ABSTRACT

DETECTION OF NETWORK ANOMALIES AND NOVEL ATTACKS IN THE INTERNET VIA STATISTICAL NETWORK TRAFFIC SEPARATION AND NORMALITY PREDICTION

by
Jun Jiang

With the advent and the explosive growth of the global Internet and the electronic commerce environment, adaptive/automatic network and service anomaly detection is fast gaining critical research and practical importance. If the next generation of network technology is to operate beyond the levels of current networks, it will require a set of well-designed tools for its management that will provide the capability of dynamically and reliably identifying network anomalies. Early detection of network anomalies and performance degradations is a key to rapid fault recovery and robust networking, and has been receiving increasing attention lately.

In this dissertation we present a network anomaly detection methodology, which relies on the analysis of network traffic and the characterization of the dynamic statistical properties of traffic normality, in order to accurately and timely detect network anomalies. Anomaly detection is based on the concept that perturbations of normal behavior suggest the presence of anomalies, faults, attacks etc. This methodology can be uniformly applied in order to detect network attacks, especially in cases where novel attacks are present and the nature of the intrusion is unknown.

Specifically, in order to provide an accurate identification of the normal network traffic behavior, we first develop an anomaly-tolerant non-stationary traffic prediction technique, which is capable of removing both pulse and continuous anomalies. Furthermore we introduce and design dynamic thresholds, and based on them we define adaptive anomaly violation conditions, as a combined function of both the magnitude and duration of the traffic deviations. Numerical results are presented that demonstrate the operational effectiveness and efficiency of the proposed approach, under different anomaly traffic scenarios and attacks, such as mail-bombing and UDP flooding attacks.

In order to improve the prediction accuracy of the statistical network traffic normality, especially in cases where high burstiness is present, we propose, study and analyze a new network traffic prediction methodology, based on the ”frequency domain” traffic analysis and filtering, with the objective of enhancing the network anomaly detection capabilities. Our approach is based on the observation that the various network traffic components, are better identified, represented and isolated in the frequency domain. As a result, the traffic can be effectively separated into a baseline component, that includes most of the low frequency traffic and presents low burstiness, and the short-term traffic that includes the most dynamic part. The baseline traffic is a mean non-stationary periodic time series, and the Extended Resource-Allocating Network (ERAN) methodology is used for its accurate prediction. The short-term traffic is shown to be a time-dependent series, and the Auto-regressive Moving Average (ARMA) model is proposed to be used for the accurate prediction of this component. Furthermore, it is demonstrated that the proposed enhanced traffic prediction strategy can be combined with the use of dynamic thresholds and adaptive anomaly violation conditions, in order to improve the network anomaly detection effectiveness. The performance evaluation of the proposed overall strategy, in terms of the achievable network traffic prediction accuracy and anomaly detection capability, and the corresponding numerical results demonstrate and quantify the significant improvements that can be achieved.

DETECTION OF NETWORK ANOMALIES AND NOVEL ATTACKS IN THE INTERNET VIA STATISTICAL NETWORK TRAFFIC SEPARATION AND NORMALITY PREDICTION

by
Jun Jiang

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

January 2005

APPROVAL PAGE

DETECTION OF NETWORK ANOMALIES AND NOVEL ATTACKS IN THE INTERNET VIA STATISTICAL NETWORK TRAFFIC SEPARATION AND NORMALITY PREDICTION

Jun Jiang

Dr. Symeon Papavassiliou, Dissertation Advisor — Date
Associate Professor of Electrical and Computer Engineering , New Jersey Institute of Technology

Dr. Nirwan Ansari, Committee Member — Date
Professor of Electrical and Computer Engineering , New Jersey Institute of Technology

Dr. Teunis Ott, Committee Member — Date
Professor of Computer Science, New Jersey Institute of Technology

Dr. Lev Zakrevski, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering , New Jersey Institute of Technology

Dr. Roberto Rojas-Cessa, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering , New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author:	Jun Jiang
Degree:	Doctor of Philosophy
Date:	January 2005

Undergraduate and Graduate Education:

- Master of Science in Electrical and Computer Engineering,
  Beijing University of Posts and Telecommunications, Beijing, China April,1998
- Bachelor of Science in Electrical and Computer Engineering,
  Beijing University of Posts and Telecommunications, Beijing, China July,1995

Major: Electrical Engineering

Presentations and Publications:

Jiang, Jun and Papavassiliou, Symeon
"Detection of Novel Attacks in the Internet via Statistical Network Traffic Normality Prediction"
Journal of Network and System Management, Vol. 12, No. 1, pp. 51-72, March 2004 .

Jiang, Jun and Papavassiliou, Symeon
"A Network Fault Diagnostic Approach Based on a Statistical Traffic Normality Prediction Algorithm,"
*Proc. of IEEE Global Communications Conference (GLOBECOM 2003),*Vol. 5 , pp. 2918 - 2922, December 2003.

Jiang, Jun and Papavassiliou, Symeon
"Providing End-to-End Quality of Service with Optimal Least Weight Routing in Next Generation Multi-service High-Speed Networks,"
Journal of Network and Systems Management, Vol.10, No.3, pp. 281-309, September 2002.

Jiang, Jun and Papavassiliou, Symeon
"Providing End-to-End Quality of Service with Optimal Least Weight Routing in Next Generation Multi-service High-Speed Networks,"
Proc. IEEE Symposium on Computers and Communications (ISCC2002), pp. 449-454, July 2002.

Jiang, Jun and Papavassiliou, Symeon
"Optimal Least Weight Routing Algorithm for Guaranteed Bandwidth Flows,"
Proc. of Conference on Information Sciences and Systems (CISS2001), pp. 564-569, March 2001.

Jiang, Jun and Papavassiliou, Symeon
"Enhancing Network Traffic Prediction and Anomaly Detection via Statistical Network Traffic Separation and Combination Strategies,"
submitted to *Computer Communications Journal*

Dedicated to my parents and my beloved wife

ACKNOWLEDGMENT

First and foremost, I would like to express my sincere appreciation to my advisor and mentor, Professor Symeon Papavassiliou, for his continuous support, encouragement and invaluable suggestions in every step of my way. I deeply appreciate his advice, guidance, and the academic atmosphere and freedom he brought to our research group, and treasure the opportunities he created for me to improve my research and professional skills. His technical and editorial advice was essential to the completion of this dissertation.

I am also deeply grateful to Professor Nirwan Ansari for his support and encouragement throughout my Ph.D research. I also acknowledge the valuable comments and discussion with my committee members: Professor Teunis Ott, Professor Roberto Rojas-Cessa and Professor Lev Zakrevski. I would like to give thanks for their suggestions and helpful reviews on my dissertation, and for the academic insight and motivation they gave me.

I am grateful to the support from New Jersey Center for Wireless Network and Internