored network device; and then generates a report, which contains the NDSI for each network device and its corresponding IP address, and sends to the Tier2 ADA. The

Tier2 ADA collects MIB variable data from the firewall and the routers as well as the reports from Tier1 ADAs. The system hierarchy is shown in Figure 2.2.

From this example, one can clearly see that with the hierarchical architecture MAD system can monitor the operating status of each device within the network domain. In case that an anomaly occurs, although several alarms may be generated at different segments of the network, it is very easy to track the anomaly at its originally occurred place using the detection results provided by the ADAs at different tiers. Furthermore, it is advantageous over the signature based NADS that tracking the anomaly in such a way doesn't require the specification of the anomaly and detailed network configurations.

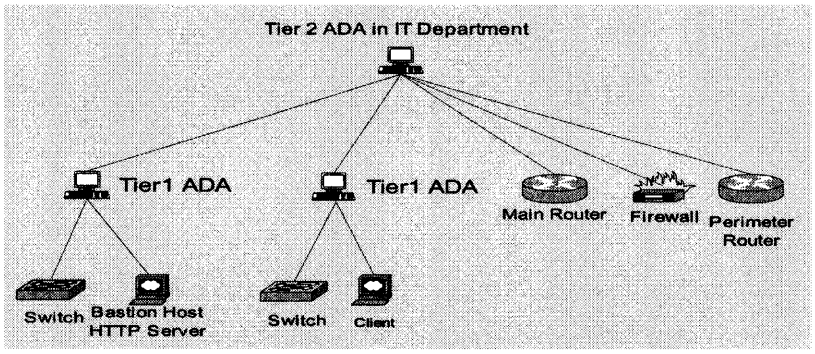

Figure 2.2 System hierarchy.

2.2 Anomaly Detection Agent: The Basic Network Anomaly Detection Unit

Anomaly Detection Agent (ADA) is the basic network anomaly detection unit, which uses statistical model and a neural network based classifier to detect anomalous network conditions. The monitoring process utilizes multi-layered time windows, ranging from a few seconds to several hours or more, each layer aggregating the layer below. ADA monitors many MIB variables simultaneously from each network element affiliated to it, analyzes statistically their performance, combines intelligently the individual decisions and

derives an integrated result of service compliance. The philosophy of ADA is that the heavier part of the analysis of the data be carried out early on, at the statistical analysis stage, before the observed status data arrives at the classifier, in order to make the critical and delicate task of classifying as easy as possible.

The statistical component builds and analyzes real-time probability density functions (PDFs) of the monitored MIB variables and continuously compares the measured PDFs to preset or generated reference PDF models of normal activity. This provides variable specific decisions on service compliance as well as combined-variable decisions employing a multivariate classifier, all based on calculations using PDFs rather than individual or averaged sampled values, to detect network/service anomalies or failures with low false alarm rates. The statistical analysis bases its calculations on PDF algebra, in departure from the commonly used isolated sample values or perhaps their averages. The use of PDFs is much more informative, allowing greater effectiveness than just using averages. A diagram of an ADA is illustrated in Figure 2.3, which consists of the following components: the MIB data probe, the low pass filter, the statistical model, the neural network classifier and the post processor. The functionality of these components is described below:

- MIB Data Probe: Collects MIB variables from each individual network element the ADA is attached to, abstracts the data into a set of statistical variables to reflect the network status, and periodically generates reports to the low pass filter module.
- The Statistical Model: Maintains a reference model of the typical network and host activities, compares the reports from the low pass filter module to the reference models, and then generates a variable-level anomaly decision for each of the monitored MIB variables individually and forms a anomaly status vector (ASV) that combines all or groups of variable-level anomaly detections to feed into the neural network classifiers.
- Neural Network Classifier: Processes the ASV vector from the statistical model to decide whether each host affiliated to the ADA operates in normal condition.

- Post Processor: Generates reports for the high tier ADA.

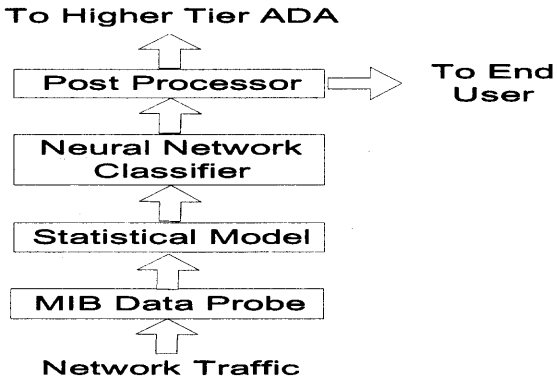

Figure 2.3 Anomaly detection agent.

2.2.1 MIB Variable Collection

In network anomaly detection systems, gathering the network traffic data needed to evaluate the operating status of the network is a significant portion of the overall processing burden. However, many networks already deploy SNMP based network management [34], thus MIB variables are available to be collected, in many network entities. By enlisting the MIB objects, MAD promises a lower overhead approach for analysis by the anomaly detection engine. The SNMP server queries the SNMP agents and retrieves the value of MIB objects to perform monitoring functions. It can also modify the value of specific MIB variables thus changing the settings of agents. According to configuration and policy specification, agents can send unsolicited information to the network manager or controller. The advantages of this approach are:

- If an SNMP agent already is operating at the node, as is likely, the collection of local information needs little in additional resources.
- A large number of traffic related performance parameters are readily available, as needed for network anomaly detection.

- The standardized representation of the data collected in each node facilitates data exchange between nodes.
- It can be extended to collect additional data relative to network activities.
- It does not depend on the operating system.

Choosing a subset of MIB variables is the first important step toward developing an efficient network anomaly detection system. The Management Information Base maintains a database of 171 variables [35]. According to the functionalities described by these variables, they fall into the following eleven groups: System, Interfaces, Address Translation (AT), Internet Protocol (IP), Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Exterior Gateway Protocol (EGP), OSI Internet Management (OIM), Transmission, and Simple Network Management Protocol (SNMP). Each group of MIB variables describes a specific functionality of a network entity. In terms of the syntax ruled by Structure of Management Information, these MIB variables can be categorized into ten types [36]. In the implementation of MAD, only the *counter* type MIB variables are selected since this type of variable changes frequent, thus can provide more clues for tracing the subtle changes in the network traffic pattern.

Selecting the appropriate MIB variables should be carried out in two steps. The first step is selecting proper MIB variable groups and the second is selecting the appropriate variables in the groups selected in the first step. Since commonly encountered network anomalies occur in the data link, network, and transport layers of IP/data networking environment, one should pay more attention to the Interface, IP, TCP and UDP MIB variable groups. Within a particular MIB variable group, there exists some redundancy. Thus, when selecting the variables within a specific group, the first thing one

needs to do is avoid the redundant variables and ensure that key information is indeed captured in at least one variable. The reason for wanting all key information is that: the anomaly detection system should be provided with all important clues for capturing the occurred anomaly. The desire to avoid redundant variables stemmed from a concern that the operating efficiency of the system designed may be reduced if implementations contained excessive instrumentation. To explore the relationship of MIB variables within a particular group so as to avoid the redundant variables, the case diagram [37] is utilized. The case diagrams for the Interface, IP, UDP and TCP (divided into two parts, one presents the relationship between the TCP connection/session related MIB variables and the other presents the relationship between the TCP segment related MIB variables) are illustrated in Figure 2.4. Considering these case diagrams and the potential use for the anomaly detection, 27 MIB variables are selected, as listed in Table 2.1.

In the Interface group, nine MIB variables are selected as listed in Table 2.1. When the network operates in the normal situation, these variables may provide redundant information for the network anomaly detection system. For instance, consider the variable interface outgoing unicast packet rate *ifOutUcastPkts*, interface outgoing non-unicast packet rate *ifOutNUcastPkts* and the interface outgoing byte rate *ifOutOctets*. The variable *ifOutUcastPkts* and *ifOutNUcastPkts* actually contain the same traffic information which may be provided by *ifOutOctets* in case of the normal network operating situation. However, this principle doesn't hold in some anomaly network operating situation. To understand this point, one may consider the following scenario. Assume that the Ethernet card of a host malfunctions due to improper configuration and sends out network frames with improper small size. In such a situation, the final destination or the medium relay

devices may receive a flooding of small packets. In measuring the variables ifOutOctets at the faulty host, one may not find out the abnormal activities since the outgoing byte rate doesn't change so much. On the other hand, the variables ifOutUcastPkts demonstrates significant change since the packet with normal size from upper layers is cut and encapsulated into frames with improper small size. Considering this issue, the implementation of the proposed network anomaly detection algorithm adopts all of these three MIB variables although they may provide redundant traffic information.

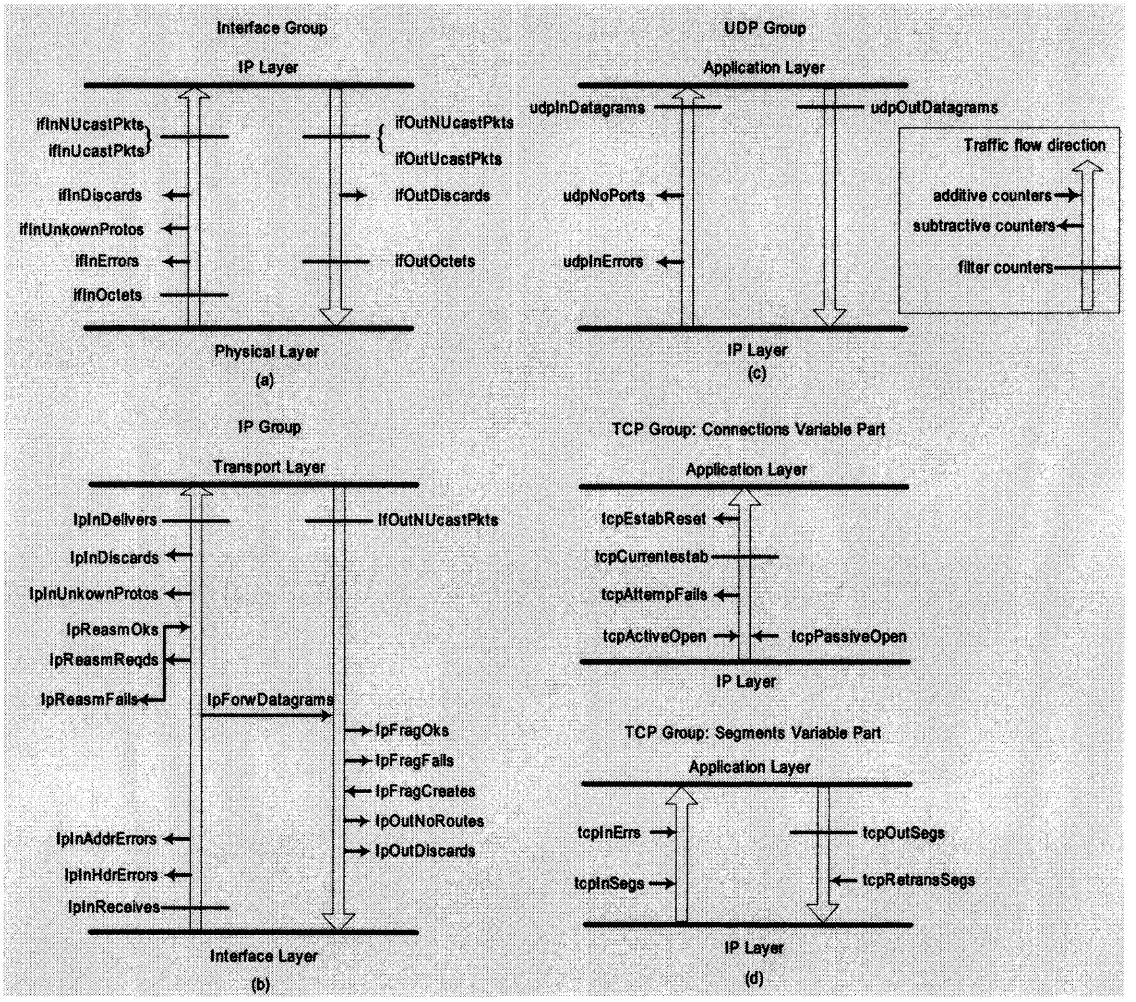

Figure 2.4 Case diagrams for the Interface, IP, UDP and TCP group MIB variables.

In the IP and UDP groups, six and four MIB variables are selected respectively. From the case diagram illustrated in Figure 2.4(b) and 2.4(c), obviously these variables are non-redundant. The *counter* type MIB variables in the TCP group may be divided into two categories, i.e., connection related variables and TCP segment related variables. Four MIB variables are selected in each category. From the case diagram shown in Figure 2.4(d), one can clearly see that these selected variables are non-redundant.

Table 2.1 List of the Selected MIB Variables in the Implementation of MAD

MIB Variable Group	MIB Variable Name
Interface Group	ifInNUcastPkts
	ifInUcastPkts
	ifInDiscards
	ifInErrors
	ifInOctets
	ifOutNUcastPkts
	ifOutUcastPkts
	ifOutDiscards
	ifOutOctets
Internet Protocol Group	ipInDiscards
	ipInAddrErrors
	ipInHdrErrors
	ipInReceives
	ipOutNUCastPkts
	ipOutDiscards
User Datagram Protocol Group	udpInDatagrams
	udpNoPorts
	udpInErrors
	udpOutDatagrams
Transport Control Protocol Group Connections Variable Part	tcpPassiveOpens
	tcpActiveOpens
	tcpAttempFails
	tcpEstabResets
Transport Control Protocol Group Segments Variable Part	tcpInErrs
	tcpInSegs
	tcpOutSegs
	tcpRetransSegs

2.2.2 The Statistical Model

The statistical model provides the two basic functionalities, i.e., 1) convert the MIB traffic variable data into PDF format, and 2) compare the real-time PDF to the preset or generated reference PDF models using the similarity measurement algorithm to generate the variable

level abnormality indicator. In this section, these two functions will be described in detail.

2.2.2.1 Data Format Conversion. Representing the MIB variable measurements into PDF formats involves partitioning the sample spaces of the measurements into a set of complete non-overlapping bins and calculating the frequencies (i.e., probabilities) that the observed events fall within particular bins of the PDF histogram, for the partition scheme deployed. The choices of the partition scheme and the value of the total number of bins utilized in that partition scheme are determined by tradeoffs that optimize classification effectiveness along with the cost of system resources, such as computational complexity and storage and processor memory. Yet, little systematic investigation seems to have been undertaken on this issue. In this dissertation, seven different partition schemes were investigated: *uniform*, *uniform percentile*, *logarithmic*, *x-axis linear*, *x-axis square root*, *y-axis linear* and *y-axis square root*.

- *Uniform Partition Scheme (UPS)*: The UPS is the most straightforward one; it partitions the sample space into bins of equal width. Assume $x_0, x_1, ...x_N$ is the partition boundaries of the sample space with the minimum x_0 and the maximum x_N, where N is the total number of bins. The partition boundaries can be calculated using the following equations:

$$x_1 - x_0 = x_2 - x_1 = ... = x_N - x_{N-1} \tag{2.1}$$

$$x_i = x_0 + i \times \frac{x_N - x_0}{N},\ i = 1,2,.....N \tag{2.2}$$

- *Uniform Percentile Partition Scheme (UPPS)*: The UPPS is an extension version of UPS. It generates bin boundaries so that a coming event falls into each bin with equal probability. Assume F(x) is the cumulative distribution function (CDF) of a PDF. Thus:

$$F(x_1) - F(x_0) = F(x_2) - F(x_1) = ... = F(x_N) - F(x_{N-1}) = \frac{1}{N} \tag{2.3}$$

$$x_i = F^{-1}(\frac{i}{N}), i = 1,2,.....N-1 \tag{2.4}$$

- *Logarithmic Partition Scheme (LPS)*: The LPS is designed to divide the sample space into the segments with the same width in the logarithm order, as represented more clearly by the following equations.

$$\log_{(1+x_N-x_0)}(1+x_i-x_0)=\frac{i}{N}, i=1,2,.....N-1 \quad (2.5)$$

$$x_i=x_0-1+(1+x_N-x_0)^{i/N}, i=1,2,.....N-1 \quad (2.6)$$

- *X-axis Linear Partition Scheme (XLPS)*: The XLPS is designed to partition the sample space symmetrically with the symmetric point placing at the central point of the sample space. At the left side of the symmetrical point, the bin span increases linearly when the bin index increases. On the contrary, at the right side of the symmetrical point the bin width decreases linearly when the bin index increases. Figure 2.5 may illustrate the XLPS partition scheme more clearly. The bin boundaries can be calculated by the following equations.

$$x_1-x_0=\frac{1}{2}(x_2-x_1)=\frac{1}{3}(x_3-x_2)..=\frac{1}{N/2}(x_{\frac{N}{2}}-x_{\frac{N}{2}-1})=\frac{1}{N/2}(x_{\frac{N}{2}+1}-x_{\frac{N}{2}})=...=\frac{1}{2}(x_{N-1}-x_{N-2})=x_N-x_{N-1}=\frac{4(x_N-x_0)}{N(2+N)} \quad (2.7)$$

$$x_i=x_{i-1}+i\times\frac{4(x_N-x_0)}{N(2+N)}, i=1,2,...\frac{N}{2} \quad (2.8)$$

$$x_i=x_{i-1}+(N+1-i)\times\frac{4(x_N-x_0)}{N(2+N)}, i=\frac{N}{2}+1,...N \quad (2.9)$$

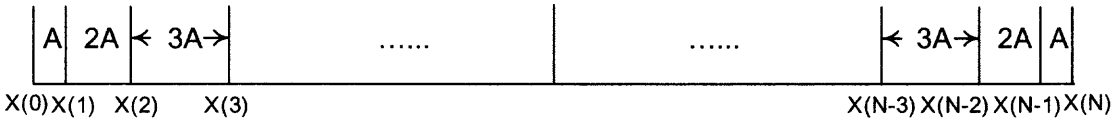

Figure 2.5 The XLPS partition scheme.

- *X-axis Square Root Partition Scheme (XSRPS)*: The XSRPS is similar to XLPS except that the bin width is decreasing or increasing at the left or right side of the symmetrical point at a rate of the square root of its index, as illustrated in Figure 2.6. The bin boundaries can be calculated by the following equations.

$$x_1-x_0=\frac{1}{\sqrt{2}}(x_2-x_1)=\frac{1}{\sqrt{3}}(x_3-x_2)..=\frac{1}{\sqrt{N/2}}(x_{\frac{N}{2}}-x_{\frac{N}{2}-1})=\frac{1}{\sqrt{N/2}}(x_{\frac{N}{2}+1}-x_{\frac{N}{2}})=...=\frac{1}{\sqrt{2}}(x_{N-1}-x_{N-2})=x_N-x_{N-1}=\frac{(x_N-x_0)}{2\sum_{n=1}^{N2}\sqrt{n}} \quad (2.10)$$

$$x_i=x_{i-1}+\sqrt{i}\times\frac{(x_N-x_0)}{2\sum_{n=1}^{N/2}\sqrt{n}}, i=1,2,...\frac{N}{2} \quad (2.11)$$

$$x_i=x_{i-1}+\sqrt{(N+1-i)}\times\frac{(x_N-x_0)}{2\sum_{n=1}^{N/2}\sqrt{n}}, i=\frac{N}{2}+1,...N \quad (2.12)$$

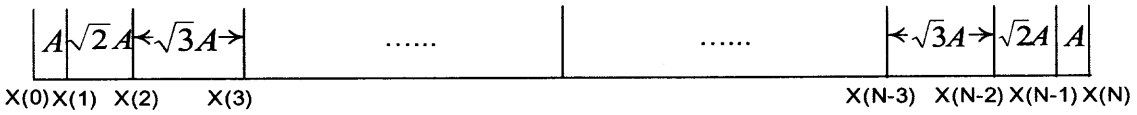

Figure 2.6 The XSRPS partition scheme.

- *Y-axis Linear Partition Scheme (YLPS)*: In the YLPS, the sample space is segmented according to the probability that a new arrival data event falls in a bin. If considering the probability space as the sample space, then the YLPS will be the same as the XLPS. Figure 2.7 may show this algorithm more clearly. Assume F(x) is the cumulative distribution function of a PDF, and then the bin boundaries can be calculated by the following equations.

$$F(x_1)-F(x_0)=\frac{F(x_2)-F(x_1)}{2}=\frac{F(x_3)-F(x_2)}{3}=\cdots=\frac{F\left(x_{\frac{N}{2}}\right)-F\left(x_{\frac{N}{2}-1}\right)}{\frac{N}{2}}=\frac{F\left(x_{\frac{N}{2}+1}\right)-F\left(x_{\frac{N}{2}}\right)}{\frac{N}{2}}=\cdots=\frac{F(x_{N-1})-F(x_{N-2})}{2}=F(x_N)-F(x_{N-1})=\frac{4}{N(2+N)} \quad (2.12)$$

where $F(x_0)=0$ and $F(x_N)=1$.

$$x_i=F^{-1}[F(x_{i-1})+i\times\frac{4}{N(2+N)}],i=1,2,\ldots\frac{N}{2} \quad (2.13)$$

$$x_i=F^{-1}[F(x_{i-1})+(N+1-i)\times\frac{4}{N(2+N)}],i=\frac{N}{2}+1,..N \quad (2.14)$$

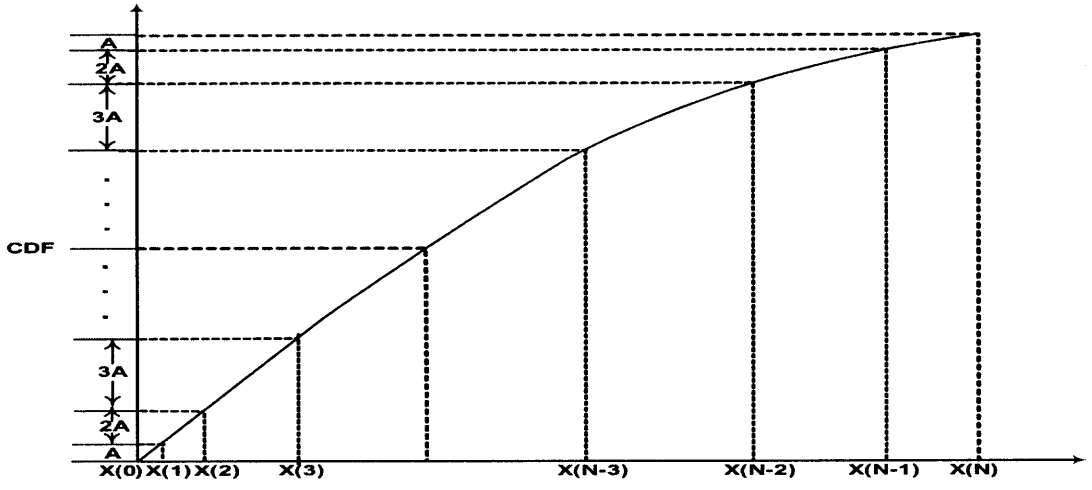

Figure 2.7 The YLPS partition scheme.

- *Y-axis Square Root Partition Scheme (YSRPS)*: As with the YLPS, the YSRPS partitions the sample space according to the probability that a new arrival data event falls in a bin. As illustrated in Figure 2.8, the YSRPS partitions the probability space of

the sample space in a similar manner that XSRPS partitions the sample space. The bin boundaries can be calculated by the following equations.

$$F(x_1)-F(x_0)=\frac{F(x_2)-F(x_1)}{\sqrt{2}}=\frac{F(x_3)-F(x_2)}{\sqrt{3}}=\cdots=\frac{F\left(x_{\frac{N}{2}}\right)-F\left(x_{\frac{N}{2}-1}\right)}{\sqrt{\frac{N}{2}}}=\frac{F\left(x_{\frac{N}{2}+1}\right)-F\left(x_{\frac{N}{2}}\right)}{\sqrt{\frac{N}{2}}}=\cdots=\frac{F(x_{N-1})-F(x_{N-2})}{\sqrt{2}}=F(x_N)-F(x_{N-1})=\frac{1}{2\sum_{n=1}^{\frac{N}{2}}\sqrt{n}} \quad (2.15)$$

where $F(x_0)=0$ and $F(x_N)=1$.

$$x_i=F^{-1}[F(x_{i-1})+\sqrt{i}\times\frac{1}{2\sum_{n=1}^{N/2}\sqrt{n}}],i=1,2,\ldots\frac{N}{2} \quad (2.16)$$

$$x_i=F^{-1}[F(x_{i-1})+\sqrt{N+1-i}\times\frac{1}{2\sum_{n=1}^{N/2}\sqrt{n}}],i=\frac{N}{2}+1,..N \quad (2.17)$$

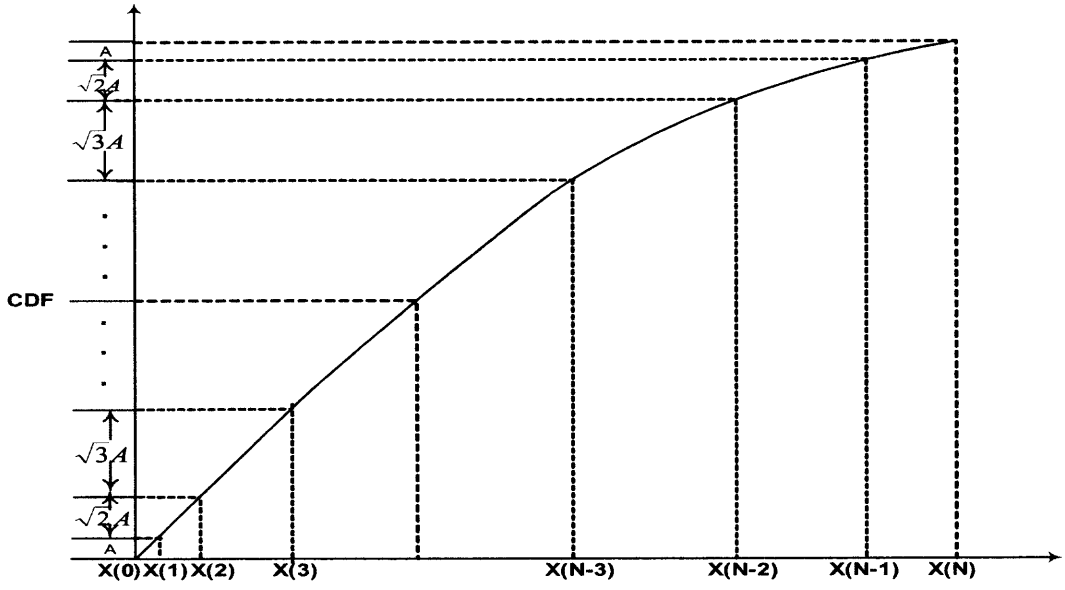

Figure 2.8 The YSRPS partition scheme.

2.2.2.2 Similarity Measurement Metrics. Statistical methods have been used in security and fault management systems to detect anomaly network activities; however, most of these systems simply measure the means and the variances of some variables and detect

whether certain thresholds are exceeded. Valdes, et al, developed a more sophisticated statistical algorithm, for SRI's NIDES system, by using a χ^2-like test to measure the similarity between short-term and long-term profiles [38]. Cabrera, et al, used Kolmogrov-Smirnov (KS) statistics to model and detect anomaly traffic patterns [39].

In this work, the similarity measurement metric (SMM) is mainly used to compare the real-time observed Probability Density Function (PDF) or Cumulative Density Function (CDF) of the MIB variable measurements to its corresponding preset or generated reference PDF or CDF models. The distance or associated probability produced by the SMM for each MIB variable will be used to construct an anomaly status vector (ASV) and fed into the Neural Network Classifier for further process. In the design of MAD, the comparative efficiency of seven groups, seventeen SMMs, was investigated.

Notation:

s(i): the values at bin i of the observed PDF histogram partition.

r(i): the values at bin i of the reference PDF histogram partition.

S(i): represents the value of bin i of observed CDF.

R(i): represents the value of bin i of reference CDF.

N_S: represents the number of samples in the observed PDF or CDF

N_R: represents the number of samples in the reference PDF or CDF

N_{PDF} : represents the number of bins in the partition of the observed or reference PDF

A. χ^2 type Tests (CST). The χ^2 type tests have been employed in the analysis of data in a number of scientific areas since they were introduced by Mise [40]. In this dissertation, two versions of χ^2 type tests are under investigation. The χ^2 distances and an

associated probability value V of the SMMs, for the two versions of χ^2 type tests, are given by:

- χ^2 type test version 1 (CST1): $\chi^2 = \sum_i \frac{[s(i) \times N_S - r(i) \times N_R]^2}{r(i) \times N_R}$ (2.18)

$$V = \frac{\tan^{-1}(0.01 * \chi^2)}{\pi/2} \quad (2.19)$$

- χ^2 type test version 2 (CST2): $\chi^2 = \sum_i \frac{[s(i) \times N_S - r(i) \times N_R]^2}{s(i) \times N_S + r(i) \times N_R}$ (2.20)

$$V = \frac{\tan^{-1}(0.01 * \chi^2)}{\pi/2} \quad (2.21)$$

B. Kolmogorov-Smirnov (KS) type Tests (KST). In the design of MAD, four versions of KS type tests are studied. The Kolmogrov-Smirnov (KS) test, in general, offers the advantage that it is distribution independent [41]. The KS type tests are given as below.

- KS type test version 1 (KS1): The KS distance of KS1 is represented by the maximum difference of the values of the bins of observed and reference CDF, as shown by the following equation.

$$D = \max_i |S(i) - R(i)| \quad (2.22)$$

- KS type test version 2 (KS2): The KS distance of KS2 is calculated by picking up the maximum difference of the values of the bins of observed and reference PDF.

$$D = \max_i |s(i) - r(i)| \quad (2.23)$$

- KS type test version 3 (KS3 or called Combined Area KS): The distance of the KS3 is shown below:

$$D = \max_i |s(i) - r(i)| + \sum_i |s(i) - r(i)| \quad (2.24)$$

  The first term is the KS test, while the second term represents an area difference between the monitored and reference PDFs.

- KS type test version 4 (KS4): The distance and associate probability of the KS4 is given by:

$$D = \max_i |S(i) - R(i)| \quad (2.25)$$

$$\Pr ob = 2\sum_{i=1}^{\infty} (-1)^{i-1} e^{-2i^2 D^2} \tag{2.26}$$

where $\lambda = [\sqrt{N_e} + 0.12 + 0.11 / \sqrt{N_e}] \times D$ and $N_e = \frac{N_S \times N_R}{N_S + N_R}$

C. Weighted Similarity Statistic (WSS). Since the network anomalies mainly reflect as the traffic overwhelm which appears as the extreme high traffic rate, or congestion which appears as the extreme low traffic rate, a successful design of the similarity measurement metric should be most sensitive to the difference between the current observed and reference PDFs/CDFs at the extreme ends of the distribution of the traffic measurements, but not the median part. Considering that, a weight function is designed, which amplifies the difference between the values in the bins of the observed and reference PDFs at the extreme ends, but reduces those at the medium part, as shown in Equation 2.27. Figure 2.9 illustrates the weight function with 16 bins.

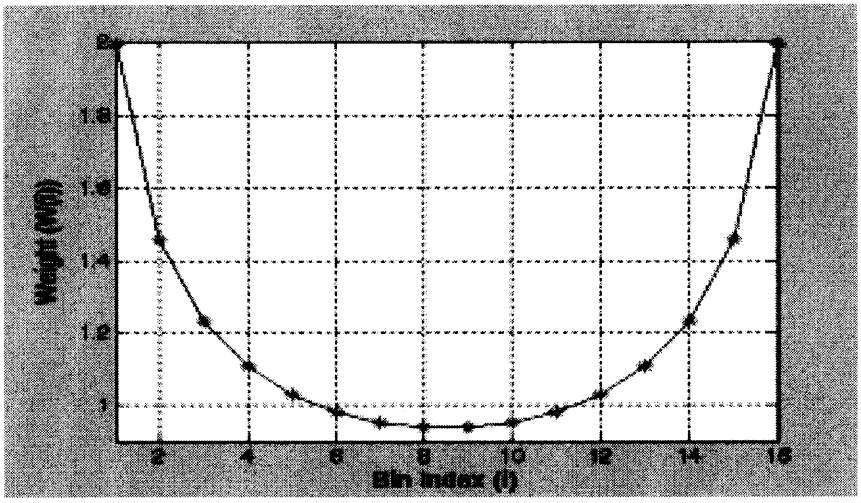

Figure 2.9 An illustration of the weight function with 16 bins.

$$W(i) = \sqrt{\frac{1/4 (N_{PDF}^{\ 2} - 1)}{i * (1 + N_{PDF} - i)}} \quad i = 1,.....N_{PDF} \tag{2.27}$$

Three versions of weighted SMMs are shown in the following. The weight function proposed above is applied in the metrics WSS1 and WSS2; while another weight function, derived from Anderson-Darling similarity measurement test [42], is applied in the metric WSS3.

- Weighted similarity statistic version 1 (WSS1): The distance of WSS1 statistic may be calculated by the following equation:

$$D = \sum_i |S(i) - R(i)| \times r(i) \times W(i), \; i = 1,.....N_{PDF} \tag{2.28}$$

- Weighted similarity statistic version 2 (WSS2): The distance of WSS2 statistic may be calculated by the following equation:

$$D = \max_i |s(i) - r(i)| \times W(i), \; i = 1,.....N_{PDF}. \tag{2.29}$$

- Weighted similarity statistic version 3 (WSS3): The distance of WSS3 statistic may be calculated by the following equation:

$$D = \sum_i |S(i) - R(i)| \times r(i) \times W'(i) \tag{2.30}$$

where $W'(i) = 1/\sqrt{R(i)[1-R(i)]} \;\; i = 1,.....N_{PDF}$.

D. Kupier's KS type Statistic (KKS). Kupier's statistic is the sum of the maximum deviations separating the S(x) from the R(x) distribution in the positive as well as negative directions [43]. In this work, two versions of Kupier's KS type statistic are under investigation.

- Kupier's KS type statistic version 1 (KKS1): The distance for KKS1 is given as below.

$$D = \max_i [S(i) - R(i)] + \max_i [R(i) - S(i)] \tag{2.31}$$

- Kupier's KS type statistic version 2 (KKS2): The distance and an associate probability value V of the statistical similarity metric are presented by the following equations.

$$D = \left[\sqrt{N_e} + 0.155 + 0.24/\sqrt{N_e}\right] * \{\max_i [S(i) - R(i)] + \max_i [R(i) - S(i)]\} \tag{2.32}$$

where $N_e = \dfrac{N_S \times N_R}{N_S + N_R}$.

$$V = 2\sum_{i=1}^{\infty} (-1)^{i-1} e^{-2i^2 D^2} \tag{2.33}$$

E. Fractional Deviation type Statistic (FDS). In the design of MAD, two versions of FDS are studied, as illustrated below.

- Fractional deviation type statistic version 1 (FDS1): The distance and an associated probability value V of the FDS1 are given below.

$$D = \sum_i \frac{|s(i) \times N_S - r(i) \times N_R|}{r(i) \times N_R} \tag{2.34}$$

$$V = \frac{\tan^{-1}(0.01 \times D)}{\pi/2} \tag{2.35}$$

- Fractional deviation type statistic version 2 (FDS2): The distance and an associated probability value V of the FDS2 are shown as below:

$$D = \Sigma_i \left[\frac{|s(i) \times N_S - r(i) \times N_R|}{\frac{s(i) \times N_S + r(i) \times N_R}{2}} \right] \tag{2.36}$$

$$V = \frac{\tan^{-1}(0.01 \times D)}{\pi/2} \tag{2.37}$$

F. Single Number Statistics (SNS)

- Single number statistic version 1 (SNS1): The distance of the SNS1 can be calculated through the following two steps:

i. Calculate the mean of the observed PDF: $m = \sum_i is(i)$ (2.38)

ii. $D = 2\frac{r(m)}{r_{\max}} - 1$ (2.39)

- Single number statistic version 2 (SNS2): The procedure to calculating the distance of the SNS2 is given by the following two steps:

i. Calculate the mean of the observed PDF: $m = \sum_i is(i)$ (2.40)

ii. $D = 2R(m) - 1$ (2.41)

G. Other types of Similarity Statistics

- Fractional Deviation from the Mean (FDM): Let A_S be the average of the measurements of a monitored variable in a given observation time window and A_R be the average of the measurements of the reference model for that variable, then the distance of the fractional deviation from the mean statistic is given as:

$$D = (A_S - A_R)/A_R \tag{2.42}$$

- Fractional Deviation from Mean over Standard Deviation (FDMSD): The distance of the fractional deviation from the standard deviation is given as:

$$D = \frac{(A_S - A_R)}{\sigma_R} \tag{2.43}$$

where σ_R stands for the standard deviation of the measurements of the reference model.

Extensive experiments have been carried out, to evaluate the effectiveness of the partition schemes and similarity measurement metrics presented above. The investigation results will be presented in Chapter 3.

2.2.3 Neural Network Classifier

Neural network classifiers are widely considered as an effective approach to detect network anomaly and intrusion. Ghosh et al., used backpropagation (BP) neural networks to detect anomalous user activities [44]. Jiang et al., built a network security management system using artificial intelligence technologies [45].

In MAD, a neural network classifier is utilized. The type of neural network adopted here is the typical BP network, shown in Figure 2.10, characterized by a 3-layer architecture. The BP is a multi-layer feedforward network, which contains an input layer, one or more hidden layers, and an output layer. BPs have strong generalization capabilities and have been applied successfully to solve a variety of difficult and diverse problems, especially as they may relate to pattern recognition challenges. For this study, after testing

BP networks with the number of hidden neurons ranging from 2 to 8, it was found that two hidden neurons were sufficient to carry out the task at hand. This results in a rather small neural network, which economizes computation and storage and improves behavior.

During initialization, a sequence of labeled ASVs is provided as input to the neural network classifier in order to train it. During run time, the ASVs are submitted to the neural network for classification. The neural network processes the pattern of the ASV components at hand and generates a value $v \in [-1, 1]$, where $v = -1$ corresponds to an anomaly and $v=1$ to normal conditions, with absolute certainty. For simplicity, it is presumed here that values for v in-between these two extremes, represent anomaly or normalcy with confidence proportional to the numerical magnitude of v. Thus, using the neural net classifier, MAD's decision process combines the similarity information of all PDFs in one integrated and unified result. This combining is powerful in that it achieves higher discrimination and decision robustness.

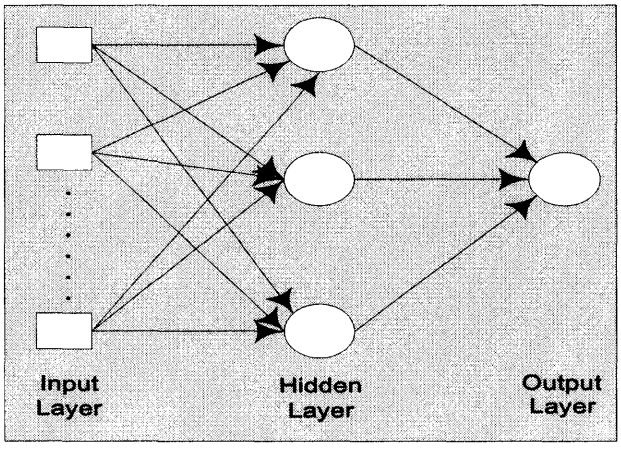

Figure 2.10 The neural network classifier architecture.

In the MAD system design, an algorithm for the real-time updating of the reference model was also designed. Here, the output of the BP neural network classifier is a continuous variable u that attains values between -1 and 1, where -1 means anomaly, with

absolute certainty, and 1 means normalcy, again with complete confidence. In between, the values of u indicate proportionate levels of certainty. Ordinarily, the value of u=0 separates anomalies (u<0) from normal occurrences (u≥0). The function for calculating s is

$$s = \begin{cases} u, & \text{if } u \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.44}$$

Let r_{old} be the reference model before updating, r_{new} the reference model after updating, and r_{obs} the observed normal activity within a particular time window. The formula to update the reference model is:

$$r_{new} = s \times \alpha \times r_{obs} + (1 - s \times \alpha) \times r_{old} \tag{2.45}$$

where α is a system operator defined learning rate, while s, the output of the neural network classifier when detecting a normal event, serves here as a dynamic adaptation rate that is proportional to the confidence value.

Through the above equations, it was ensured that the reference model would be updated actively and proportionately for typical traffic, while kept unchanged when anomalies occurred. The anomaly events will be diverted and stored for future neural network learning.

CHAPTER 3

THE PERFORMANCE EVALUATION PROCESS

3.1 The Evaluation Testbed Configuration

A research stand-alone network was constructed for the purpose of carrying out actual network anomaly experiments in a controlled setting. The network topology is shown in Figure 3.1. It comprises four network subnet segments, connected by a layer 3 switch, labeled L3S. Each subnet consists of several workstations of various operating systems, namely, Windows, Linux and Solaris. A powerful PC located in Subnet 4 serve as HTTP, SMTP and FTP servers, and visited by the workstations in Subnet 1, 2 and 3. The L3S switch provides the routing functionality needed for communication from one network segment to another. In these experiments, traffic with self-similar characteristics that mimics Internet type traffic was constructed, as the background traffic of a given intensity.

Many actual network experiments have been carried out focusing on the various network anomalies; including ethernet broadcast storm, ethernet runt flood, ethernet improper long/short frame errors and so on. In order to demonstrate the integrated performance of the proposed MAD system, representative results collected during nine different Ethernet improper short frame error scenarios are presented here, which were constructed by improperly configuring Network Interface Card (NIC) of the workstation located in Subnet 1 and triggering off transmitting ethernet frames with improper short size, namely, 100 bytes/frame, from it. These scenarios ranged from modest (10%) to small (0.5%) values of the ratio of anomaly to background, R=(A/B), traffic intensity, so as to

investigate the sensitivity of MAD to the network anomaly activity, as the ratio R became small, to almost insignificant levels. The results obtained with other anomalies are similar.

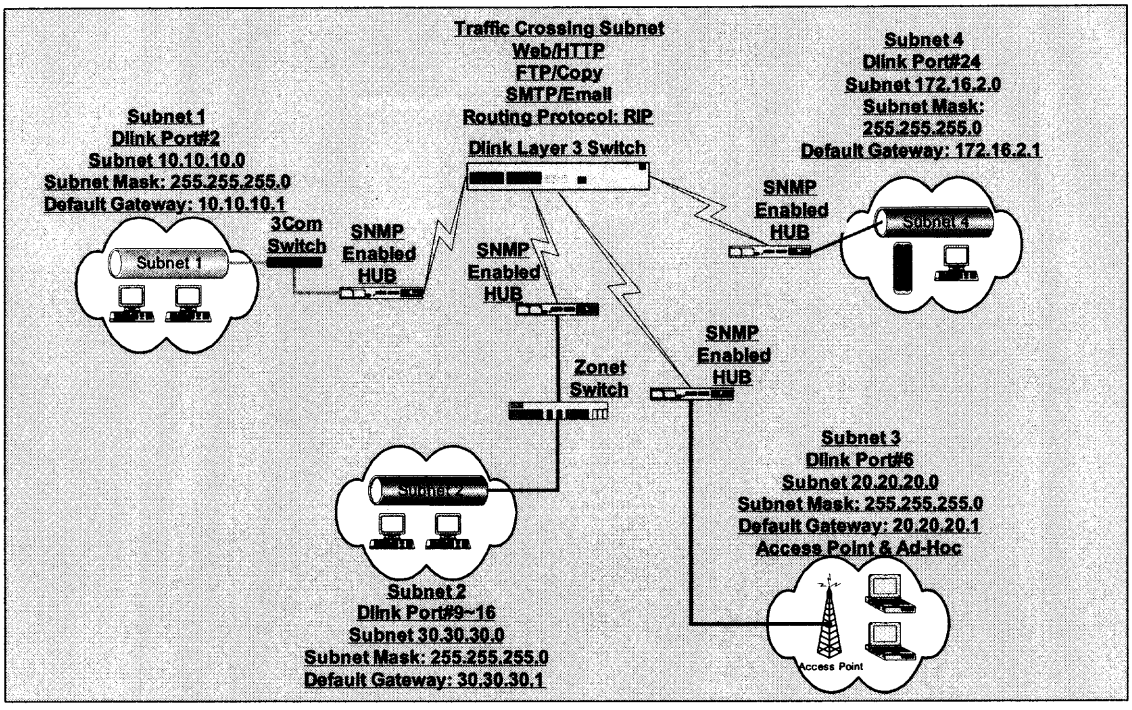

Figure 3.1 Schematic of the testbed network facility.

As mentioned, representative results of nine Ethernet improper short frame error scenarios, with different background and anomaly traffic loads, are presented here, as listed in Table 3.1. For each simulation scenario, network traffic was collected over a duration of 24 hours. The data were recorded, by polling the monitored network entities through sending them SNMP queries from the anomaly detection agent (ADA) periodically, with a period of one second. As listed in the last two columns of Table 3.1, the degree of self similarity (measured by Hurst Parameter [46]), and intensity of Noah effect (measured by the Hill estimator [47]), for the background traffic simulated in the experiments, have been estimated. The estimating results of Hurst parameters indicated that the background traffic simulated in the experiments present medium degree of self-

similarity, or high degree in some cases, such as scenario 8 and 9; and the estimating results of the Hill estimator indicated that the background traffic demonstrated high degree of burst. The background traffic simulated in such way is conductive to investigate the performance of MAD operating in the some extreme worse network environments.

Table 3.1 Experimental Ethernet Improper Short Frame Anomaly Scenario Specification

Scenario Index	Background Traffic (B)	Anomaly Traffic (A)	Ratio R=(A/B)	Hurst Parameter of Background Traffic	Intensity of Noah Effect of Background Traffic
1	320923 bps	32092 bps	10%	0.69	1.73
2	320923 bps	27278 bps	8.5%	0.71	1.69
3	320923 bps	20860 bps	6.5%	0.69	1.78
4	320923 bps	16046 bps	5%	0.72	1.62
5	320923 bps	12837 bps	4%	0.65	1.79
6	320923 bps	9628 bps	3%	0.74	1.58
7	320923 bps	6418 bps	2%	0.68	1.55
8	320923 bps	3209 bps	1%	0.92	1.57
9	320923 bps	1605 bps	0.5%	0.91	1.63

The observation window, over which the variable PDFs were built, was of 64 seconds in duration. Therefore, the total collected data were gathered using 24*3600/64 =1350 observation windows, generating 1350 PDF records for each variable and for each scenario. These data were segmented into two separate sets, one set of 800 records for training and the other of 550 records for testing. The training records included typical as well as anomaly traffic, labeled accordingly. In each scenario, the system was trained for 100 epochs. As mentioned, the system classifier used throughout this study was a backpropagation neural network classifier with 2 hidden neurons.

3.2 Experiment Results and Discussion

The performance of MAD is evaluated based on the *misclassification rates* and the *mean squared root errors (MSR)* of the classifier. The misclassification rate is defined as the percentage of the network traffic that is misclassified by the neural network classifier. The

misclassification rate is the sum of the *false positive* and *false negative* misclassifications. Typically, the *mean squared root errors (MSR)* metric, computed at the output of the classifier, serves as an indicator of convergence of the neural network; it is, of course, directly related to the misclassification rate.

3.2.1 The Results for Investigating the Efficaciousness of the Partition Schemes

The MSR criterion data for the seven partition schemes, UPS, UPPS, LPS, XLPS, XSRPS, YLPS and YSRPS, over the nine anomaly scenarios, are given in Table 3.2. It should be noted that KS3 similarity measurement algorithm is employed in this evaluation experiment. The values are presented after multiplication by a factor of 100 for ease of display. It may be seen that the neural network may be judged as having converged for all seven schemes for scenarios 1 through 8 (R=10 to 1%), while for scenario 9 (R=0.5%) convergence appears to have failed. This assessment is confirmed in Table 3.3 that depicts the misclassification performance, as should be the case due to the direct relationship of MSR to misclassification rate.

The experimental results of the evaluation of the performance of the above mentioned seven partition schemes are listed in Tables 3.3, 3.4 and 3.5. Specifically, Table 3.3 depicts the total misclassification rate, while Tables 3.4 and 3.5 show its component false positive and false negative rates, respectively. From the results, it is found that all of the seven metrics perform well and achieve roughly comparable results. For example, for scenario 6 (R=3%), the total misclassification rate achieved by the UPPS schemes is only 0.32%, with the UPS, LPS, XLPS, XSRPS, YLPS and YSRPS schemes only slightly behind at 0.43%. These are low misclassification rates for such small anomaly ratio values. For the thresholds used in this study at the output of the classifier (s=0), most of the

misclassifications are of the false positive character and not the false negative type; however, this balance could be tipped the other way by a different threshold value choice.

Most importantly, MAD discerns a network anomaly when the anomaly is still in its very early stages, at about 1 percent of the background intensity. This early warning of a developing anomaly is beneficial because it allows countermeasures to be launched by the network administrator before damage occurs.

Table 3.2 The Mean Square Root Error (x100) Neural Network Convergence Criterion in the Emulation of the Partition Scheme

PS* \ Scenario #	1	2	3	4	5	6	7	8	9
UPS	0	0	0	0	1.57	1.24	1.62	1.56	45.23
UPPS	0	0	0	0	1.53	1.25	1.61	1.57	47.41
LPS	0	0	0	0	1.58	1.27	1.64	1.59	49.30
XLPS	0	0	0	0	1.63	1.34	1.69	1.64	58.21
XSRPS	0	0	0	0	1.62	1.32	1.71	1.65	55.64
YLPS	0	0	0	0	1.59	1.31	1.65	1.61	52.23
YSRPS	0	0	0	0	1.58	1.32	1.67	1.60	53.79

*PS stands for Partition Scheme

Table 3.3 The Misclassification Rate in the Emulation of the Partition Scheme (in Percentage)

PS* \ Scenario #	1	2	3	4	5	6	7	8	9
UPS	0	0	0	0	0.43	0.43	0.43	0.43	19.74
UPPS	0	0	0	0	0.43	0.32	0.43	0.53	19.92
LPS	0	0	0	0	0.43	0.43	0.43	0.53	20.31
XLPS	0	0	0	0	0.43	0.43	0.43	0.53	23.12
XSRPS	0	0	0	0	0.43	0.43	0.43	0.53	23.53
YLPS	0	0	0	0	0.43	0.43	0.43	0.53	26.12
YSRPS	0	0	0	0	0.43	0.43	0.43	0.53	27.85

*PS stands for Partition Scheme

Table 3.4 The False Positive Rate in the Emulation of the Partition Scheme (in Percentage)

PS* \ Scenario #	1	2	3	4	5	6	7	8	9
UPS	0	0	0	0	0.43	0.22	0.43	0.43	9.42
UPPS	0	0	0	0	0.43	0.32	0.43	0.53	14.54
LPS	0	0	0	0	0.43	0.43	0.43	0.43	11.89
XLPS	0	0	0	0	0.43	0.43	0.43	0.53	15.92
XSRPS	0	0	0	0	0.43	0.43	0.43	0.53	12.24
YLPS	0	0	0	0	0.43	0.11	0.32	0.53	20.70
YSRPS	0	0	0	0	0.43	0.43	0.43	0.53	14.96

*PS stands for Partition Scheme

Table 3.5 The False Negative Rate in the Emulation of the Partition Scheme (in Percentage)

PS* \ Scenario #	1	2	3	4	5	6	7	8	9
UPS	0	0	0	0	0	0.21	0	0	10.32
UPPS	0	0	0	0	0	0	0	0	5.38
LPS	0	0	0	0	0	0	0	0.10	8.42
XLPS	0	0	0	0	0	0	0	0	7.20
XSRPS	0	0	0	0	0	0	0	0	11.29
YLPS	0	0	0	0	0	0.32	0.11	0	5.42
YSRPS	0	0	0	0	0	0	0	0	12.89

*PS stands for Partition Scheme

3.2.2 The Results for Investigating the Effectiveness of the Similarity Measurement Metrics

The classification results for evaluating the effectiveness of the seventeen similarity measurement metrics are presented in Table 3.6, 3.7, 3.8 and 3.9. Specifically, Table 3.6 lists the MSR criterion data over the nine scenarios; Table 3.7 presents the total misclassification rate and Table 3.8 and 3.9 depict its component false positive and false negative rates, respectively. It should be noted that UPBS partition scheme is employed in these evaluation experiments. From the results, one can make the following conclusions:

A. All of the proposed similarity measurement metrics, except the WSS3, can achieve good performance over the scenarios 1 through 8; while the neural network can not achieve convergence in the case of scenario 9, due to its anomaly traffic is too low to be detected by MAD. The results are consistent with those received from the experiments for evaluating the partition schemes.

B. Comparing the two versions of χ^2 type metrics, it is found out that both of CST1 and CST2 can achieve prefect performance, while CST2 has a slightly lower misclassification rate than CST1.

C. Regarding the versions of KS type metrics, the KS1 and KS3 perform significant better than KS2 and KS4, especially in the cases of scenario 1 through 4.

D. The three versions of weighted similarity statistic (WSS) present far different performance. In comparison of the classification results of WSS1 and WSS3 which applied the weight function designed by ourselves and the weight function borrowed from Anderson-Darling similarity metric respectively, to the same similarity calculation equation, it is observed that the performance of WSS3 is not comparable

with that of WSS1. Also the WSS3 can not achieve stationary performance, which indicates that the weight function borrowed from Anderson-Darling statistic doesn't work consistently with the rest part of WSS3. On the other hand, the WSS1 performs very well. It can achieve 0% misclassification rate over scenario 1 through 4, which means that the weight function designed by us works consistently with the rest part of WSS1. When applying the weight function designed by ourselves to KS2, resulting in WSS2, one can find out that WSS2 works worse than KS2, which indicates that the weight function designed by us doesn't work consistently with KS2.

E. Referring to the two versions of Kupier's KS type statistic, one can find out that the KKS1 achieves better results than KKS2 does over most scenarios. For example, over scenario 1 through 4, KKS1 can achieve 0% misclassification rate, but KKS2's misclassification rates are ranged from 0.32% to 2.37%.

F. In case of two versions of fractional deviation type statistic, the classification results indicate that FDS2 performs slightly better than FDS1.

G. The SNS1, SNS2, FDM and FDMSD are only four versions of similarity measurement metrics studied here that utilized the scalar value, instead of PDF, for their evaluation. Although these metrics has shown similar performance as other PDF based metrics has when the anomaly traffic is low, it performs rather poorly when the anomaly traffic is higher (more than R=5% or so), therefore they are not the best choice here.

Based on the observations above, it is found out that CST1, CST2, KS1, KS3, WSS1, KKS1 and FDS2 achieve roughly comparable classification results and perform significantly better the other metrics. Especially, they achieve 0% misclassification rate over the scenario 1 through 4, which means utilizing these metrics MAD can detect all of the anomalies with no false alarm. Therefore, they can be the best choices for MAD.

Table 3.6 The Mean Square Root Error (x100) Neural Network Convergence Criterion in the Emulation of the Effectiveness of the Similarity Measurement Metrics

SMM* \ Scenario #	1	2	3	4	5	6	7	8	9
CST1	0	0	0	0	1.52	0.96	1.63	1.18	42.57
CST2	0	0	0	0	1.16	1.01	1.57	0.96	19.64
KS1	0	0	0	0	1.46	1.06	1.64	0.77	46.60
KS2	2.05	0.77	0.59	2.95	3.40	4.91	2.19	2.89	33.28
KS3	0	0	0	0	1.57	1.24	1.62	1.56	45.23
KS4	1.22	0.79	6.69	3.58	1.38	1.85	1.59	1.60	37.40
WSS1	0	0	0	0	1.24	1.01	1.62	1.14	39.94
WSS2	5.51	2.57	2.63	3.70	5.77	5.37	4.42	4.55	39.54
WSS3	26.89	32.01	4.92	2.92	45.17	1.27	2.55	31.91	44.65
KKS1	0	0	0	0	1.47	1.10	1.60	1.01	43.45
KKS2	0.91	1.63	6.13	5.28	1.60	2.08	1.22	0.82	41.16
FDS1	1.18	1.01	1.51	1.58	1.55	1.55	1.63	1.57	25.25
FDS2	0	0	0	0	0.86	0.95	1.55	0.61	27.17
SNS1	1.65	1.24	2.53	2.82	1.66	1.64	1.66	1.62	40.00
SNS2	1.99	1.58	0.94	1.33	1.62	1.62	1.65	1.58	46.78
FDM	1.91	2.04	6.36	1.68	1.64	1.53	1.57	1.71	55.36
FDMSD	3.69	3.19	4.47	3.04	6.27	1.62	2.06	6.02	50.49

*SMM stands for Similarity Measurement Metric.

Table 3.7 The Misclassification Rate in the Emulation of the Effectiveness of the Similarity Measurement Metrics (in Percentage)

SMM* \ Scenario #	1	2	3	4	5	6	7	8	9
CST1	0	0	0	0	0.43	0.32	0.43	0.32	19.53
CST2	0	0	0	0	0.32	0.32	0.43	0.32	19.42
KS1	0	0	0	0	0.43	0.32	0.43	0.21	18.46
KS2	0.59	0.30	0.30	0.89	0.89	1.48	0.59	0.90	21.07
KS3	0	0	0	0	0.43	0.43	0.43	0.43	19.74
KS4	0.32	0.43	2.37	1.78	0.43	0.53	0.43	0.43	18.46
WSS1	0	0	0	0	0.32	0.32	0.43	0.32	18.67
WSS2	1.48	0.89	0.89	1.19	1.78	1.48	1.48	1.50	21.07
WSS3	6.72	8.00	2.08	0.89	11.31	0.32	0.64	16.43	32.01
KKS1	0	0	0	0	0.43	0.43	0.43	0.32	19.00
KKS2	0.32	0.53	2.37	1.78	0.43	0.53	0.43	0.43	19.74
FDS1	0.53	0.32	0.59	0.59	0.43	0.42	0.43	0.43	24.55
FDS2	0	0	0	0	0.64	0.32	0.43	0.21	27.11
SNS1	0.43	0.32	0.89	0.89	0.43	0.43	0.43	0.43	19.00
SNS2	0.53	0.43	0.59	0.59	0.43	0.43	0.43	0.43	18.78
FDM	0.53	0.53	2.08	0.59	0.43	0.43	0.43	0.43	19.53
FDMSD	2.03	0.85	2.37	0.89	1.71	0.43	0.53	1.60	19.53

*SMM stands for Similarity Measurement Metric.

Table 3.8 The False Positive Rate in the Emulation of the Effectiveness of the Similarity Measurement Metrics (in Percentage)

SMM* \ Scenario #	1	2	3	4	5	6	7	8	9
CST1	0	0	0	0	0.43	0.32	0.43	0.32	9.18
CST2	0	0	0	0	0.32	0.32	0.43	0.32	9.61
KS1	0	0	0	0	0.43	0.21	0.43	0.21	8.43
KS2	0.59	0.30	0	0.89	0.89	1.48	0.59	0.90	12.46
KS3	0	0	0	0	0.43	0.43	0.43	0.43	8.96
KS4	0.32	0.43	2.37	1.78	0.43	0.53	0.43	0.43	8.43
WSS1	0	0	0	0	0.32	0.32	0.43	0.32	8.43
WSS2	1.48	0.59	0.89	0.89	1.78	1.48	1.48	1.45	13.06
WSS3	0	0	1.78	0.89	0.21	0.21	0.43	9.18	23.91
KKS1	0	0	0	0	0.43	0.43	0.43	0.32	7.90
KKS2	0.32	0.53	2.37	1.78	0.43	0.53	0.43	0.43	8.96
FDS1	0.53	0.32	0.30	0.59	0.43	0.43	0.43	0.43	17.29
FDS2	0	0	0	0	0.64	0.21	0.43	0.21	11.63
SNS1	0.43	0.32	0.89	0.89	0.43	0.43	0.43	0.43	8.64
SNS2	0.53	0.43	0	0.59	0.43	0.43	0.43	0.43	8.64
FDM	0.53	0.53	2.08	0.30	0.43	0.43	0.43	0.43	9.28
FDMSD	0.43	0.85	2.37	0.89	1.07	0.43	0.53	1.60	8.96

*SMM stands for Similarity Measurement Metric.

Table 3.9 The False Negative Rate in the Emulation of the Effectiveness of the Similarity Measurement Metrics (in percentage)

SMM* \ Scenario #	1	2	3	4	5	6	7	8	9
CST1	0	0	0	0	0	0	0	0	10.35
CST2	0	0	0	0	0	0	0	0	9.82
KS1	0	0	0	0	0	0.11	0	0	10.03
KS2	0	0	0.30	0	0	0	0	0	8.61
KS3	0	0	0	0	0	0	0	0	10.78
KS4	0	0	0	0	0	0	0	0	10.03
WSS1	0	0	0	0	0	0	0	0	10.25
WSS2	0	0.30	0	0.30	0	0	0	0	8.01
WSS3	6.72	8.00	0.30	0	11.10	0.11	0.21	7.26	8.11
KKS1	0	0	0	0	0	0	0	0	11.10
KKS2	0	0	0	0	0	0	0	0	10.78
FDS1	0	0	0.30	0	0	0	0	0	7.26
FDS2	0	0	0	0	0	0.11	0	0	15.47
SNS1	0	0	0	0	0	0	0	0	10.35
SNS2	0	0	0.59	0	0	0	0	0	10.14
FDM	0	0	0	0.30	0	0	0	0	10.25
FDMSD	1.60	0	0	0	0.64	0	0	0	10.57

*SMM stands for Similarity Measurement Metric

CHAPTER 4

THE GRAFTED AND RE-USE CLASSIFIER TRAINING METHODS

4.1 Methodology

In the previous chapters, a hierarchical, multi-tier, multi-observation-window, statistical based network anomaly detection system have been prototyped, namely the MIB Anomaly Detection (MAD) system. In installing and operating MAD, while both normal and anomaly data may be available in a test network, only normal data may be routinely available in a production network, thus MAD may be ill-trained for the unfamiliar network environment. In this section, two approaches for adequately training the neural network classifier in the target network environment are presented in detail, namely the re-use and the grafted classifier methods. The re-use classifier method is better suited when the target network environment is fairly similar to the test network environment, while the grafted method can also be applied when the target network may be significantly different from the test network.

In a production network, during its intended operation, anomaly detection is expected to find deviant activity. The classifier processes the current pattern of activity and rates it with a similarity score as to whether it is more similar to normal or anomalous activity. The MAD classifier has employed supervised training that utilized labeled data collected during normal and anomalous activity periods in some test network. The MAD classifier, like most such classifiers, requires data records of the typical (normal) and anomaly-labeled variety in sufficiently large amounts. Moreover, if one desires to distinguish between different classes of anomalies or perhaps individual anomalies, data for all such different anomalies need to be made available for training. Such data will

enable the neural net classifier to learn the difference between normal and anomalous patterns of activities, thus achieving high detection and low false alarm rates. In a test or laboratory network, the experimenter or developer can investigate and record the launching of all desired known anomalies selecting from a library of anomalies. Such investigations of anomalies can be carried out and recorded for various background traffic types and levels.

The challenge that is faced is that while both normal and anomaly data are easily and conveniently available for a test network, only normal data are routinely available for a production network. The classifier of the network anomaly detection system will probably be ill-trained for the new network environment and thus unsuitable for use without changes if (1) the unfamiliar network is sufficiently different from the test network, so the new background will not be close enough to that of the test network, or (2) it lacks training that includes anomaly data for the unfamiliar network. One or the other, or more likely both conditions hold in most realistic cases of installations of new NADS and thus the NADS classifier will need to be re-trained at some stage during or soon after installation.

A network anomaly detection tool that is to be installed in an unfamiliar real network environment can reasonably expect some initialization period that will provide sufficient estimates of the normal behavior of the network. In a typical setting, by far most of the traffic that the NADS sees is normal, with only a sprinkling of anomaly traffic. Thus, algorithms could conceivably be used to separate out the normal traffic from any anomalies, whenever they might occur in the unfamiliar network, based on intensity comparison considerations alone. Such algorithms, employing clustering techniques, have been investigated elsewhere [48].

Additionally, it may be that during the initialization phase, all the traffic in the new setting is in fact normal. Even if some anomaly traffic is found in the unfamiliar network, it may be difficult, at this stage, to characterize what type of anomaly it is. This leaves us with an estimate of normal traffic in the unfamiliar network setting, which is possible to carry out, but no clear description of network anomalies. Although theoretically possible, it would be unreasonable and unsafe to launch anomalies in the production network to be protected, solely for the benefit of the learning phase of the NADS. Most likely it would not be permitted to take place. Thus, somehow, while using the normal and anomaly data from a test network, as well as the normal data from the production network, it is needed to ensure that a properly trained classifier is generated.

In this dissertation, two alternative solution approaches to this challenge are proposed, the *re-use* classifier and the *grafted* classifier:

- The re-use classifier method is straightforward and consists of employing the test-network classifier in the new network, as is after training only in the test network that is without any modifications at the beginning. This approach is expected to perform adequately if the unfamiliar network is similar in architecture and usage to the test network, but will likely increasingly deteriorate in performance the more dissimilar the two settings are.
- The grafted classifier method essentially consists of abstracting an estimate of the anomaly model, in "pure" form, from the test network and then "grafting" this anomaly onto the normal model in the new network, thus "transplanting" the anomaly model from the test network to the production network.

In this way, the grafted classifier method will provide anomaly models as well as normal models in the unfamiliar network setting, thus enabling the training of the classifier in this new network. Simulation measurements have been used to investigate the effectiveness of these techniques in an experimental setting, where both have been found to be effective. The grafted classifier approach will work adequately for the class of

parameters where the effect of the anomaly is of "additive" character to that of the background.

Nevertheless, it should be noted that the expectation regarding these techniques is not that they generate the final best model of an anomaly, but that they will provide a good "seed" of such a model, allowing the system to adapt the model by "learning" from then on, thus bootstrapping onto an accurate anomaly model from an initially only adequate estimate. MAD employs an adapting algorithm that adjusts the normal as well as the anomaly models, by using the descriptions of the current normal and anomaly data, after they are detected to be so.

The grafted classifier approach described above, may, in general, depend on the nature of the monitored parameter as well as the type of anomaly. Basically, it relies on the normal background model to vary most from network to network while the anomaly models vary least. This is intuitively reasonable and has been observed to be the case for many types of networks and anomalies. The normal background network traffic changes in time and from network to network. On the other hand, the behavioral profiles of various kinds of anomalies are comparatively stationary and exhibit similarities from network to network.

In more detail, this approach involves extracting anomaly profiles, for each monitored parameter, from a test network and then applying them to a new production network. First, the models of anomalies and background traffic from the data collected in a test network are extracted. Second, the anomaly profiles are calculated by removing the components of normal traffic from the anomaly models. Third, the anomaly models for the new network can be estimated by combining and merging the models of normal traffic of

the real network with the models of anomalies that have been extracted from the test network. Finally, the anomaly traffic itself can be simulated according to the estimated anomaly models, thus enabling the NADS classifier to be trained using the data thus generated.

Before describing the details of the grafted classifier method, it is desired to provide and differentiate the definitions of anomaly models and anomaly profiles:

- *Anomaly model* is the statistical representation of the network traffic pattern when both the background traffic and the anomaly traffic are observed in a network.
- *Anomaly profile* is the statistical representation of the network traffic pattern when only the anomaly traffic is observed in a network.

The grafted classifier method operates through the following five steps. The details of the algorithms are clearly illustrated in Figure 4.1.

Step 1. The reference models of both the normal traffic and the anomaly traffic are calculated from the test network through statistically averaging the normal and anomaly PDF data records. Afterward, the test models of normal background traffic will be called M_{B_T} and the test anomaly traffic will be named M_{A_T} for simplicity.

Step 2. The anomaly profiles are generated. The reference models M_{A_T} contain both anomaly traffic and background traffic, because both kinds of traffic may be observed within a time window. This is the reason why the anomaly models collected in the test network cannot be applied directly to an unfamiliar production network. The profiles of anomalies P_A are calculated by removing the components of the background traffic M_{B_T} from the anomaly models M_{A_T}. The equations to calculate the profiles of the pure anomalies are given below:

$$P_A(i) = \frac{N_A \times M_{A_T}(i) - N_B \times M_{B_T}(i)}{N_A - N_B} \tag{4.1}$$

$$N_P = N_A - N_B$$

- i: The i^{th} bin of PDF.
- P_A: The profile of the particular anomaly traffic at hand.
- M_{A_T} : The reference model of the particular anomaly traffic at hand.
- M_{B_T} : The model of the normal background traffic.
- N_A: The number of samples comprising M_{A_T} .
- N_B: The number of samples comprising M_{B_T} .
- N_P: The number of samples comprising P_A.

The basic idea here is that an estimate of the change that the anomalies impose to the normal traffic PDF models, for some and perhaps most parameters, can be inferred by subtracting out the contributions of the normal model from the anomaly model, bin-by-bin. Even if this works well for only a few of the parameters, the combining power of the classifier assures that the classification results will be better than those of any single parameter by itself. This is sufficient to seed correctly the MAD system into its learning mode, so that it improves its models using actual data, from then on. The experiments suggest that many parameters can in fact be modeled correctly in this manner.

Step 3. The reference models of the background traffic in the unfamiliar production network are calculated. In this step, the models of background traffic in the production network, M_{B_P}, are calculated for use in the anomaly model estimation, subsequently.

Step 4. The anomaly models for an unfamiliar production network can be estimated as the sum of the model of the background traffic in the production network (M_{B_P}) and the anomaly profiles derived from the test network (P_A). The equations of estimation are:

$$M_{A_P}(i) = \frac{N_B * M_{B_P}(i) + N_P * P_A(i)}{N_B + N_P} \qquad (4.2)$$

$$N_A = N_B + N_P$$

where:

- i: The i^{th} bin of PDF.
- M_{A_P} : The estimated reference anomaly model in the unfamiliar production network.
- N_A: The number of samples comprising the estimated anomaly model M_{A_P}.

- M_{B_P} : The measured model of the background traffic in the production network.
- N_B : The number of samples comprising M_{B_P} .
- P_A : The anomaly profile derived from the test network.
- N_P : The number of samples comprising P_A .

Step 5. At this point, an estimate of the anomaly model in the unfamiliar production network has been computed. Then anomaly PDFs can be generated as the sum of the estimated anomaly models and the background traffic collected from the production network. The simulated anomaly PDFs will be used to train the neural net classifier so as to let it adapt to the production network environment. The equations are:

$$K_{A,p}(i,t) = \frac{N_{K_{B,P}}(t) * K_{B,p}(i,t) + N_P * P_A(i)}{N_{K_{B,p}}(t) + N_P} \tag{4.3}$$

$$N_{K_{A,p}}(t) = N_{K_{B,p}}(t) + N_P$$

where:

- i: The i^{th} bin of the PDF.
- t: The time window
- $K_{A,p}(t)$: The simulated anomaly traffic in the production network.
- $K_{B,p}(t)$: The measured background traffic in the production network at time window t.
- P_A : The extracted anomaly profile.
- $N_{K_{A,p}}(t)$: The number of samples of $K_{A,p}(t)$.
- $N_{K_{B,p}}(t)$: The number of samples of $K_{B,p}(t)$.

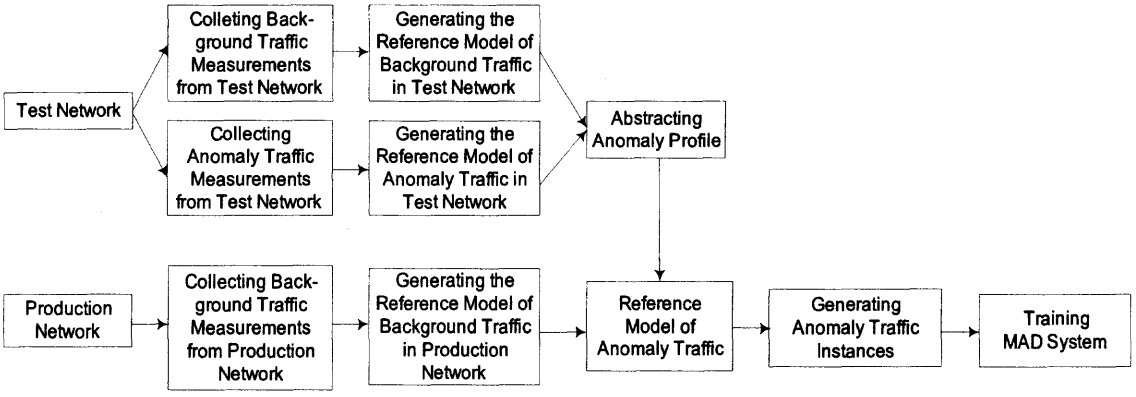

Figure 4.1. The grafted classifier method.

4.2 Evaluation Process

In order to investigate the performance of the re-use and grafted classifiers methods, extensive simulation experiments were carried out, under various conceivable production network environments, i.e., variations in network topology, typical background traffic and anomaly traffic characteristics and load.

The rest of this section is organized as follows: Section 4.2.1 present the test and production network model architectures and the corresponding background network traffic configurations used in the various simulation scenarios. Numerical results and additional discussions regarding the effectiveness of these two classifier methods are presented in Section 4.2.2.

4.2.1 The Testbed Configuration

The test network was built using the Optimized Network Engineering (OPNET) tool, as shown in Figure 4.2. In this network model, there are 3 servers, namely, Main Server (responsible for file access and mail service), HTTP Server, and TELNET Server, as well as 11 clients, connected by a hub.

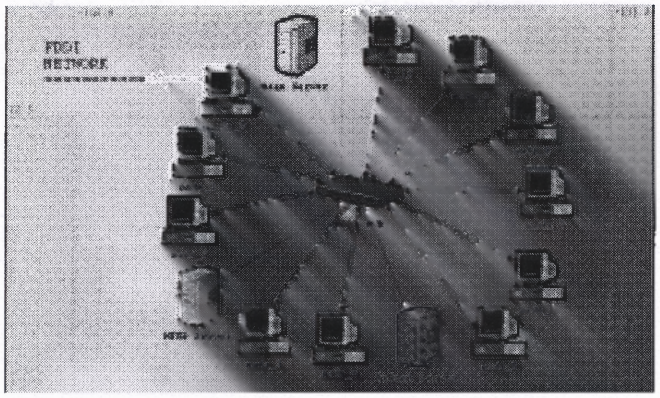

Figure 4.2 The test network.

The production network model, shown in Figure 4.3, consists of four subnetworks connected by four routers. The links that connect the network components are assumed to be T1 links. The clients are located in all the four subnetworks as follows: Subnet_1 (Ethernet Network), Subnet_2 (Token Ring Network), Subnet_3 (Fast Ethernet Network) and Subnet_Server (FDDI Network). The clients located in each of the four subnetworks communicate with the server located in Subnet_Server.

In the simulation experiments, in order to comparatively evaluate the performance of the two proposed classifier methods under the variation of the production network topology, the number of the clients located in each subnet, which establish the communication with the server located in the Subnet_Server, was changed. Three simulation scenarios are investigated, i.e., "10 clients", "15 Clients" and "20 Clients".

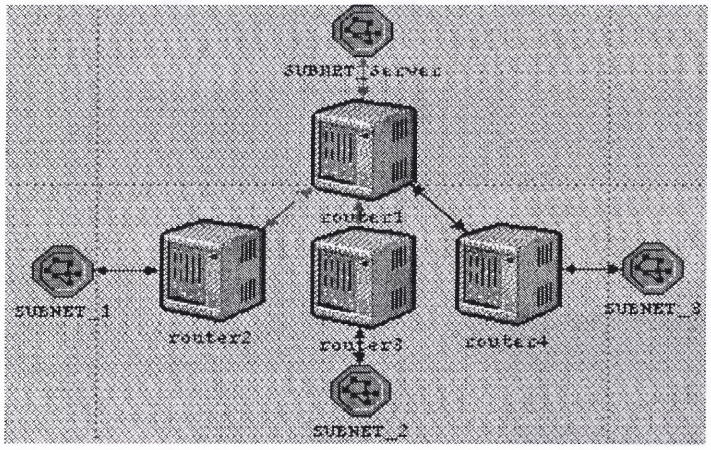

Figure 4.3 The production network.

In the simulation experiments, the four most popular TCP/IP services were modeled: HTTP (TCP), TELNET (TCP), FTP (TCP) and SMTP (UDP). The application-layer workload characteristics of these four Internet services, as used in the simulations, derived from the literature, as in [49-52]; such work indicates that the workload characteristics utilized closely resemble network conditions.

In this experiment, the network anomaly conditions were simulated by injecting various volumes of TCP anomaly traffic into the simulation testbed, i.e. the TCP small packet flooding anomaly traffic. Specifically, in the test network, a client (workstation) is assumed to generate and send TCP flooding packets to the HTTP server periodically, and in the production network, a client located in the Subnet_1 will send the TCP flooding anomaly packets to the server located in Subnet_Server. As presented in detail in Section 4.2.2, various experiments were performed and the performance of *re-use* and *grafted* classifier methods under many different scenarios was tested, where several characteristics regarding the network topology and the anomalous traffic and background traffic were varied, such as the number of clients located in each subnet, and the background and anomalous traffic patterns.

4.2.2 Numerical Results and Discussion

In this section, the numerical results for the various simulation experiments described in the previous sections are presented. The objective of the results presented, in the remainder of this section, is to evaluate the effectiveness of the re-use and grafted classifier methods, and demonstrate that the grafted method performs significantly better than the re-use method in most simulation scenarios.

As mentioned, extensive simulation experiments have been carried out. The purpose of these simulation experiments is to investigate the performance of *re-use* and *grafted* classifier methods, when the network topology and the background and anomaly traffic of the production network vary. The production network configurations, for these simulation experiments, are given in Table 4.1.

Three scenarios are carried out. In each scenario, the data sets collected from the test network (the test network configuration presented in Table 4.2) are used as training data for training the test classifier. This test classifier is used in the investigation of the efficacy of the *re-use* classifier. This data are also used for the calculation of the anomaly profiles, by using the algorithms described earlier that are part of the grafted classifier method. The data sets collected from the production network are used to estimate anomaly models and to simulate anomaly data for that environment. The classification rates of MAD using the simulated data are compared with those using the actual data, to carry out performance comparisons.

For each simulation scenario, three different experiments are performed to examine the classification performance outcomes of MAD for the production networks at hand for the two approaches, the re-use classifier and the grafted classifier.

1. The actual PDF data collected from the production network are used for the training and testing/validation of the MAD classifier. The results are used as the base line of the system performance, in that they should be the best that can be achieved for the actual production network. For simplicity, this experiment is called as the *baseline* for experimental outcomes.

2. The generated PDF data, using the grafted classifier method, are used for training the MAD classifier, while the actual data are used for testing. The results reflect the performance of this method and is referred to as the *grafted* method experiment.

3. The PDF data collected from the test network are used for both training and testing of the MAD classifier. Subsequently, the classifier is used to validate the production network setting PDF data. Thus, this corresponds to examining the efficacy of the *re-use* method, to be referred to as the *re-use* method experiment.

Table 4.1 The Production Network Configuration

Scenario1	Network topology	# of clients in Subnet 1	3
		# of clients in Subnet 2	3
		# of clients in Subnet 3	3
		# of clients in Subnet Server	1
		Total # of clients	10
	Background Traffic	213 k bps*	
	Anomaly Traffic	4.5 kbps	
Scenario2	Network Topology	# of clients in Subnet 1	4
		# of clients in Subnet 2	4
		# of clients in Subnet 3	4
		# of clients in Subnet Server	3
		Total # of clients	15
	Background Traffic	320 kbps*	
	Anomaly Traffic	4.86 kbps	
Scenario3	Network Topology	# of clients in Subnet 1	5
		# of clients in Subnet 2	5
		# of clients in Subnet 3	5
		# of clients in Subnet Server	5
		Total # of clients	20
	Background Traffic	427 kbps*	
	Anomaly Traffic	6.48 kbps	

*The combination of the background traffic in the production network is the same for the three simulation scenarios, which is HTTP: 50%, FTP: 20%, SMTP: 20%, TELNET: 10%, but different from that of the test network.

Table 4.2 The Test Simulation Network Configuration

Network Topology	11 clients connected
Background Traffic	Total Traffic in bits/sec: 234k Combination: Http: 65%, FTP: 15%, SMTP: 15% and TELNET: 5%
Anomaly Traffic	Total Traffic in bits/sec: 7k

Table 4.3 The Misclassification Rate of the Simulation Experiments

	Baseline	Grafted	Re-use
Scenario 1	0.0117	0.0474	0.0864
Scenario 2	0.0160	0.0871	0.2483
Scenario 3	0.0127	0.0983	0.1665

The misclassification rates of MAD, for the baseline, grafted classifier and re-use classifier experiments, are listed in Table 4.3. The Receiver Operating Characteristic (ROC) diagrams are presented in Figure 4.4, where the x-axis is the false alarm rate and the y-axis is the detection rate. The false alarm rate is the rate of the typical traffic events being classified as anomalies, while the detection rate is calculated as the ratio between the number of correctly detected anomalies to their total number. The values seen in the table follow what was observed from Figure 4.4.

From the simulation experiments, one can observe that the misclassification rates of the grafted experimental measurements perform significantly better than the re-use classifier experiments. This may be due to the significant dissimilarity between the test network configuration and the production network configuration. In fact, the grafted classifier performance is close to that found for the baseline experiments; the results of the latter are, of course, the best that can be achieved for the production network. This indicates that the grafted modeling and estimation algorithms can be used as effective starting points for the case which the network topology of the production network is significantly different from the topology of the test network, when migrating an anomaly NADS from the test network to an network setting, where accurate anomaly models are not available.

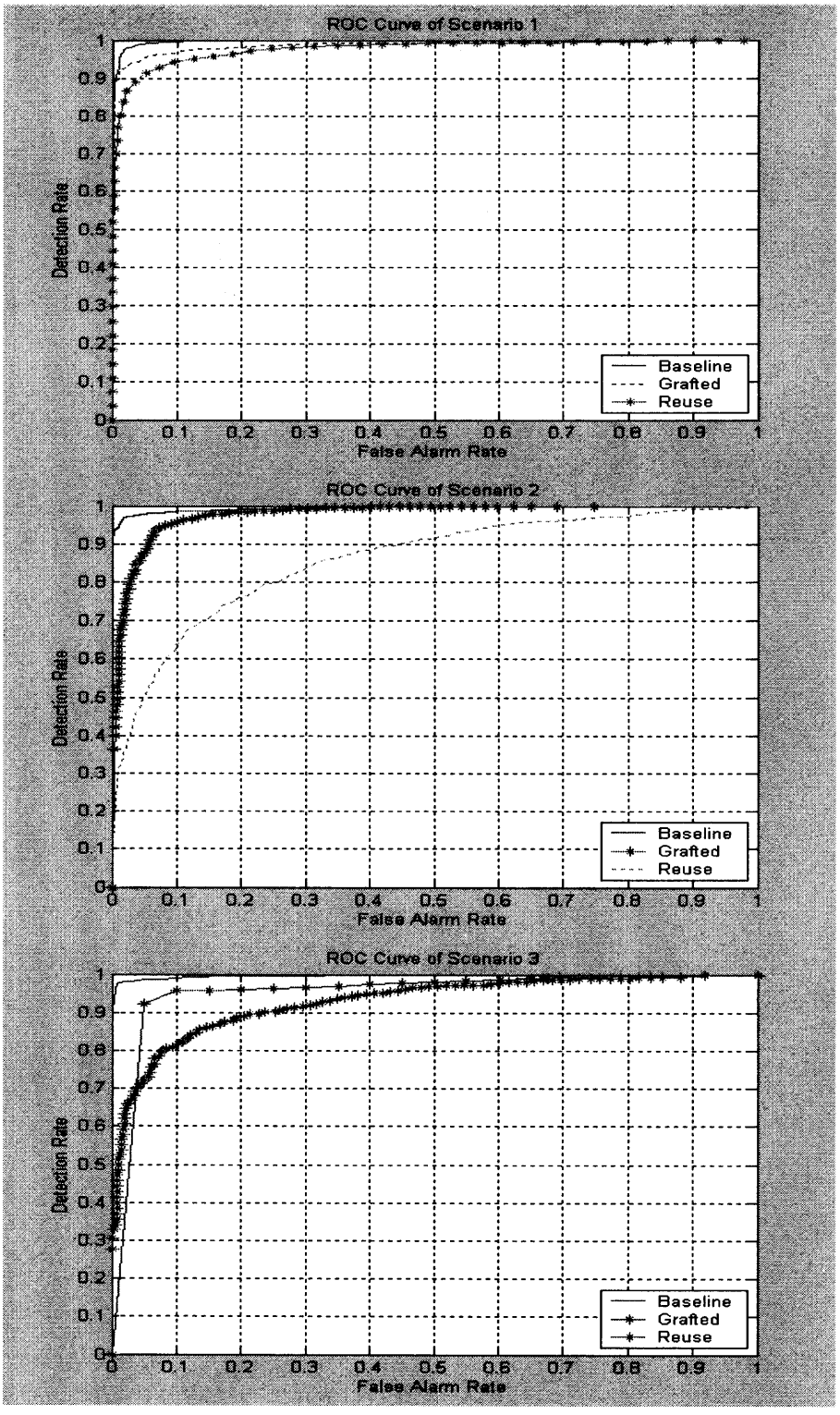

Figure 4.4 The ROC curves.

CHAPTER 5

THE APPLICATION OF LOW PASS FILTERS IN NADS DESIGN

A common method of identifying network anomaly with statistical based NADS is to detect significant deviations in network traffic compared to normal conditions. Such changes may include unexpected high traffic volume, caused by e.g., a packet storm or Denial of Service (DoS) attack. However, recent research on traffic engineering has demonstrated that modern data network traffic exhibits high burstiness at a wide range of observation window sizes. Moreover, these bursts can not be smoothed by simply increasing the observation time window, which is in contrary to those encountered in the traditional telephony network. This effect is described statistically as long-range dependence (LRD), and the time series description showing this effect is said to be self-similarity [29]. The challenge here is that in a network environment with a traffic pattern of high burst rate and self-similarity, a statistical based network anomaly detection scheme may wrongly identify network traffic burst as network anomalies, thus suffering from high false alarm rate. Thus, to achieve high anomaly detection capability while maintaining low false alarm rate, the statistical based NADS should recognize the different characteristics between network bursts and network anomalies.

Through analyzing network traffic measurements containing traffic bursts and network anomalies collected from various network environments, it is found out that the main difference between network anomalies and traffic bursts lies in the duration of high traffic volume caused by them; the former always appears as high traffic volume with a

long period until it is corrected by the network administrator, while the latter always presents itself only as a high traffic volume lasting for a very short period.

Figure 5.1 illustrates a typical example of network traffic measurements collected from the research testbed (details of the configuration of the testbed presented in Chapter 3) when ethernet improper short frame errors are launched. From this example, one can clearly observe the different characteristics of traffic bursts and network anomalies; that is, traffic bursts typically appear as a sprinkling of high traffic volume instances of short duration, scattered apart from one another, while network anomalies appear as cases of high traffic rate of long duration. However, since both traffic bursts and network anomalies appear as high traffic volume occasions, statistical based NADS may have great difficulty in distinguishing one from the other, thus generating false alarms. Hence, to diminish the false alarms caused by traffic bursts, one should find out a method to reduce the degree and frequency of traffic bursts. Since traffic bursts appearing as a smattering of high traffic volume, one can consider them as *noise* in the numerical processing, and utilize a low pass filter to smooth the noise.

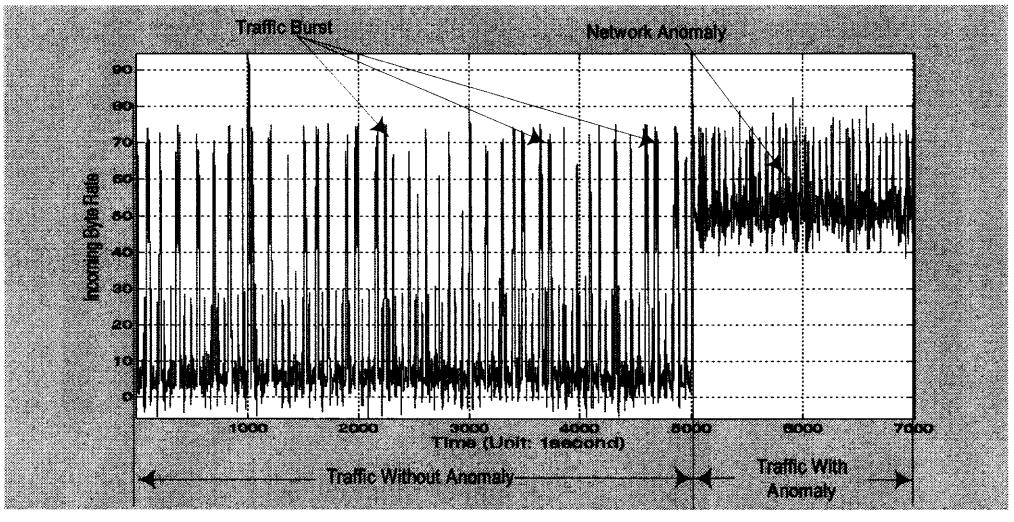

Figure 5.1 Comparing the different characteristics of traffic burst and network anomaly.

In the design of MAD, a low pass filter is introduced to smooth the burstiness in network traffic measurements, so as to reduce its false alarm rate. Seven types of low pass filters have been investigated, including the moving window averaging (MWA) filter, the Savizky-Golay filter [53], and four variants of 4^{th} order Butterworth filters with different cutoff frequencies [54]. Choosing the optimal filter in network anomaly detection should be carried out in accordance with two criteria:

1) The filter should not significantly change the statistical properties of the network traffic measurements, upon which the network anomaly detection scheme relies to identify network anomalies, and

2) The filter should improve the false alarm rate of statistical based NADS, by reducing the bursts encountered in network traffic measurements.

Through statistically analyzing the traffic measurements processed by the low pass filters and further testing their performance by applying them to MAD, it was found out that the Savizky-Golay filter and the 4^{th} order Butterworth filter with cutoff frequency 0.2 are effective choices, and could be the preferred choices of all, according to the criteria mentioned above.

In the rest of this Section, the low pass filters investigated in this dissertation will be discussed. Specifically, in Section 5.1 the Moving Window Average filter and Savitzky-Golay filter will be introduced, while Section 5.2 will present the Butterworth filter. In Section 5.3, the effect of the low pass filters on alternating the statistical properties of network traffic measurements will be compared. And in Section 5.4, the effectiveness of the low pass filters on reducing the false alarm rate of NADS is investigated through applying the filters to the MAD system.

5.1 Comparing Savitzky-Golay Filter with Moving Window Average Filter

Since the *Savitzky-Golay* filter (also called least-squares [55], or DISPO (Digital Smoothing Polynomial) [56] filters) was first introduced by Savitzky and Golay (1964) [57], it has been used to analyze data in a number of scientific areas. Rather than having their properties defined in the Fourier domain, and then translated to the time domain, Savitzky-Golay filters derive directly from a particular formulation of the data smoothing problem in the time domain.

It is well known that a digital filter is applied to a series of equally spaced data values $f_i \equiv f(t_i)$, where $t_i = t_0 + i\Delta$ for some constant sample spacing Δ and $i = \ldots -2,-1,0,1,2\ldots$. The simplest type of digital filter (the nonrecursive or finite impulse response (FIR) filter) replaces each data value f_i by a linear combination g_i of itself and some number of nearby neighbors,

$$g_i = \sum_{n_L}^{n_R} c_n f_{i+n} \tag{5.1}$$

where n_L is the number of points used "to the left" of a data point i, i.e., earlier than it, while n_R is the number used to the right, i.e., later. A so-called *causal* filter would have $n_L = 0$.

The simplest low pass filter computes each g_i as the average of the data points from f_{i-n_L} to f_{i+n_R} for some fixed $n_L = n_R$. This is sometimes called *Moving Window Averaging (MWA)* filter and corresponds to Equation 5.1 with constant $c_n = 1/(n_L + n_R + 1)$. If the underlying function is constant, or is changing linearly with

time (increasing or decreasing), then no bias is introduced into the result. Higher points at one end of the averaging interval are, on average, balanced by lower points at the other end. A bias is introduced, however, if the underlying function has a nonzero second derivative. At a local maximum, for example, MWA always reduces the function value.

The main concept, behind the Savitzky-Golay low pass filtering, is to approximate the underlying function within the moving window not by a constant (whose estimate is the average), but by a polynomial of higher order, typically quadratic or quartic: For each point f_i, the Savitzky-Golay filter fits a polynomial to all $n_L + n_R + 1$ points in the moving window by the least square algorithm, and then sets g_i to be the value of that polynomial at position i.

To derive the coefficients in Equation 5.1, one first needs to consider how g_0 might be obtained. One may want to fit the polynomial of degree M in i, e.g., $a_0 + a_1 i + a_2 i^2 + ... + a_M i^M$ to the values $f_{-n_L} ... f_0 ... f_{-n_R}$ at position 0. Thus, the value of g_0 will be set to the value of that polynomial at $i = 0$, e.g., a_0. The design of the matrix that is used for the least square fitting is

$$A_{i,j} = i^j \tag{5.2}$$

where $i = -n_L \cdots n_R$ and $j = 0 \cdots M$.

Then least square fitting of $f_{-n_L} ... f_0 ... f_{-n_R}$ to a polynomial, presented in matrix notation will be:

$$\left(A^T \cdot A\right) \cdot a = A^T \cdot f \text{ or equivalently } a = \left(A^T \cdot A\right)^{-1} \cdot \left(A^T \cdot f\right) \tag{5.3}$$

In Equation 5.3, the components $(A^T \cdot A)^{-1}$ and $(A^T \cdot f)$ also can be presented as the following forms:

$$\left(\{A^T \cdot A\}_{i,j}\right)^{-1} = \left(\sum_{-n_L}^{n_R} A_{ki} A_{kj}\right)^{-1} = \left(\sum_{-n_L}^{n_R} k^{i+j}\right)^{-1} \tag{5.4}$$

and

$$\left(A^T \cdot f\right)_j = \sum_{-n_L}^{n_R} A_{kj} f_j = \sum_{-n_L}^{n_R} k^j f_j \tag{5.5}$$

Recall that g_0 will be set to the value of the polynomial at position 0 if one wants to least-squares fit the values of $f_{-n_L} \ldots f_0 \ldots f_{-n_R}$ to the polynomial at that position. Thus the coefficient c_n will be the component of a_0 if f is replaced by the unit vector e_n, $-n_L < n < n_R$. Finally, the coefficient c_n will be:

$$c_n = \left\{\left(A^T \cdot A\right)^{-1} \cdot \left(A^T \cdot e_n\right)\right\}_0 = \sum_{m=0}^{M} \left\{\left(A^T \cdot A\right)^{-1}\right\}_{0m} \cdot n^m \tag{5.6}$$

The main advantage of the Savitzky-Golay low pass filtering over other low pass filtering methodology is that it can filter out the noise (e.g., unreasonable burstiness) in the data, while preserving its high statistic moments, thus keeping its statistical properties unchanged.

5.2 The Butterworth Filter

Butterworth filter is an infinite impulse response (IIR) filter [54, 58]. It is well known by its Butterworth or Maximum-flat response. It exhibits a nearly flat low passband with no ripple, and the rolloff is smooth and monotonic. The general equation for a Butterworth low pass filter's frequency response is given below:

$$\overline{B(w)}B(w) = \frac{1}{1 + (w/w_o)^{2n}} \tag{5.7}$$

where n is the order of the filter, and can be any positive integer number (1, 2, 3 ...), and w is the cutoff frequency, i.e., the -3dB frequency of the filter.

Figure 5.2 presents the frequency responses of the 2nd, 4th and 8th order Butterworth low pass filters. As shown in this figure, the Butterworth low-pass filter does not completely pass the frequency components lower than the cutoff frequency, nor completely stops those higher than the cutoff frequency. As the filter order increases, the transition from the pass band to the stop band gets steeper. In the meantime, the computation complexity for implementing the filter also increases exponentially. Thus, considering online operation of network anomaly detection system, in the design of MAD the 4th order Butterworth low pass filter was considered.

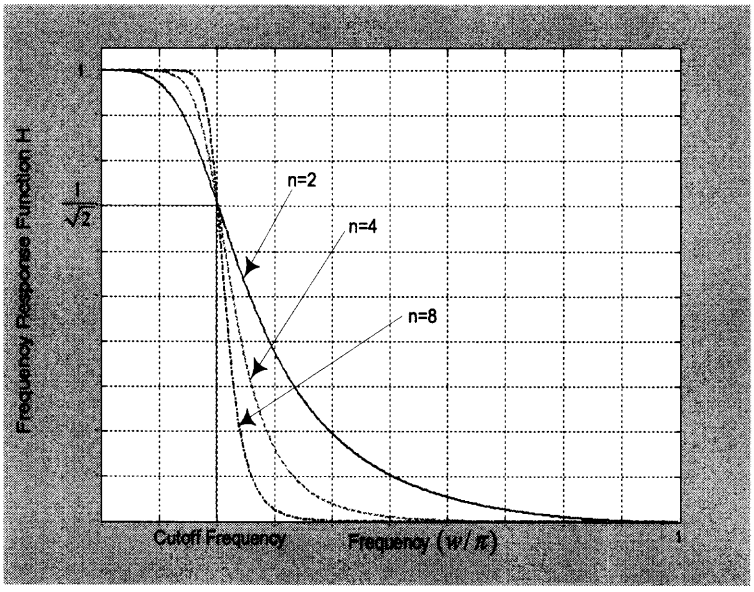

Figure 5.2 The frequency responses of the 2nd, 4th, and 8th order Butterworth low pass filters.

5.3 Do the Low Pass Filters Alternate the Statistical Properties of Network Traffic Measurements?

Recall that choosing the optical low pass filter applied in statistical based NADS should accord with two criteria: 1) the filter shouldn't significantly change the statistical properties of network traffic measurements, and 2) the filter should improve the false alarm rate of statistical based NADS by reducing the bursts encountered in network traffic measurements. In this section, the effect of the low pass filters on the statistical properties of network traffic measurements will be investigated. In the next section, the effect of these low pass filters on reducing the false alarm rate of statistical based NADS will be studied, through applying them to MAD system.

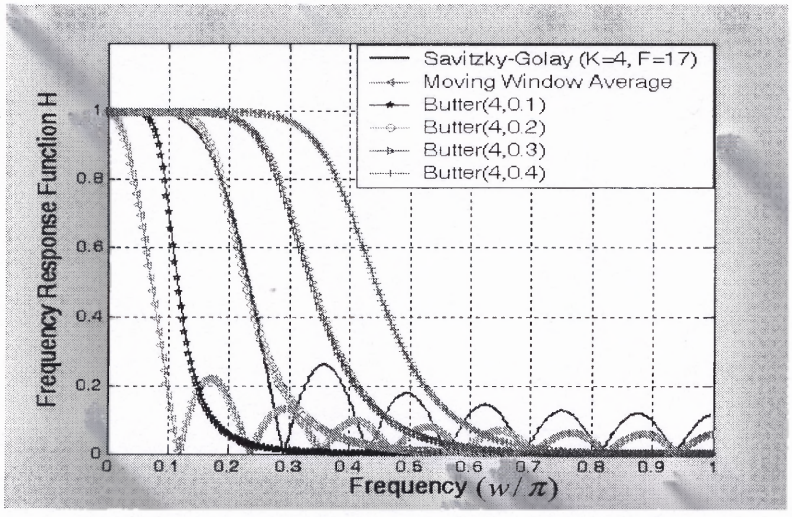

Figure 5.3 The frequency responses of the low pass filters under investigation.

Figure 5.3 presents the frequency response of the seven types of low pass filters studied in this work, i.e., the Moving Window Average (MWA) filter, Savitzky-Golay filter and four variants of the 4^{th} order Butterworth filters with different cutoff frequencies 0.1, 0.2, 0.3 and 0.4 Hz. From now on, the four variants of Butterworth filters will be noticed as Butterworth (0.1) filter, Butterworth (0.2) filter, Butterworth (0.3) filter and

Butterworth (0.4) filter according to their corresponding cutoff frequencies. From Figure 5.3, one can see that the spectrums of these low pass filters are significantly different from one to another. The MWA filter has very narrow passband, and thus may eliminate most frequency components. Though it removes the bursts from the network traffic measurements which normally appear as the high frequency components; it may also strain some statistical information, which reflects in the medium or low frequency components, out of the network traffic measurements, which is crucial for statistical based NADS to identify network anomaly. One can observe that the passband of the Butterworth filter expands when the cutoff frequency increases. It is also interesting to observe that the Savitzky-Golay and Butterworth (0.2) filters have similar spectrum except the ringing in the high frequency band of the Savitzky-Golay filter.

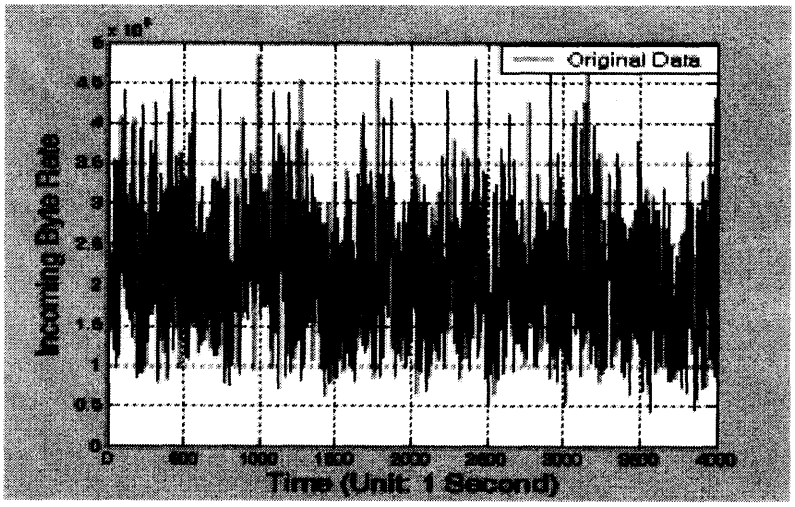

Figure 5.4 A sample of the network traffic measurements collected from the main router of Bergen County Library Network System.

In the remainder of this section, how the statistical properties of network traffic measurements are altered by these low pass filters will be investigated, based on six criteria, i.e., mean, standard deviation, variance, autocorrelation, cumulative density

function (CDF) and Hurst parameter [46]. The data used for this examination is collected from the main router of Bergen County Library Network System, as illustrated in Figure 5.4. The results are present in Figure 5.5, 5.6 and Table 5.1.

According to the results, one can observe that these low pass filters alternate the statistical properties of the network traffic measurements in different degrees. The MWA and Butterworth (0.1) filters change the statistical properties of the network traffic measurement data significantly. For instance, the variance of the data after processing by MWA is almost 1/4 of that of the original data. The Savitzky-Golay and Butterworth (0.2) filters alter the data moderately, while the Butterworth (0.3) and Butterworth (0.4) filters change data lightly. Obviously, the degree that the low pass filter alters the statistical properties of the network traffic measurements closely relates to its passband. The broader passband the low pass filter has, the more lightly it would alter the statistical properties of the network traffic measurements. As mentioned previously, the low pass filter with narrow passband may change the statistical properties of the network traffic measurements significantly and render the loss of crucial information for statistical based NADS to identify network anomalies. On the other hand, though the low pass filter with broad passband may alter the statistical properties of the network traffic measurements lightly, it may not effectively remove traffic burstiness and further diminish the false alarm rate of statistical based NADS. For this reason, it is deserved to calibrate the performance of the low pass filter through applying it to a statistical based NADS and measuring on what degree it can reduce the false alarm rate of the NADS, i.e., the second criteria mentioned previously. In this dissertation, this task is carried out through applying the low pass filters to MAD system. The details of the experiments are present in the next section.

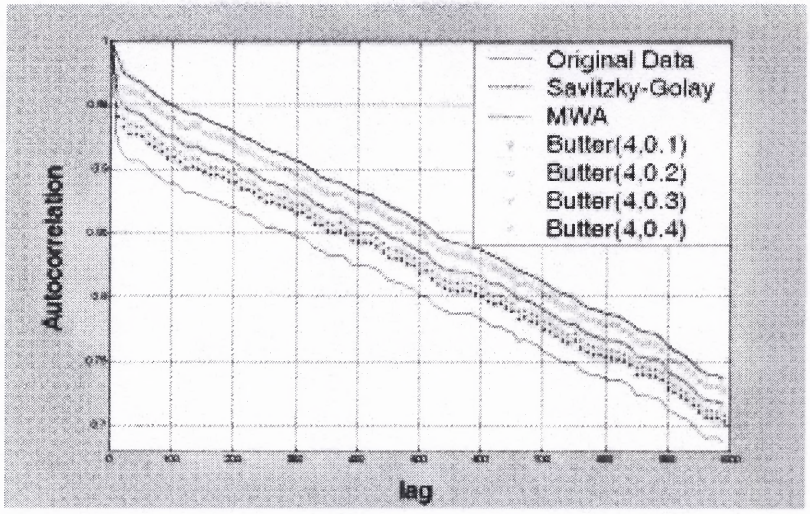

Figure 5.5 Autocorrelation.

Table 5.1 Mean, Variance, Standard Deviation and Hurst Parameter

	Original	MWA	Butterworth (0.1)	Savitzky-Golay	Butterworth (0.2)	Butterworth (0.3)	Butterworth (0.4)
Mean	2.1139e+5	2.1136e+5	2.1095e+5	2.1138e+5	2.1119e+5	2.1128e+5	2.1132e+5
Variance	4.3976e+9	1.2445e+9	1.8253e+9	2.4636e+9	2.4965e+9	2.9926e+9	3.3358e+9
Standard Deviation	6.6315e+4	3.5278e+4	4.2724e+4	4.9635e+4	4.9965e+4	5.4705e+4	5.7757e+4
Hurst Parameter	0.8650	0.9012	0.8742	0.8733	0.8690	0.8669	0.8661

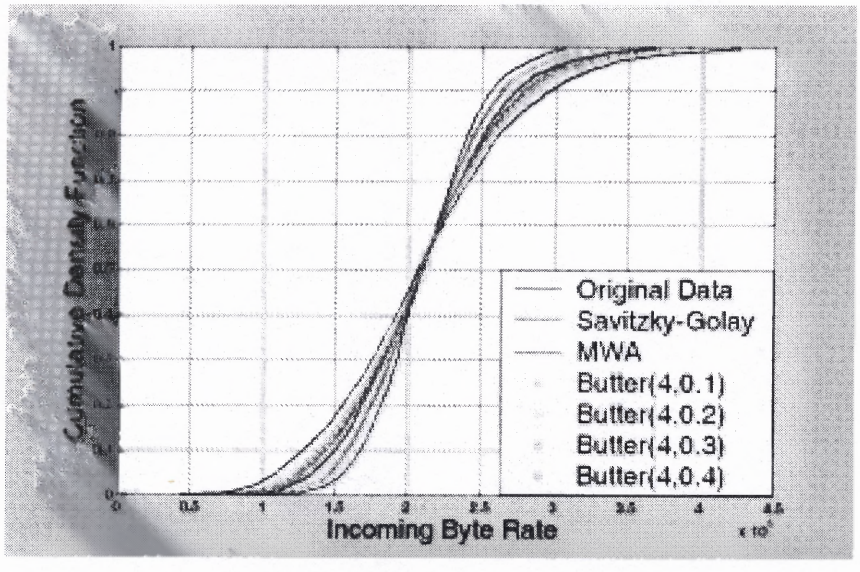

Figure 5.6 Cumulative density function.

5.4. Can the Low Pass Filters Reduce the False Alarm Rate of NADS?

The effectiveness of the low pass filters on reducing the false alarm rate of NADS is investigated through applying them to the MAD system. In Figure 5.7 the refined architecture of an Anomaly Detection Agent (ADA) is presented. Comparing the refined architecture of the ADA shown in this figure with that presented in Figure 2.3, one can clearly observe that a Low Pass Filter module is inserted between the MIB Data Probe module and the Statistical Model module to smooth the collected MIB variable data.

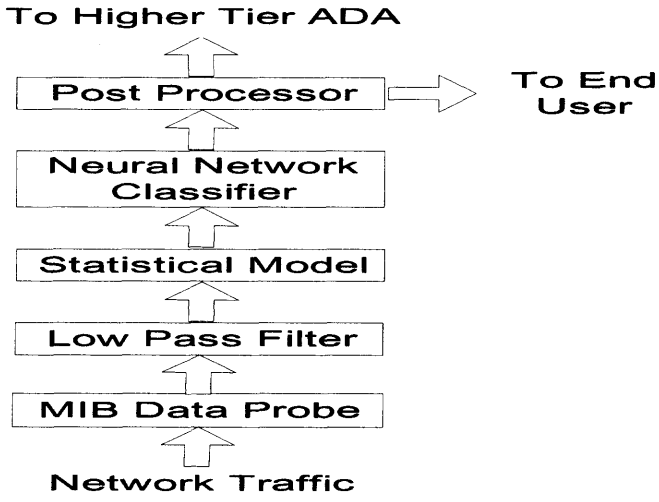

Figure 5.7 The refined architecture of ADA.

The performance of the low pass filters on reducing the false alarm rate of MAD system is evaluated using two criteria, including the *improvement of misclassification rates* and the *improvement of mean squared root errors (MSR),* i.e., the difference of the misclassification rate and MSR before and after employing the low pass filter. The data used in these evaluation experiments are the same as those used to evaluate the performance of MAD in Chapter 3. The results are illustrated in Figure 5.8 and 5.9 and corresponding numerical results list in Table 5.2 and 5.3. In Section 3.2.2, the effectiveness of seventeen alternative similarity measurement metrics has been investigated. The results

presented here reflect the average improvement of these criteria when all the similarity metrics are applied to MAD.

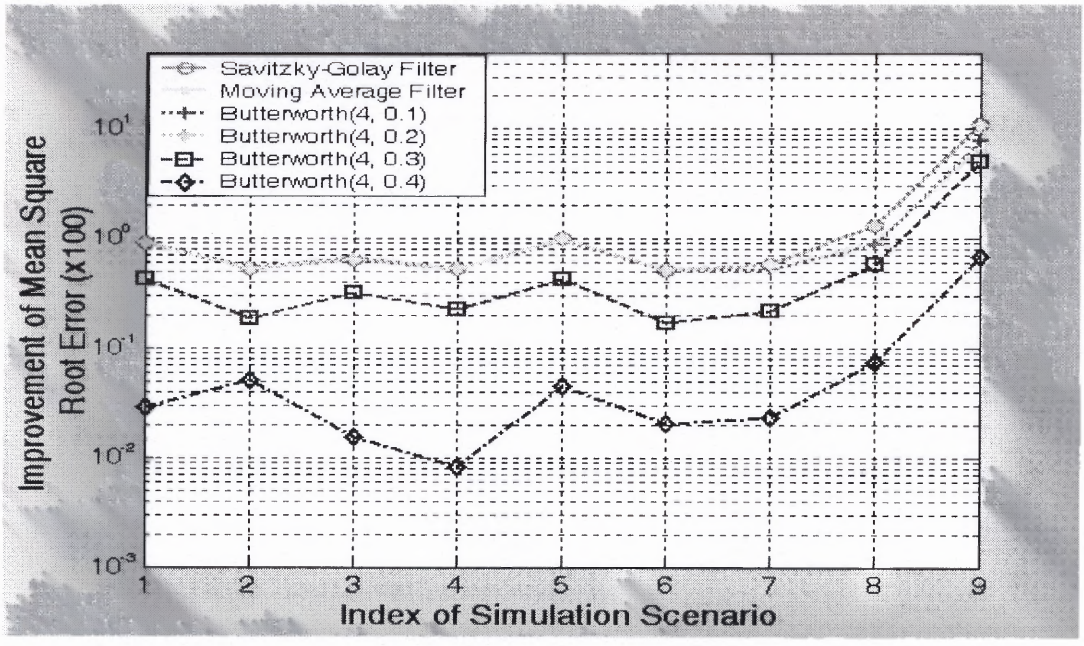

Figure 5.8 The improvement of mean square root error.

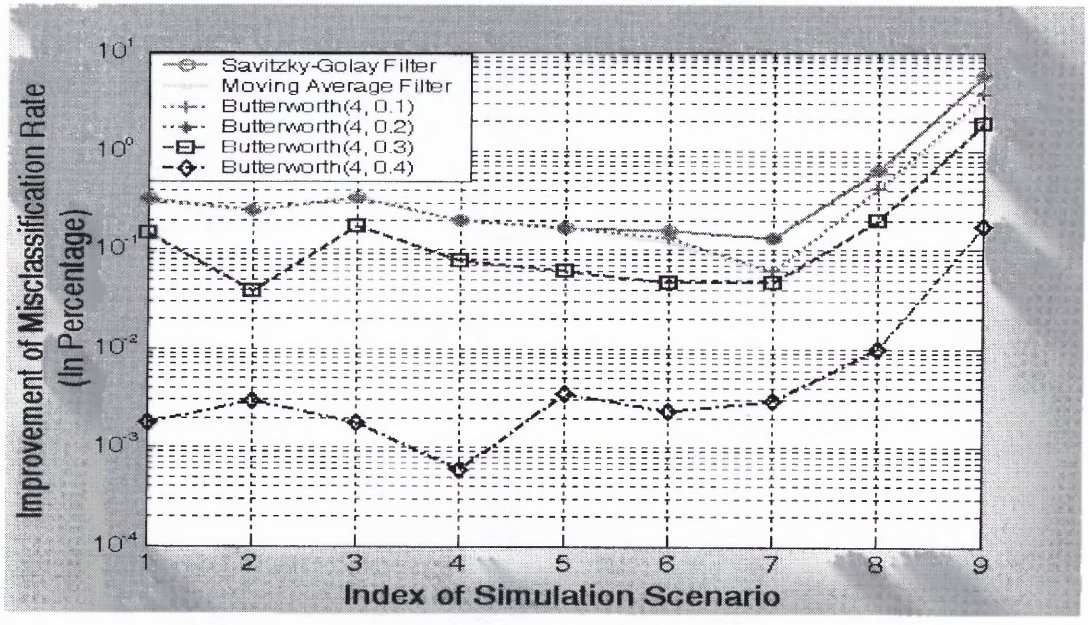

Figure 5.9 The improvement of misclassification rate

Table 5.2 The Improvement of the Mean Square Root Error (x100)

LPF* \ Scenario #	1	2	3	4	5	6	7	8	9
Savitzky-Golay	0.9265	0.5394	0.6365	0.5365	0.9971	0.5241	0.5953	1.3494	11.1282
MWA	0.9271	0.5524	0.6918	0.5376	1.0100	0.5171	0.5400	0.9347	6.8882
Butterworth(4, 0.1)	0.9212	0.5576	0.6624	0.5418	0.9865	0.5253	0.5371	0.8729	7.8788
Butterworth(4, 0.2)	0.9265	0.5412	0.6376	0.5382	0.9912	0.5176	0.5935	1.3371	10.8247
Butterworth(4, 0.3)	0.4288	0.1876	0.3171	0.2253	0.4253	0.1729	0.2194	0.5959	5.0288
Butterworth(4, 0.4)	0.0288	0.0506	0.0153	0.0082	0.0453	0.0206	0.0235	0.0747	0.6835

* LPF stands for low pass filter.

Table 5.3 The Improvement of the Misclassification Rate (in Percentage)

LPF \ Scenario #	1	2	3	4	5	6	7	8	9
Savitzky-Golay	0.3324	0.2512	0.3371	0.1971	0.1647	0.1576	0.1324	0.6512	5.7976
MWA	0.3218	0.2506	0.3471	0.1971	0.1647	0.1194	0.0612	0.4171	3.1876
Butterworth(4, 0.1)	0.3224	0.2524	0.3376	0.1976	0.1647	0.1318	0.0612	0.4088	3.6682
Butterworth(4, 0.2)	0.3329	0.2529	0.3376	0.1971	0.1647	0.1576	0.1324	0.6524	5.6912
Butterworth(4, 0.3)	0.1441	0.0382	0.1729	0.0771	0.0618	0.0465	0.0459	0.1994	1.8871
Butterworth(4, 0.4)	0.0018	0.0029	0.0018	0.0006	0.0035	0.0024	0.0029	0.0100	0.1759

A close look at the results reveals that the performance of Butterworth (4, 0.3) and Butterworth (4, 0.4) filters cannot compare to that of other filters. MWA, Butterworth (4, 0.1), Savitzky-Golay and Butterworth (4, 0.2) filters perform similarly in the simulation scenarios 1 through 6 when the anomaly traffic intensity is high. However, in scenarios 7, 8 and 9 when the anomaly traffic intensity is comparably low, the Savitzky-Golay and Butterworth (0.2) filters perform better than the MWA and Butterworth (4, 0.1) filter. One may find the reason for this in the analysis results presented in Section 5.3. When the MWA and Butterworth (4, 0.1) remove the traffic bursts, it also significantly changes the statistical properties of the network traffic measurements which are crucial for MAD to identify network anomalies. Since the characteristics of network anomalies become more subtle when the anomaly traffic intensity gets lower, the statistical properties of the network traffic measurements become more important for MAD to identify the anomalies. Thus, when these statistical properties are altered, new false alarms may be introduced.

Combining the results presented in this section with the statistical analysis results shown in Section 5.3, the Savitzky-Golay and Butterworth (4, 0.2) filters should be the preferred choices.

CHAPTER 6

A FREQUENCY BASED NETWORK TRAFFIC STATISTICAL MODELING SCHEME

6.1 Motivation

As discussed previously, the operation of the statistical based network anomaly detection system is based on the expectation that the traffic behavior with anomaly will be noticeably different from that without anomaly. Thus, for the purpose of anomaly detection one must characterize normal traffic behavior. Actually, the more accurately the normal traffic behavior can be modeled the better the anomaly detection scheme will perform. These models may be describing cumulative density functions (CDFs) or probability density functions (PDFs) of the monitored traffic variables; examples, of such variables include, incoming packet rate, outgoing byte rate, etc., during an observation time window. Although the size of the observation window varies depending on implementation, in practice, it ranges from a few seconds to many minutes. Thus, for typical actual traffic volumes, the generated CDF/PDF may consist of large numbers of samples for each traffic variable and for each of the observation windows; such representations are expensive in terms of memory storage and computational expense. It is greatly desirable to generate analytical models that are accurate and efficient in terms of representing such data; this directly reduces the storage requirements of the raw data that are collected; such models also allow methods that carry out mathematical computations on the CDFs/PDFs, in terms of their parametric analytic model representations, rather than the raw data that constitute the CDFs/PDFs; this reduces the associated processing cost. Thus, efficient statistical network traffic activity models are desirable.

Accurately and efficiently modeling the traffic variables is a hard task. The biggest challenge in this case is the well-known fact that these variables of networks undergo cyclic evolution and temporal fluctuation [18-19, 59-61]. Figure 6.1 and 6.2 illustrated this phenomenon more clearly. Figure 6.1 plots three weeks of network traffic observations collected from the Main Router of BCCLNS network, while Figure 6.2 depicts an example of a weekday's traffic observations collected from the same router (the details of the traffic variable data collection are present in section 6.3.1). From Figure 6.1, one may observe that there is an underlying trend for every week. For each weekday morning, the network traffic volume increases as people arrive at work. Soon afterwards, the network traffic volume peaks and remains there for most of the day. In the late afternoon, values return to a lower level. Such high variable traffic patterns are illustrated more clearly in Figure 6.2. On weekends, the usage is comparably low throughout the day.

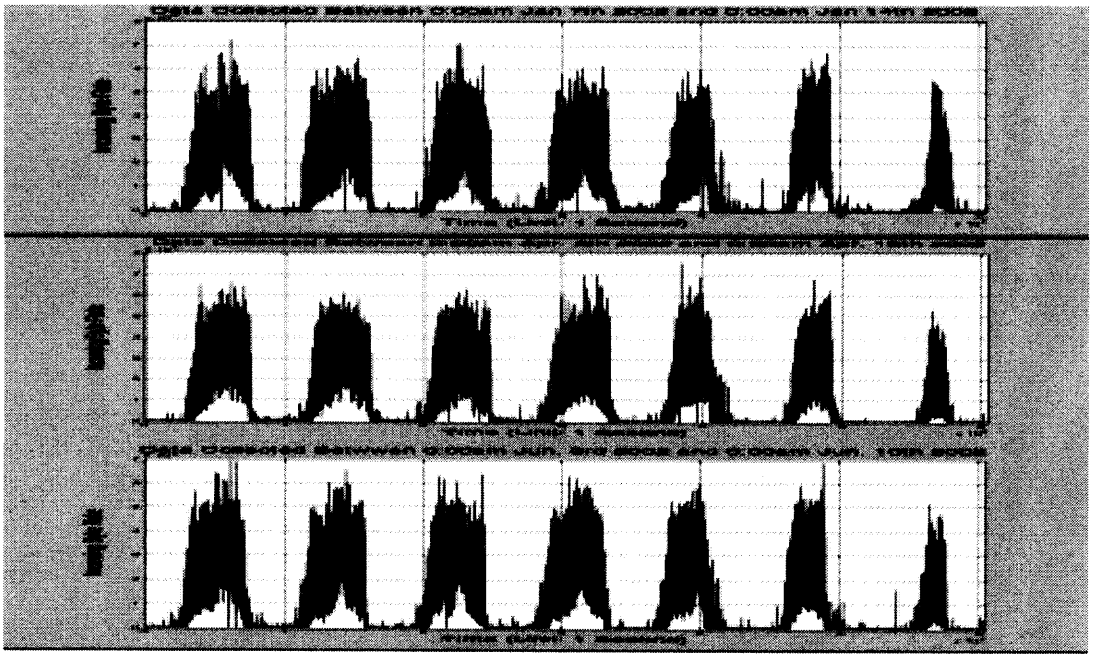

Figure 6.1 Plot of three weeks of traffic variable observations.

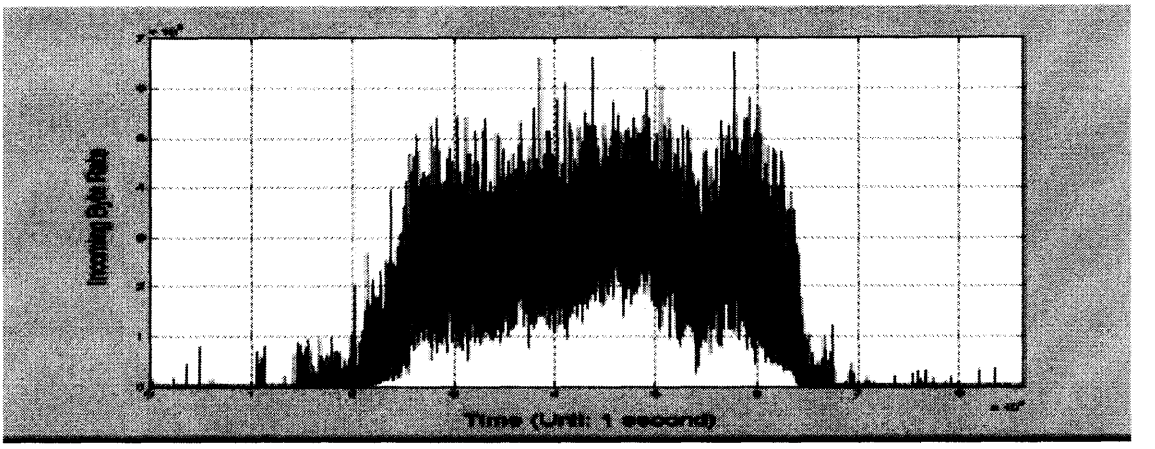

Figure 6.2 Plot of one day of traffic variable observations.

Obviously, the traffic variable observations are non-stationary in that the mean changes with time-of-day and day-of-week. However, the previous figures suggest that the same non-stationary pattern is present from week to week (or that it changes slowly). Thus, one may view the full process of the traffic variable as having a stable mean for each time of the day along with fluctuations that are modeled by a random variable with a zero mean.

6.2 A Frequency Based Network Traffic Statistical Modeling Scheme

In this work, a frequency based network traffic statistical modeling scheme is introduced, that is, dividing the full process of the traffic variable observations into three parts according to their frequency spectrum, namely, low frequency part signal, middle frequency part signal and high frequency part signal by the 4^{th} order Butterworth low pass filter, and then modeling each part of signal separately.

- Low Frequency Part Signal, obtained by filtering the original traffic variable data process by the 4^{th} order Butterworth low pass filter with cutoff frequency 0.05 Hz. Thus the filtered process represents the mean process (or traffic trend pattern) of the original data process.

- Middle Frequency Part Signal, obtained by filtering the residue process after subtracting the low frequency part signal from the original data, by the 4th Butterworth low pass filter with cutoff frequency 0.2 Hz.
- High Frequency Part Signal, obtained by subtracting the low and middle frequency part signals from the original data process.

In the following sections, the traffic variable data, collected from the main router of BCCLNS beginning at 0:00:00am Thursday, April 8th, 2002 and ending at 00:00:00am Friday, April 9th 2002 (shown in Figure 6.3), will be used to illustrate how the proposed frequency based network traffic statistical modeling scheme works.

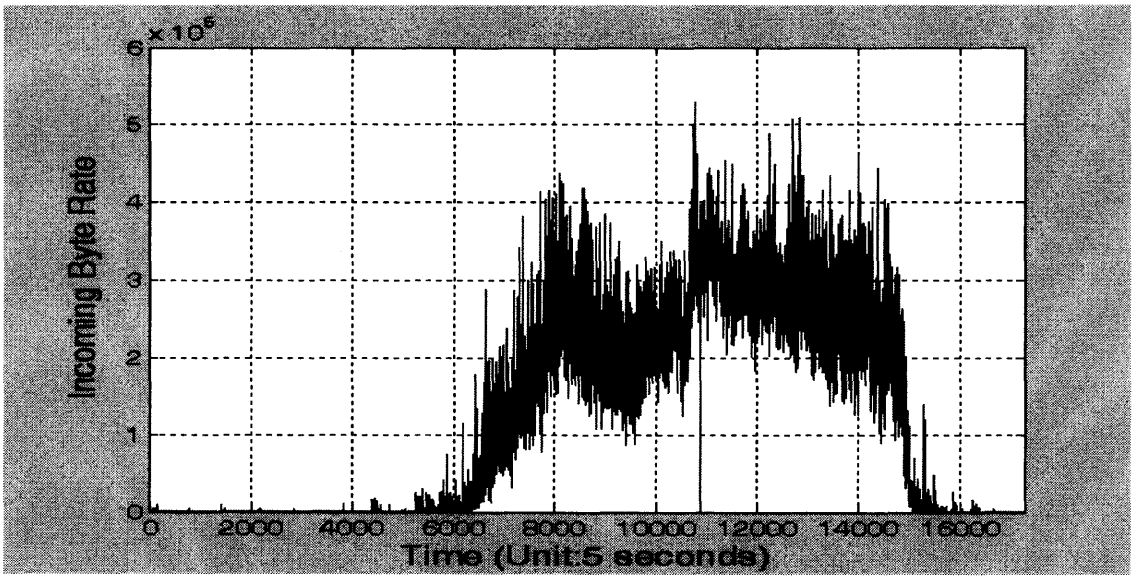

Figure 6.3 A sample of one day's traffic variable observations collected from the main router of BCCLNS.

6.2.1 Modeling Low Frequency Part Signal

Figure 6.4 shows the low frequency part signal of the traffic variable data plotted in Figure 6.3. In this figure, the sampled data processes, obtained by taking one sample in every 100, 150 and 200 successive data records of the low frequency part signal data process, are also presented. A close look at the figure reveals that the low frequency part signal is quite

smoothed in comparison with the original data presented in Figure 6.3. Obviously, it can be found that the low frequency part signal data process is non-stationary since the mean changes significantly during most of the day; thus it is infeasible to use a single random distribution to accurately present these data. From the figure, it is found that the curves of the sampled data follow that of the original low frequency part signal data very well even though the sampling interval is quite coarse. This observation is confirmed in Figure 6.5, where the cumulative density functions (CDFs) of the original low frequency part signal data and the sampled data are plotted. From this figure, one can observe that the curves of the CDFs of the sampled low frequency signal data are quite close to that of the original data in shape. To further measure the deviations between the CDF of the original low frequency part signal and those of the sampled data, the Kolmogorov-Smirnov (KS) type similarity measurement metric was employed, as given in Equation 2.25. The results are listed in Table 6.1. From Table 6.1, one may notice that all these K-S distances are significantly small, which indicates that the CDFs of the original low frequency part signal data and the sampled data are quite close to each other. Thus, one may use the sampled low frequency part signal data, rather than the original data, as a data model, to present the low frequency components of the traffic variable observations. Thus, the size of memory required for storing the data model for the low frequency part signal is sharply reduced.

Table 6.1 The K-S Distance Results for Modeling the Low Frequency Part Signal

Sampled data	K-S distance
Data sampled every 100 records	0.008
Data sampled every 150 records	0.011
Data sampled every 200 records	0.013

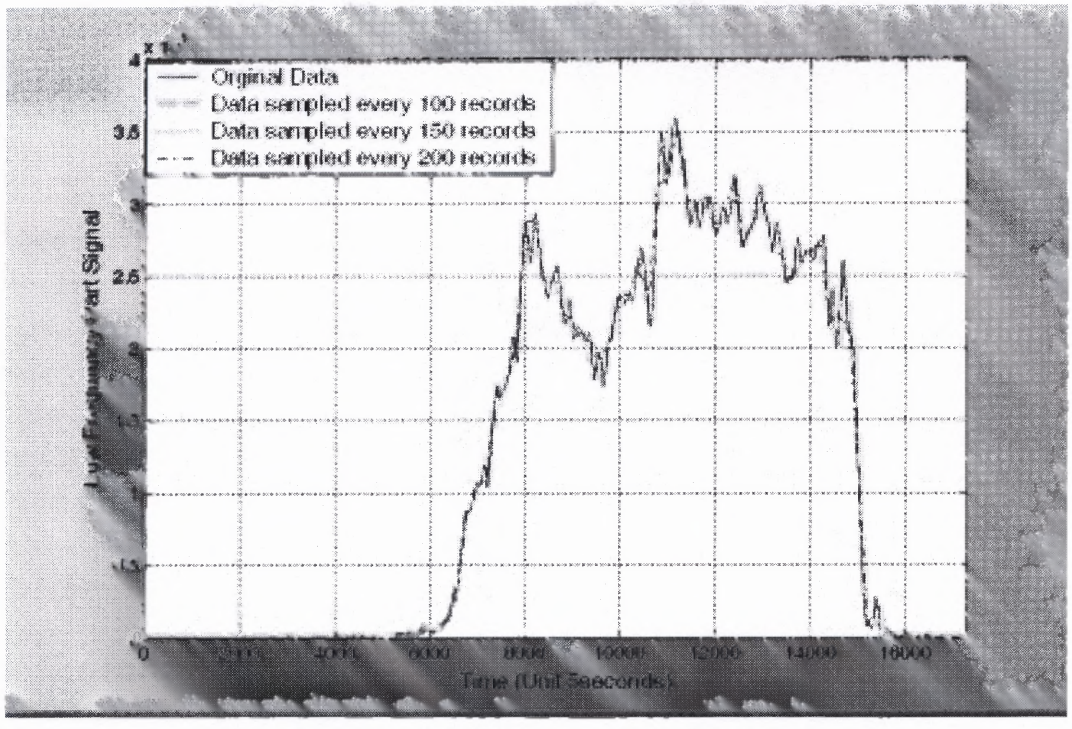

Figure 6.4 The original and sampled low frequency part signal data.

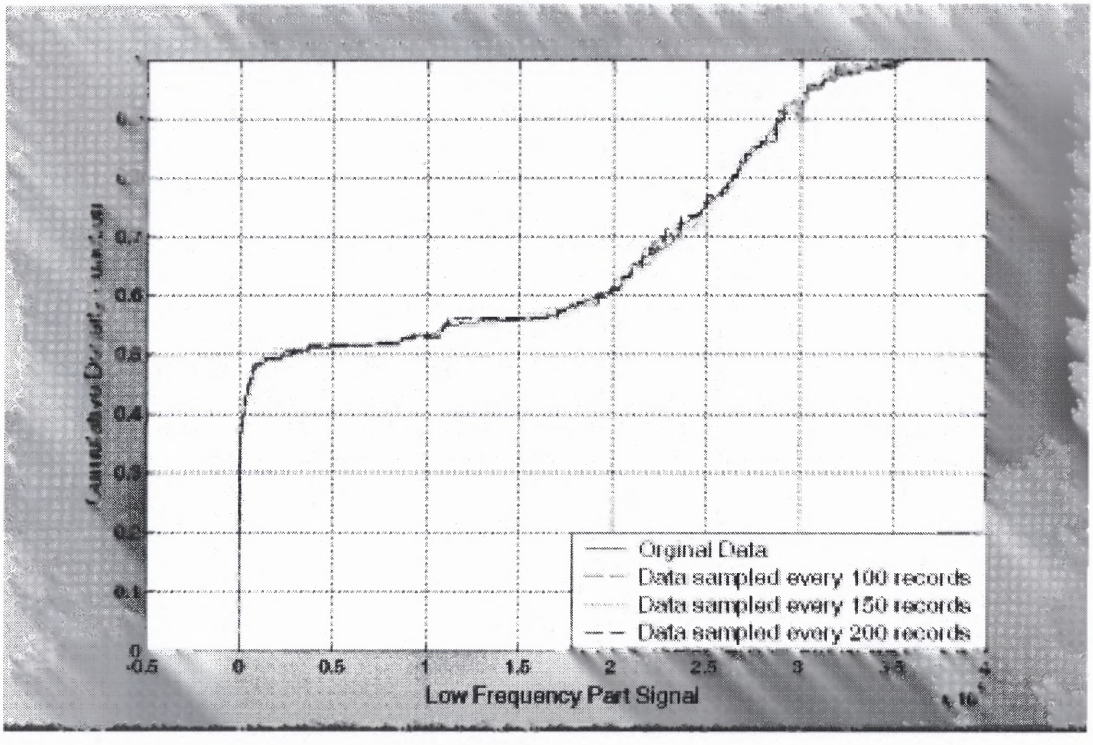

Figure 6.5 The CDF plots of the original and sampled low frequency part signal data.

6.2.2 Modeling Middle Frequency Part Signal

The middle frequency part signal of the traffic variable data plotted in Figure 6.3 is shown in Figure 6.6. From this figure, one can observe that there still exists high variability in the middle frequency components of the traffic variable data. As shown in Figure 6.7, where the whole data process is cut into 10 segments with equal size and then the CDF of the data in each segment is plotted, the CDFs from the different data segments show significant deviations, which testifies that the middle frequency part signal data process is also non-stationary. Thus, like the low frequency part signal, the middle frequency part signal also cannot be modeled accurately by a single random distribution.

As with the low frequency part signal, the middle frequency part signal can also be well modeled by its sampled data. As shown in Figure 6.6 and 6.8, the sampled data look very similar to the original data of the middle frequency part signal. This observation is confirmed numerically by the K-S distances between the CDFs of the original middle frequency part signal data and the sampled data listed in Table 6.2. From the results shown in this table, it is seen that the values of the KS distances are quite small, which indicates that the CDFs of the original and sampled middle frequency part signals are quite close to each other. Thus, it is desirable to use the sampled data as a data model to present the middle frequency components of the traffic variable observations, so as to save the memory space for containing the data model.

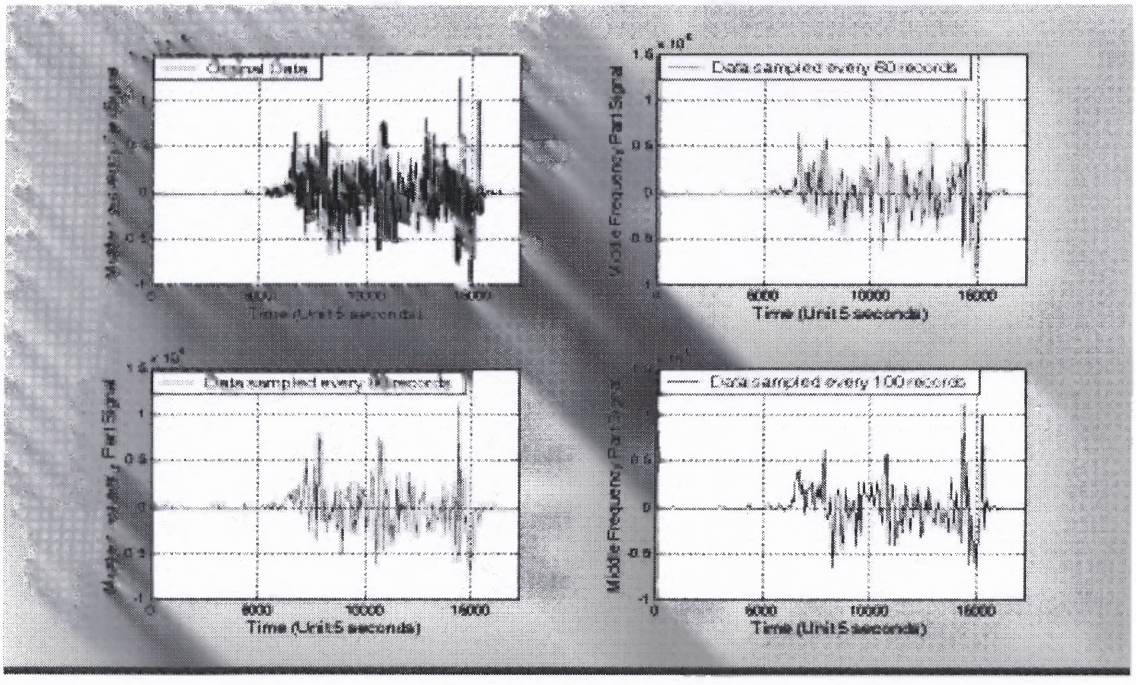

Figure 6.6 The original and sampled middle frequency part signal data.

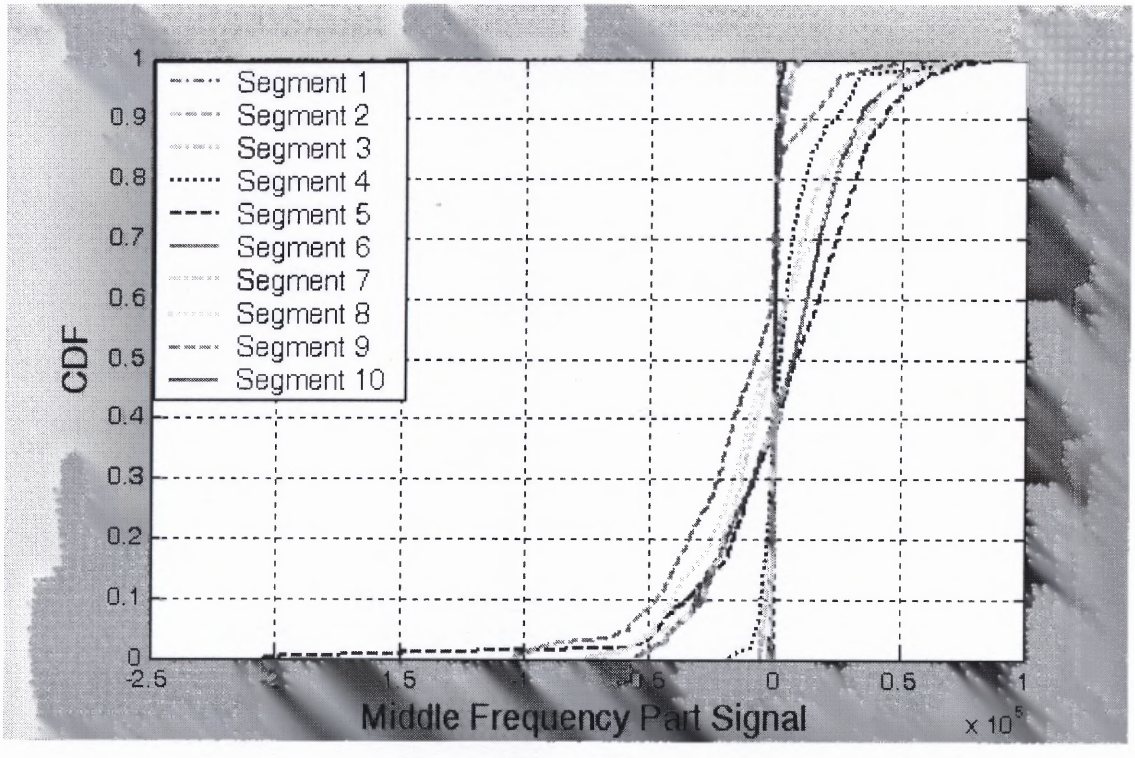

Figure 6.7. The CDF plots of the segmented middle frequency part signal data.

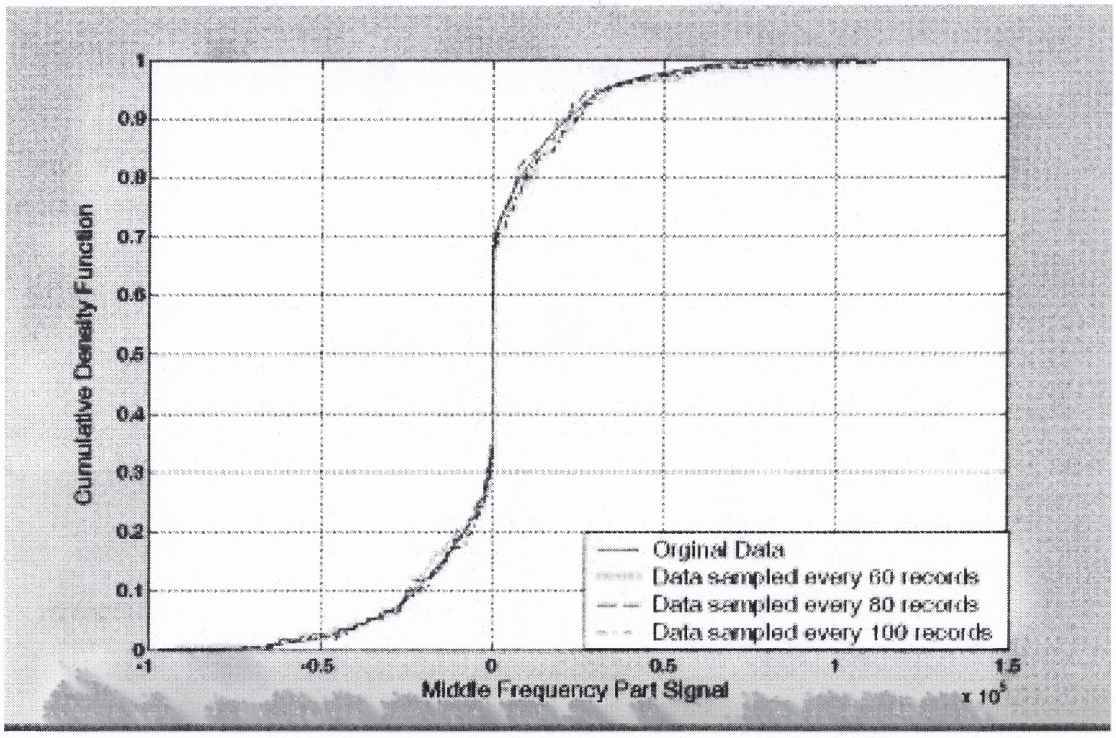

Figure 6.8 The CDF plots of the original and sampled middle frequency part signal data.

Table 6.2 The K-S Distance Results for Modeling the Middle Frequency Part Signal

Sampled data	K-S distance
Data sampled every 60 records	0.009
Data sampled every 80 records	0.013
Data sampled every 100 records	0.016

6.2.3 Modeling High Frequency Part Signal

Figure 6.9 depicts the high frequency part signal of the traffic variable data shown in Figure 6.3. Clearly, this plot is more consistent with a stationary process than when the low and middle frequency part signals are present.

As with the analysis for the middle frequency part signal, the whole data set of the high frequency part signal is cut into ten segments. The CDFs of the data in the different segments are compared in Figure 6.10. Also, in this figure the CDF for the whole data

process of the high frequency part signal is plotted in order to compare with the CDFs of the segmented data. From this figure, one can clearly observe that the CDFs from the segmented data are quite close to one another in shape. This observation is confirmed by the numerical results presented in Table 6.3, where the averaged K-S distances between the CDFs of the segmented data are listed. From the results shown in this table, one can clearly see that the values of the averaged K-S distances are quite small. A close look at Figure 6.10 also reveals that the curve of the CDF of the whole data set of the high frequency part signal follows those of the segmented data quite well, which indicates that a single random distribution is enough to accurately present the data in the different time period of the high frequency part signal.

Similar results are obtained when the high frequency part signal data process is cut into 20, 40 and 60 and 80 segments. The investigation results are presented in Table 6.4. The first column of this table gives the number of segments that the whole data set of the high frequency part signal is cut into. The second and third columns list the average and the standard deviation of the KS distances between the CDFs of the segmented data, while the last column shows the ratio between the statistics listed in the second and third columns, which can be used to measure how the individual KS distance differs from its corresponding average value. From the results shown in this table, one can clearly observe that the values of the averages and standard deviations of the KS distances are quite small, while the values of the ratio between these two statistics as listed in the fourth column are significantly large. These results demonstrate that the KS distances between the CDFs of the segmented data converge to a significantly small range with the central point at the averaged KS distance which is a quite small value. The results indicate that the data of the

high frequency part signal process in the different time period has the same (or similar) CDF. In other words, the CDF of the high frequency part signal is independent of time, which indicates that the high frequency part signal is approximately consistent with a first-order stationary process.

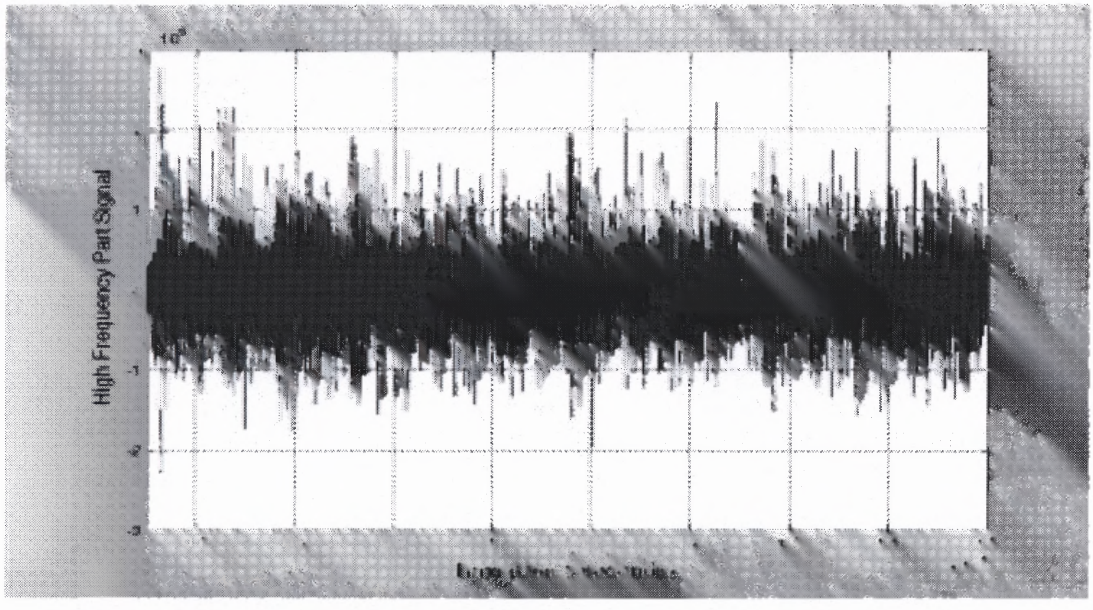

Figure 6.9 The high frequency part signal data.

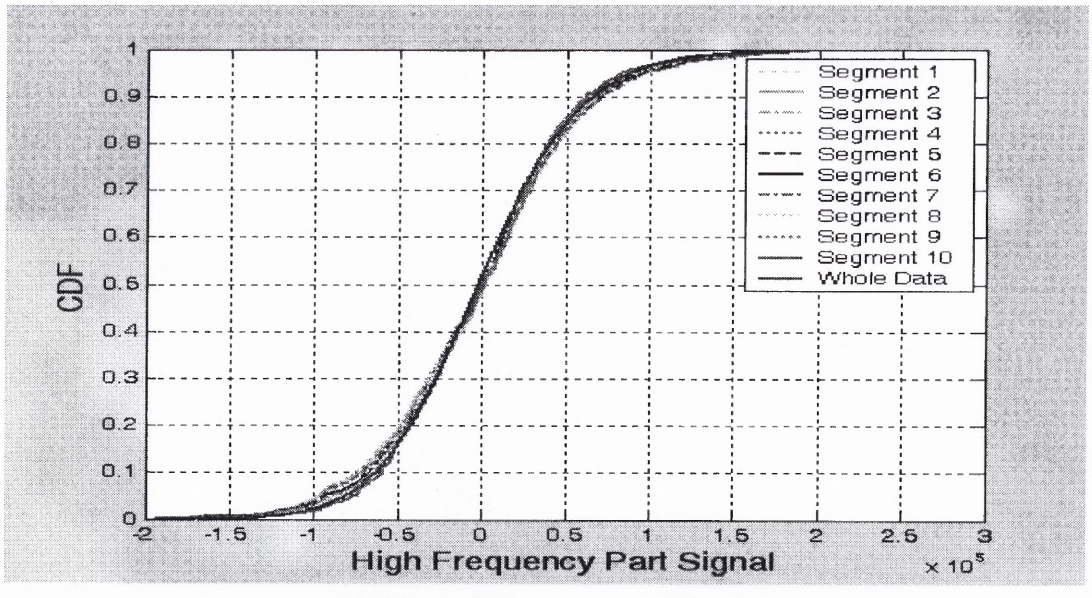

Figure 6.10 The CDF plots of the segmented high frequency part signal data.

Table 6.3. The Averaged K-S Distance Between the CDFs of the Segmented High frequency Part Signal Data

The Index of the data segment	K-S distance
1	0.0030
2	0.0031
3	0.0031
4	0.0031
5	0.0031
6	0.0032
7	0.0030
8	0.0030
9	0.0029
10	0.0030

Table 6.4. Investigating the deviation Between the CDFs of the Segmented High frequency Part Signal Data

The Number of the data segment	Average KS Distance (D)	Standard Deviation of Average KS Distance (ΔD)	D/ΔD
10	0.0031	8.4984e-5	36.4776
20	0.0037	1.8723e-4	19.7618
40	0.0068	9.5923e-4	7.0890
60	0.0101	1.2933e-3	7.7322
80	0.0171	2.7542e-3	6.2087

In this work, normal distribution is employed to model the high frequency part signal data. Besides that, a new statistical model to the study of network statistical modeling is introduced, namely the one-dimensional hyperbolic distribution. In the recent past, the hyperbolic distribution has served as a useful alternative to the normal distribution in the analysis of data in a number of scientific areas [62-64]. The hyperbolic distribution that has been used as the model in fitting the high frequency part signal data is presented below:

$$\frac{\kappa/\delta}{2\alpha K_1(\kappa\delta)} e^{-\alpha\{\delta^2+(x-\mu)^2\}^2+\beta(x-\mu)} \tag{6.1}$$

where $\kappa = \sqrt{\alpha^2 - \beta^2}$ while K_1 denotes the modified Bessel function of the third kind and with index 1, as shown in Equation 6.2.

$$K_{\lambda}(x) = \frac{1}{2}\int_{0}^{\infty} y^{\lambda-1} e^{-\frac{x}{2}(y+y^{-1})}\, dy \quad x>0 \tag{6.2}$$

From Equation 6.1, one can see that the hyperbolic distribution depends on four parameters, α, β, δ and μ. This is two more than the normal distributions, but still a very compact representation of large numbers of data points. By constructing "best fits" to the data in this work, it is shown that the hyperbolic distribution outperforms the normal distributions.

The data values for the high frequency part signal were organized into CDF. Next, a best fit was obtained to the candidate representational distribution model, followed by the calculation of the distance of the set of measured points to the model curve using the Kolmogorov-Smirnov (K-S) distance metric, as given in eq. 2.25, and the following well-known associated probability expression, shown in eq. 2.26.

The fitting results for the high frequency part signal data depicted in Figure 6.9 are shown in Table 6.5. The first column gives the distributions used to fit to the data set. The second and third columns give the fitting results, i.e., the calculated K-S distance, as well as the associated probability that the data derives from the fitted model distribution.

From the results, one can observe that for normal distribution, though the K-S distance is small, they are not small enough, so that the K-S probabilities are negligible. With respect to the hyperbolic distribution, it is found that the K-S distance is significantly smaller in comparison to that of the normal distribution, resulting in very large K-S probabilities. This indicates that one should be inclined to accept the hypothesis that the data in question derive from a population whose true density is hyperbolic.

Visually as well, in Figure 6.11, it is seen that though the normal distributions fit the main body of the data reasonably well, there are deviations in the lower and upper tails of the data. On the contrary, the curve of the hyperbolic distribution not only matches the main body of the curve of the data quite well, but does so for the lower and upper tails as well.

Table 6.5 The Fitting Results for the High Frequency Part Signal

	K-S Distance	K-S Probability
Normal Distribution	0.032	0.000
Hyperbolic Distribution	0.009	0.495

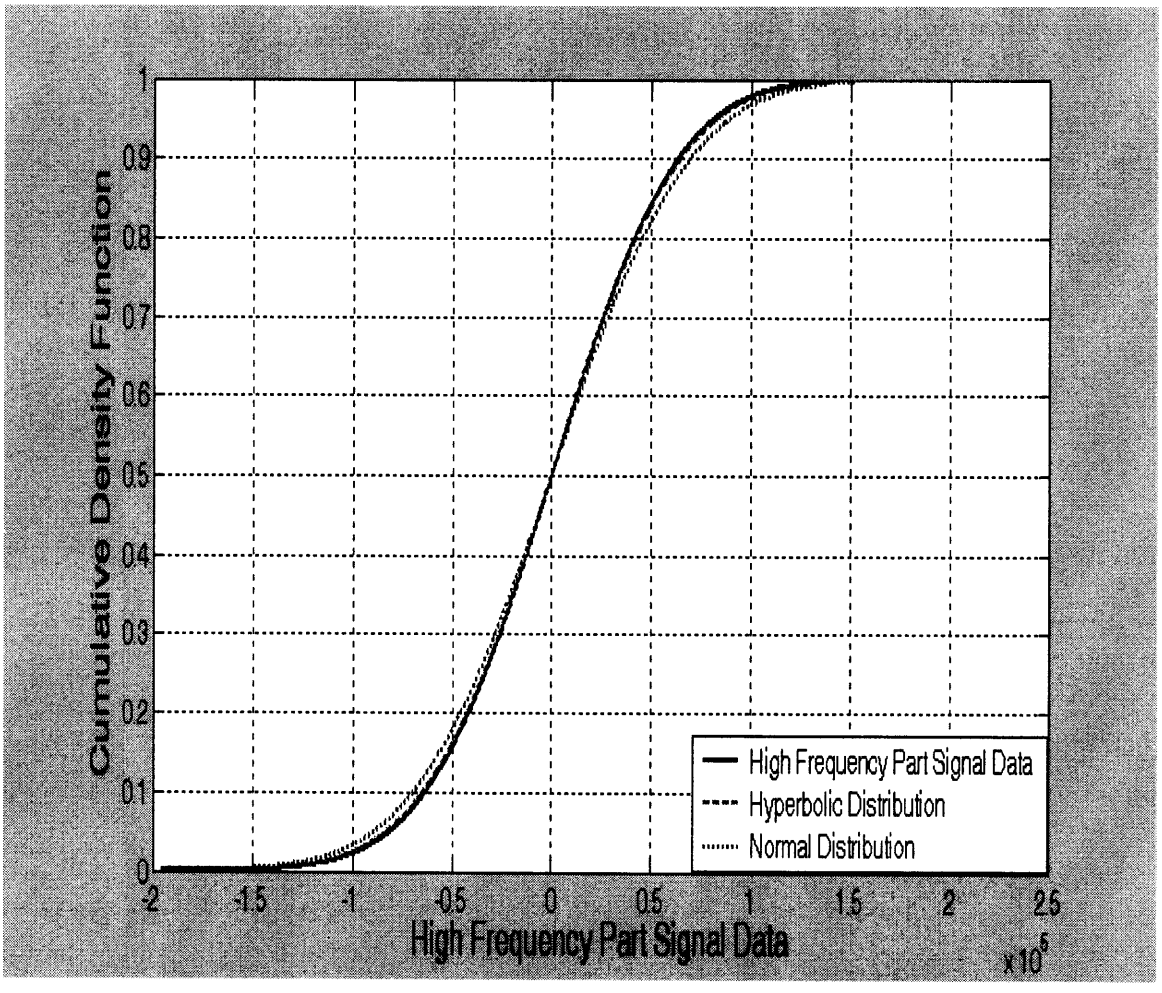

Figure 6.11 The fitting results for the high frequency part signal.

6.3 Model Evaluation

6.3.1 Data Collection

The data traces used in this work were collected from an actual Wide Area Network (WAN) [65], namely, the Bergen County Cooperative Library Network System (BCCLNS). In [65] by Y. Y. Lee, the author analyzed the data collected and tried to use the normal, Pareto, Weibull, and hyperbolic distribution to model the data at hand. In this dissertation, these data traces will be used to evaluate the performance of the proposed frequency based traffic variable statistical modeling scheme described in the previous section.

BCCLNS is a consortium of public libraries that delivers quality library service to the general public through sharing a computer system and providing common access to electronic resources. The 74 members include all 62 of the County's public libraries and 12 libraries from neighboring counties. BCCLNS supports resource sharing with an automated library circulation, catalog, and wide area network. High-speed connections (2 T3 lines) are connected to local libraries (65 T1 and 13 56k lines from local libraries via frame relay) and a T1 connection to the Internet. Local libraries are connected to BCCLNS in a TCP/IP environment with PCs via routers. As of November 2002, 1158 PC devices are connected to the WAN.

A simplified abbreviated network topology for BCCLNS is presented in Figure 6.12 that illustrates the network connectivity of this WAN. Each subnet represents a local area network that serves a library. A central router (labeled BCCLNS Main Router) is utilized to connect all libraries, as well as provide connectivity to the Internet. An element

labeled Main Frame, located in the subnet Main Office, maintains the databases for all of book catalogs that can be queried by the readers and librarians.

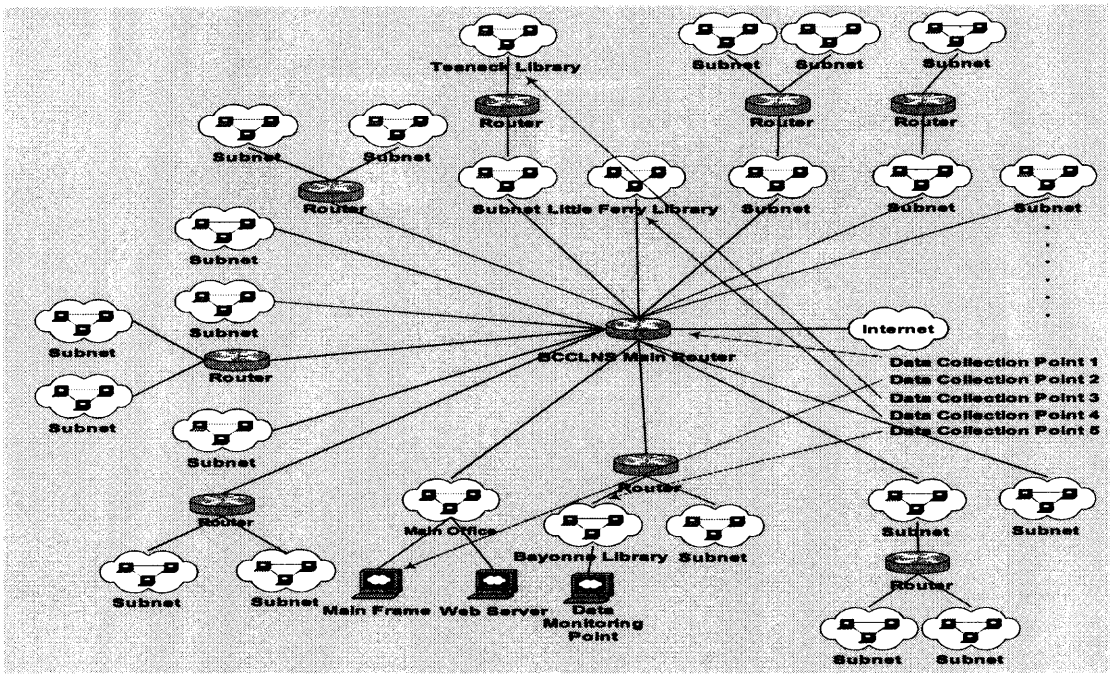

Figure 6.12 The simplified abbreviated network topology of BCCLNS

The network traffic was gathered using a collection time window of size 5 seconds, from five different points, i.e., the BCCLNS Main Router, the BCCLNS Main Frame, the Access Router of Teaneck Library, the Access Router of Little Ferry Library and the Access Router of Bayonne Library. Such five different collection points may be reasonably expected to provide a variety of network traffic traces. The interface of the Main Router that is monitored is the only access point to the internet for the whole WAN, thus, the traffic collected at this point presents the data transfer requests from the whole library system community for internet services. The traffic observations obtained from the Main Frame presents the traffic pattern flowing through a large server. At the data collection points 3, 4, and 5, the traffic data were collected from the access routers which served a

local area network of similar scale. Thus, such traffic measurements can present the network pattern for a local area network. A detailed description of the traffic data traces that were gathered from these five collection points is given in Table 6.6.

Table 6.6 The Specification of the Data Traces for the Model Evaluation

Data Traces Collection Point	Data Trace Index	Collection Time
BCCLNS Main Router	MR-1	00:00:00, Mar. 2, 2002 ~ 00:00:00, Mar. 3, 2002
	MR-2	00:00:00, May 26, 2002 ~ 00:00:00, May 27, 2002
BCCLNS Main Frame	MF-1	00:00:00, Sep. 17, 2002 ~ 00:00:00, Sep. 18, 2002
	MF-2	00:00:00, Oct. 9, 2002 ~ 00:00:00, Oct. 10, 2002
Teaneak Library Access Router	TLAR-1	00:00:00, May 23, 2002 ~ 00:00:00, May 24, 2002
	TLAR-2	00:00:00, Jun 10, 2002 ~ 00:00:00, Jun 11, 2002
Little Ferry Library Access Router	LFLAR-1	00:00:00, Jun. 13, 2002 ~ 00:00:00, Jun. 14, 2002
	LFLAR-2	00:00:00, July 8, 2002 ~ 00:00:00, July, 9, 2002
Bayonne Library Access Router	BLAR-1	00:00:00, Jan. 7, 2002 ~ 00:00:00, Jan. 8, 2002
	BLAR-2	00:00:00, Feb, 20, 2002~ 00:00:00, Feb. 21, 2002

6.3.2 Model Evaluation Results and Discussion

The numerical results for modeling the low and middle frequency part signal data are present in Table 6.7 and 6.8. The leading columns of these tables give the indices of the data traces collected from BCCLNS. The following three columns give the K-S distances between the CDFs of the original low or middle frequency part signal data and its sampled data obtained by taking samples from the original data using different sampling frequencies. From the results shown in Table 6.7 and 6.8, it can be seen that all the K-S distances are quite small, which confirms the results received in Section 6.2 that the low and middle frequency part signals can be well presented by their sampled data.

Virtually as well, from Figure 6.13 and 6.14, where the CDFs of the original low or middle frequency part signal data and its sampled data for the data trace MF-1 are plotted, one can see that the original and sampled data curves are visually indistinguishable. This visual estimate is, in fact, well supported by the numerical results that the sampled low and

middle frequency part signal data can present their original data very well, for all the data traces at hand.

Table 6.7 The Evaluation Results for Modeling the Low Frequency Part Signal

Data Trace Index	K-S Distance		
	Data sampled every 100 records	Data sampled every 150 records	Data sampled every 200 records
MR-1	0.009	0.011	0.014
MR-2	0.007	0.010	0.012
MF-1	0.008	0.012	0.014
MF-2	0.009	0.010	0.012
TLAR-1	0.010	0.011	0.015
TLAR-2	0.009	0.012	0.014
LFLAR-1	0.006	0.009	0.012
LFLAR-2	0.007	0.010	0.013
BLAR-1	0.008	0.011	0.014
BLAR-2	0.009	0.012	0.016

Table 6.8 The Evaluation Results for Modeling the Middle Frequency Part Signal

Data Trace Index	K-S Distance		
	Data sampled every 60 records	Data sampled every 80 records	Data sampled every 100 records
MR-1	0.010	0.013	0.017
MR-2	0.009	0.012	0.015
MF-1	0.009	0.012	0.015
MF-2	0.009	0.013	0.017
TLAR-1	0.011	0.013	0.016
TLAR-2	0.010	0.013	0.015
LFLAR-1	0.009	0.011	0.015
LFLAR-2	0.010	0.014	0.017
BLAR-1	0.010	0.013	0.016
BLAR-2	0.011	0.015	0.018

To investigate whether the high frequency part signals of the data traces listed in Table 6.6 are consistent with a first order stationary process, similar method is employed as described in Subsection 6.2.3. The investigation results are listed in Table 6.9. The leading column gives the data trace indices that are designated in Table 6.6. Similarly to Table 6.4, the second column shows the number of segments that the whole data trace of the high frequency part signal is cut into, while the last three columns give the statistics used to measure the deviation between the CDFs of the segmented data. From this table,

one can clearly find out that the investigation results got from the high frequency part signals of these data traces are very similar to those listed in Table 6.4, which indicates that the high frequency part signals of these data traces are also approximately consistent with a first-order stationary process.

Table 6.9 Investigating the Deviation Between the CDFs of the Segmented High frequency Part Signal Data

Data Trace Index	The Number of the data segment	Average KS Distance (D)	Standard Deviation of Average KS Distance (ΔD)	D/ ΔD
MR-1	20	0.0034	1.7834e-4	19.0647
	40	0.0059	7.3452e-4	8.0325
	80	0.0151	1.9283e-3	7.8307
MR-2	20	0.0037	1.9231e-4	19.2398
	40	0.0071	5.6439e-4	12.5800
	80	0.0184	2.3495e-3	7.8315
MF-1	20	0.0039	2.1923e-4	17.7895
	40	0.0069	1.2903e-3	5.3480
	80	0.0189	4.2934e-3	4.4021
MF-2	20	0.0035	2.1029e-4	16.6437
	40	0.0064	9.2931e-4	6.8868
	80	0.0148	2.3342e-3	6.3405
TLAR-1	20	0.0039	2.4023e-4	16.2344
	40	0.0072	1.1232e-3	6.4103
	80	0.0178	3.2944e-3	5.4031
TLAR-2	20	0.0036	2.4535e-4	14.6729
	40	0.0063	9.3495e-4	6.7383
	80	0.0201	4.2934e-3	4.6816
LFLAR-1	20	0.0041	1.9182e-4	21.3742
	40	0.0087	9.9832e-4	8.7146
	80	0.0211	4.1923e-3	5.0330
LFLAR-2	20	0.0044	2.1293e-4	20.6641
	40	0.0069	1.0293e-3	6.7036
	80	0.0174	2.8759e-3	6.0503
BLAR-1	20	0.0035	2.0239e-4	17.2933
	40	0.0065	9.7823e-4	6.6447
	80	0.0173	2.8456e-3	6.0796
BLAR-2	20	0.0038	2.4532e-4	15.4900
	40	0.0074	1.4324e-4	5.1662
	80	0.0163	3.3534e-3	4.8607

Table 6.10 The Evaluation Results for Modeling the High Frequency Part Signal

Data Trace Index	Normal Distribution		Hyperbolic Distribution	
	K-S Distance	K-S Probability	K-S Distance	K-S Probability
MR-1	0.041	0.000	0.010	0.362
MR-2	0.052	0.000	0.009	0.495
MF-1	0.051	0.000	0.008	0.647
MF-2	0.039	0.000	0.011	0.254
TLAR-1	0.043	0.000	0.009	0.447
TLAR-2	0.062	0.000	0.010	0.348
LFLAR-1	0.021	0.001	0.007	0.798
LFLAR-2	0.033	0.000	0.009	0.459
BLAR-1	0.014	0.071	0.006	0.899
BLAR-2	0.046	0.000	0.010	0.350

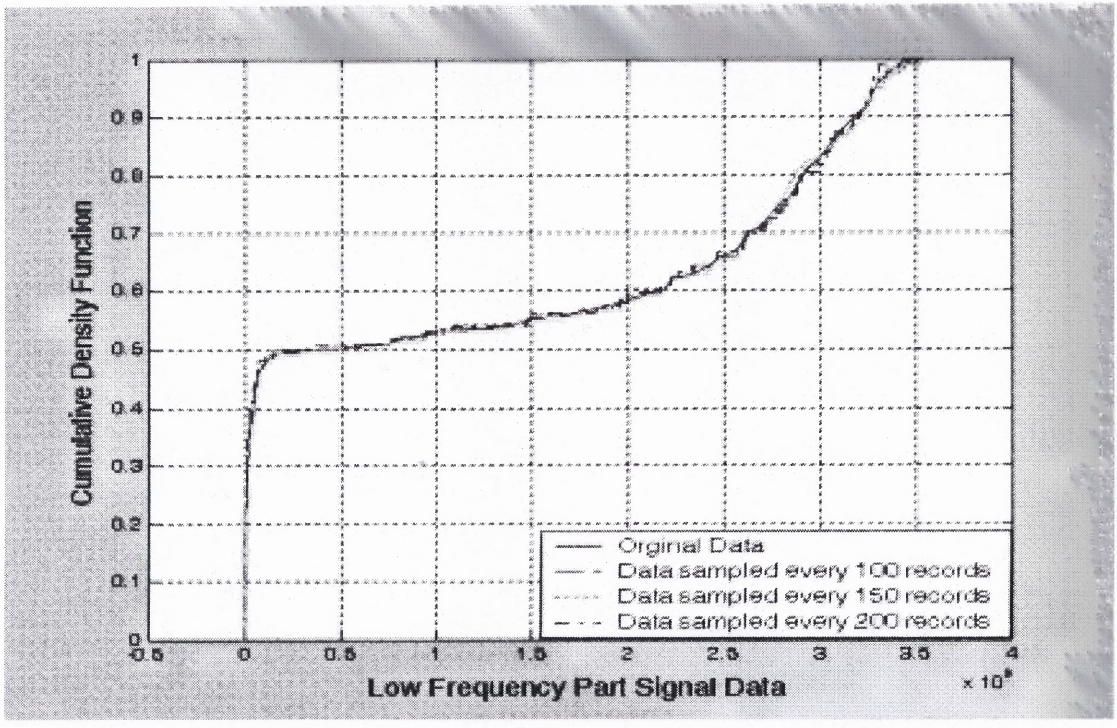

Figure 6.13 The CDF plots of the original and sampled low frequency part signal data for the data trace MF-1.

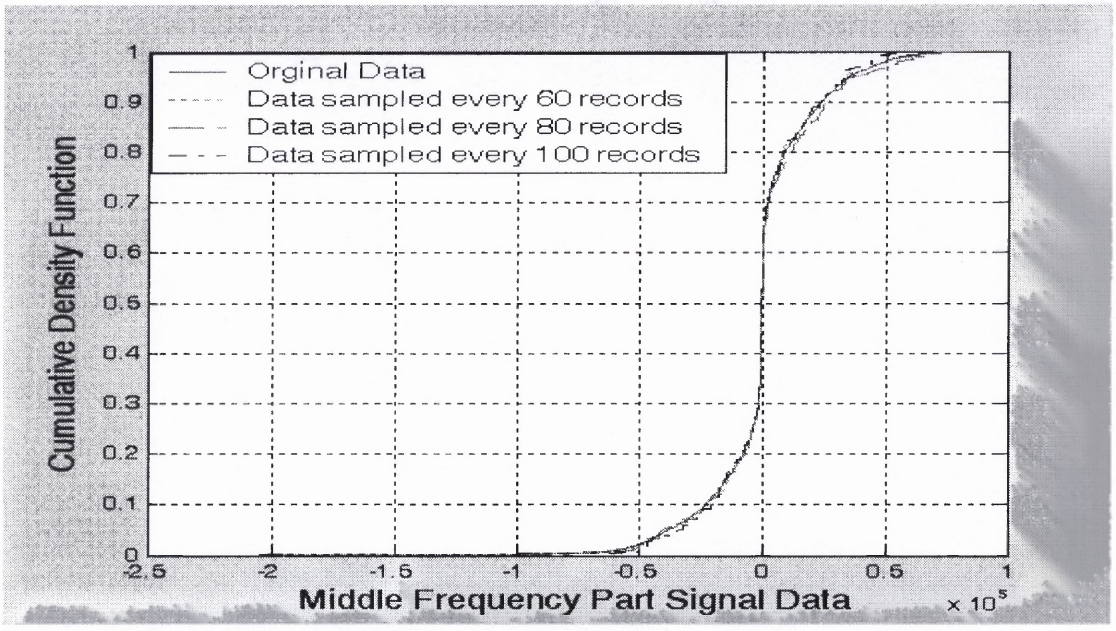

Figure 6.14 The CDF plots of the original and sampled middle frequency part signal data for the data trace MF-1.

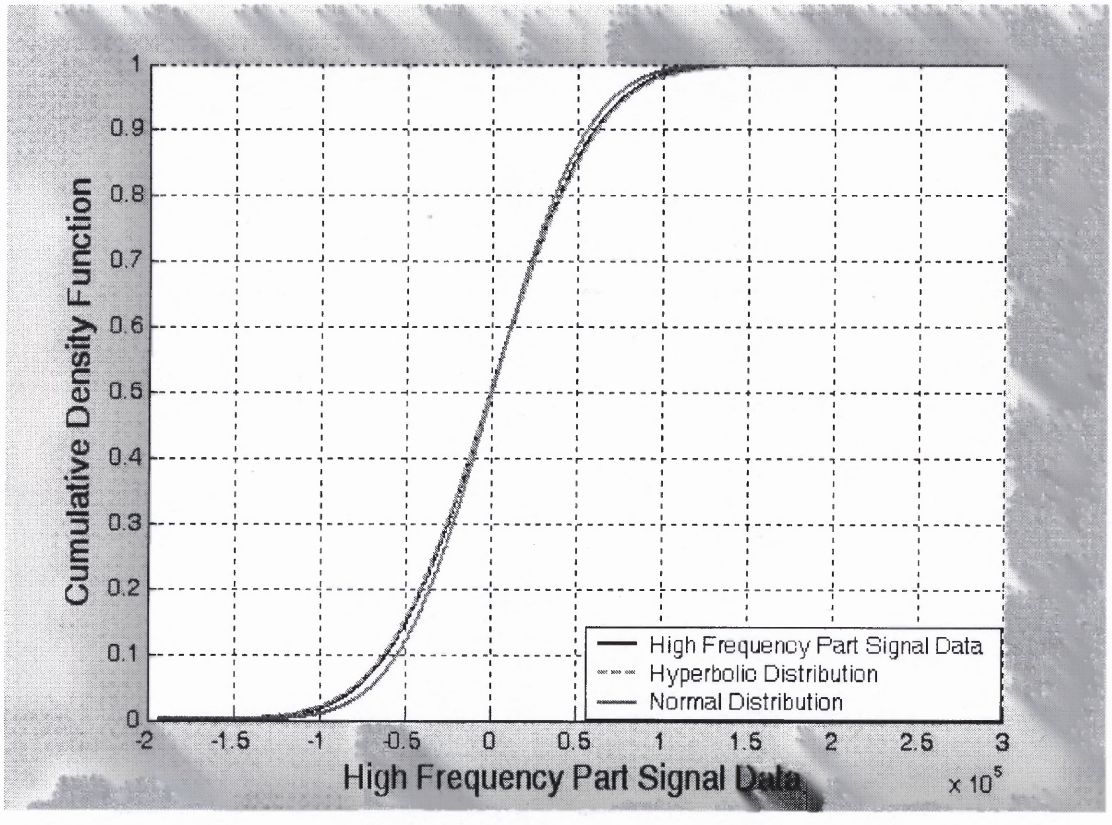

Figure 6.15 The fitting results for the high frequency part signal for the data trace MF-1.

The fitting results for modeling the high frequency part signal are shown in Table 6.10. The first column gives the indices of the data traces gathered from the BCCLNS. The second and third columns give the fitting results for the normal distribution, i.e., the calculated K-S distance, as well as the associated probability that the data derives from the fitted model distribution; while the last two columns present the fitting results for the hyperbolic distribution.

From the results shown in Table 6.10, one can clearly see that although in some cases normal distribution can achieve very small K-S distances, the K-S distances are not small enough to get large associated probability values. Often, the K-S distances computed are large and the associated probability values are negligible. With regard to the hyperbolic distribution, it is observed that consistently the K-S distances are small and the associated probability values are large. In some cases, the associated probability values approach 1 closely, which implies high confidence that the data derive from the hyperbolic distribution model. Moreover, its performance is very stable. Going over the results listed in this table, one cannot find any case in which the associated probability value of the hyperbolic distribution is less than 0.25. Visually as well, as shown in Figure 6.15 where a sample of fitting results for the data trace MF-1 is present, the hyperbolic distribution fits the data very well, it is clearly seen that the curve of the hyperbolic distribution not only matches the main body of the data quite well, but does so for the lower and upper tails as well.

6.4 Summary

In this chapter, a frequency based network traffic variable modeling scheme is introduced, that is, dividing the full process of the traffic variable observations into three parts according to their frequencies, and then modeling each part signal separately. The analysis

results demonstrated that the low and middle frequency part signal processes are non-stationary and should not be modeled by a random process or distribution. Instead, they can be well modeled by their sampled data. On the contrary, the high frequency part signal is more consistent with a first-order stationary process. The one-dimensional hyperbolic distribution is proposed as an effective statistical model for the high frequency part signal of the traffic variable observations. The results showed that the hyperbolic distribution provides a significantly better fit, and thus a more efficient model than the normal distribution.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This dissertation is divided into two parts. First, a hierarchical, multi-tier, multiple-observation-window, statistical based Network Anomaly Detection System (NADS), i.e., the MIB Anomaly Detection (MAD) system, is proposed, which is capable of detecting and diagnosing the network anomalies proactively and adaptively using only Management Information Base (MIB) II supplied traffic related variables. Extensive simulation experiments of network anomaly have been carried out, the corresponding numerical results demonstrate that MAD is very efficient and can reliably detect the network anomaly with anomaly traffic intensity as low as one percent of the typical background traffic intensity.

In the design of MAD, seven partition schemes as well as seventeen prominent and/or promising similarity measurement metrics, applied to each MIB variable, are also proposed. The results for evaluating the partition schemes indicate that all the seven partition schemes perform well and achieve roughly comparable misclassification rate. The results for evaluating the similarity measurement algorithms show that CST1, CST2, KS1, KS3, WSS1, KSS1 and FD2 perform significantly better than other metrics and should be the best choices for MAD system.

The classifier of MAD, while well trained in a test network, is ill-trained in an unfamiliar network setting. This is because while typical background traffic data is available in the new network, anomaly data usually is not. To solve this problem, two

techniques have been designed, i.e., the re-use classifier and the grafted classifier methods. Several experiments that compare the baseline outcomes to the results for the grafted and re-use classifier methods at various background patterns and network topologies have been carried out. The classification results show that the re-use classifier method can achieve modest performance, but cannot perform reliably. However, the classification performance of the grafted classifier methods is quite satisfactory, being only a little inferior to the baseline results. This indicates that the grafted modeling and estimation algorithms can be used as effective starting points, when migrating a statistical NADS from the test network to an unfamiliar network setting. MAD's adaptation and learning algorithm can then bootstrap the system from adequate anomaly models to excellent ones. Additional experimentation with a greater variety of test-target network combinations, in both simulated and real network environments, will be helpful in further investigating the applicability of these two methods.

In this dissertation, the low pass filter is introduced in the statistical based NADS design to reduce the false alarm rate. Several low pass filters are investigated, including the MWA filter, Savitzky-Golay filter and four variants of the 4^{th} order Butterworth filters with different cutoff frequencies. Through analyzing the network traffic measurements after filtering by these low pass filters and further applying these filters to the operation of MAD, it is found out that the Savitzky-Golay filter and Butterworth filter with cutoff frequency 0.2 should be the better choices of low pass filter that may be used in NADS to eliminate traffic bursts and further reduce the false alarm rate.

In the second part of this dissertation, a frequency based network traffic statistical modeling scheme is introduced. The traffic variable measurements are cut into three

divisions by the 4^{th} order Butterworth low pass filter according to their frequency spectrums, and then each part signal is modeled separately. The numerical results demonstrated that the low and middle frequency part signals could be well modeled by their sample data. To model the high frequency part signal, a new effective network statistical model is introduced, i.e., the one-dimensional hyperbolic distribution. The results showed that the hyperbolic distribution provides a significantly better fit, and thus, a more efficient model than the normal distribution.

7.2 Future Work

In addition to the coverage in this dissertation of introducing a new statistical based network anomaly detection system design and a frequency based network traffic statistical modeling scheme, the following issues should be further studied:

- Integrating the proposed frequency based network traffic statistical modeling scheme into the design of MAD system.
- Investigating the performance of MAD system on detecting reconnaissance attacks.

REFERENCES

1. P. Barford, J. Kline, D. Plonka, and R. Amos, "A signal analysis of network traffic anomalies," In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, (Marseilles, France), Nov. 2002.

2. M. Thottan and C. Ji, "Proactive Anomaly Detection Using Distributed Intelligent Agents", *IEEE Network, Special Issue on Network Management*, Sep./Oct. 1998.

3. M. A. Miller, "*Management Internetworks with SNMP*", third version, M&T Books, IDG Books Worldwide, Inc., Foster City, CA

4. G. Jakobson, M. D. Weissman, "Alarm correlation," IEEE Network, pp 52-59, Nov. 1993.

5. T. D. Ndousse and T. Okuda, "Computational intelligence for distributed fault management in networks using fuzzy cognitive maps," in *Proceedings of IEEE ICC*, pp 1158-1562, (Dallas TX), June, 1996.

6. C. Hood and C. Ji, "Proactive network fault detection," *IEEE Trans. Reliability*, vol. 46, no.3, 333-341, Sep. 1997.

7. J. Huard, and A. A. Lazar, "Fault isolation based on decision-theoretic troubleshooting," Technical Report. TR 442-96-08, Center for Telecommunications Research, Columbia University, 1996.

8. H. Li and J. S. Baras, "A framework for supporting intelligent fault and performance management for communication networks", in *Proceedings of IEEE/IFIP MMNS 2001*, pp. 227-240, (Chicago, IL), Oct. 2001.

9. D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines – a survey," *IEEE Trans Computers*, vol. 84, pp. 1090-1123, Aug. 1996.

10. A. Bouloutas, G. W. Hart and M. Schwartz, "Simple finite-state fault detectors for communication networks," *IEEE Trans. Communications*, vol. 40, no. 3, Mar. 1992.

11. A. Bouloutas, G. W. Hart and M. Schwartz, " Fault identification using a finite state machine model with unreliable partially observed data sequences,", IEEE Trans. Communications, vol. 41, no. 7, July 1993.

12. C. Wang and M. Schwartz, "Fault detection with multiple observers", *IEEE Trans. Networking*, vol. 1, no. 1, Feb. 1993.

13. I. Rouvellou and G. W. Hart, "Automatic alarm correlation for fault identification", In *Proceedings of IEEE INFOCOM*, pp. 553-561, June1995.

14. I. Katzela, M. Schwartz, "Schemes for fault identification in communication networks", *IEEE/ACM Trans Networking*, vol. 3, no. 6, Nov. 1995.

15. A. A. Lazar, W. Wang, R. Deng, "Models and algorithms for network fault detection and identification: a review," in *Proceedings of the International Conference on Communications*, (Singapore), Nov. 1992

16. L. Ho, D. Cavuto, S. Papavassiliou and A. Zawadzki, "Adaptive anomaly detection in transaction-oriented networks," *Journal of Network and Systems Management*, vol. 9, no. 2, pp.139-159, June 2001.

17. L. Ho, D. Cavuto, S. Papavassiliou and A. Zawadzki, "Adaptive/automated detection of service anomalies in transaction WANs: network analysis, algorithms, implementation, and deployment", *IEEE Journal on Selected Areas in Communications (JSAC)*, vol.18, no. 5, pp. 744–757, May 2000.

18. V. Paxson, "Empirical derived analytic models of wide-area TCP connections," *IEEE Trans Networking*, vol. 2, no. 4, Aug. 1994.

19. K. Thompson, G. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, pp. 10–23, Nov./Dec. 1997.

20. R. A. Maxion and F. E. Feather, "A case study of ethernet anomalies in a distributed computing environment," IEEE Trans Reliability, vol. 39, no. 4, Oct. 1990.

21. F. E. Feather, D. Siewiorek and R. Maxion, "Fault detection in an ethernet using anomaly signature matching," *ACM SIGCOMM*, vol. 23, no. 4, 1993.

22. M. Thottan and C. Ji, "Adaptive thresholding for proactive network problem detection," In *Proceedings of IEEE International Workshop on Systems Management*, (Newport, RI) 1998.

23. M. Thottan and C. Ji, "Fault prediction at the network layer using intelligent agents," in *Proceedings of 6th IEEE/IFIP International Symposium Integrated Network Management*, pp. 745-759, 1999.

24 M. Thottan and C. Ji, "Anomaly detection in IP networks" *IEEE Trans. Signal Processing, Special Issue of Signal Processing in Networking,* vol. 51 issue 8, pp. 2191 -2204, Aug. 2003.

25. J. B. D. Cabrera, B. Ravichandran and R. K. Mehra, "Statistical traffic modeling for network intrusion detection," in Proceedings of 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (San Francisco, Ca), Aug. 2000.

26. J. L. Hellerstein, F. Z, P. Shahabuddin, "An Approach to Predictive Detection for Service Management," *Symposium on Integrated Net-work Management*, 1999.

27. P. Hoogenboom, J. Lepreau, "Computer System Performance Detection Using Time Series Models ," in *Proceedings of the Summer USENIX Conference*, pp.15-32, 1993.

28. M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE Trans. Networking*, vol. 5, no. 6, Dec. 1997.

29. W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the self-similarity nature of Ethernet traffic (extended version)," *IEEE Trans. Networking*, vol. 2, pp 1-15, Feb. 1994.

30. W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of ethernet LAN traffic at the source level", *IEEE Trans. Networking*, vol. 5, no. 1, Feb. 1997

31. R. V. Mises, "*Mathematical Theory of Probability and Statistics*," Chapters IX(C) and IX(E), Academic Press, New York.

32. M. A. Stephens, "Statistical description of data," *Journal of the Royal Statistical Society*, ser. B, vol. 32, pp. 115-122. 1970.

33. Y. Burnod, "*An Adaptive Neural Network*," Prentice Hall, London, 1990.

34. J. Case, K. McCloghrie, M. Rose and S. Waldbusser, "Protocol operations for version 2 of the Simple Network Management Protocol (SNMPv2)," *RFC 1448*, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

35. K. McCloghrie, and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II," *STD 17, RFC 1213*, Hughes LAN Systems, Dover Beach Consulting, Inc., March 1991.

36. J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)," *RFC 1442*, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

37. J. D. Case and C. Partridge, "Case diagrams: a first approach to diagrammed management information bases," *Computation Communication Review*, vol. 19, pp. 13-16, Jan 1989.

38. A. Valdes, D. Anderson, "Statistical methods for computer usage anomaly detection using NIDES" *Technical Report*, SRI International, Jan. 1995.

39. J. B.D. Cabrera, B. Bavichandran, R.K. Mehra, "Statistical traffic modeling for network intrusion detection" in *Proceedings of 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 466-473, Aug. 2000.

40. R. V. Mises, *Mathematical Theory of Probability and Statistics*, Academic Press, New York, 1964.

41. M. A. Stephens, "EDF statistics for goodness of fit and some comparisons," *Journal of the American Statistical Association*, vol. 69, pp. 730-737, 1974.

42. D. A. Darling, "Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes," *Annals of Mathematical Statistics*, vol. 28, pp. 823–838. 1957

43. N. I. Fisher, T. Lewis, and B. J. J. Embleton, *Statistical Analysis of Spherical Data* Cambridge University Press, New York, 1987.

44. A. K. Ghosh, J. Wanken, F. Charron, "Detecting anomalous and unknown intrusions against programs," in *Proceedings of IEEE 14th Annual Computer Security Applications Conference*, pp. 259 –267, Aug. 1998.

45. S. Jiang, D. Siboni, A.A. Rhissa, G. Beuchot, "An intelligent and integrated system of network fault management: artificial intelligence technologies and hybrid architectures", in *Proceedings of IEEE Singapore International Conference on Networks*, pp. 265-268, July 1995.

46. J. Beran, *Statistics for long-memory processes monographs on statistics and applied probability*, Chapman and Hall, New York, NY, 1994

47. W. Willinger, M. S. Taqqu, and A. Erramilli. "A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks," Stochastic Networks: Theory and Applications, volume 4 of Royal Statistical Society Lecture Notes Series, Oxford University Press, 1996.

48. L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in *Proceedings of ACM Workshop on Data Mining for Security Applications*, pp. 1–14, Nov. 2001.

49. V. Paxon and S. Floyd, "Wide-area traffic: the failure of poisson modeling", *IEEE Trans. Networking*, vol. 3, no. 3, pp. 226-244, June 1995.

50. S. Deng, "Empirical model of WWW document arrivals at access link", in *Proceedings of the IEEE International Conference on Communications*, pp. 483-489, June 1997.

51. P. Barford; M. Crovella, "Generating representative web workloads for network and server performance evaluation", HTTP://cs-pub.bu.edu/techreports/1997-006-surge.ps.Z.

52. M. Molina, P. Castelli and G. Foddis, "Web traffic modeling, exploiting TCP Connections' temporal clustering through HTML-REDUCE", *IEEE Network*, May/June 2000.

53. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical recipes in c, the art of scientific computing*, Second Edition, Cambridge University Press, New York.

54. T. W. Parks and C. S. Burrus, *Digital Filter Design*, John Wiley and Sons, 1987.

55. R. W. Hamming, *Digital Filters*, 2nd edition, Prentice-Hall, Englewood Cliffs, NJ, 1983.

56. H. Ziegler, *Applied Spectroscopy* vol. 35, pp. 88-92, 1981.

57. A. Savizky and M. J. E. Golay, *Analytical Chemistry*, vol. 36, pp. 1672-1639, 1964.

58. I. W. Selesnick and C. S. Burrus, "Generalized digital Butterworth filter design," In Proceedings of IEEE International Conference on Acoustic, Speech, Signal Processing, May 1996.

59. M. Arlitt and T. Jin, "Workload characterization of the 1998 World Cup web site", Technical report HPL-99-35R1, Hewlett-Packard Labs, September 1999.

60. P. Barford, "Changes in web client access patterns," World Wide Web Journal, Special Issue on Characterization and Performance Evaluation, 1999.

61. M. Arlitt and C. Williamson, "Internet web servers: workload characterization and performance implications" IEEE/ACM Transaction on Networking, Vol. 5, No. 5, pp. 631-645, October 1997.

62. O. E. Barndorff-Nielsen, "Exponentially decreasing distributions for the logarithm of particle size", *Proc. Roy. Soc. London Ser.* A 353, pp. 401-419, 1977.

63. O. E. Barndorff-Nielsen, "Hyperbolic distributions and distributions of hyperbolae" *Scand. J. Statist.* 5, pp. 151-157, 1978.

64. J. Huntley, J. Jorgenson, and R. Lundelius, "On the asymptotic behavior of counting functions associated to degenerating hyperbolic Riemann surfaces", *Journal of Functional Analysis*, 149, pp. 58-82, 1997.

65. Y. Y. Lee, "Internet traffic over wide area network -- statistical modeling and analyze," Master Thesis, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, 2003.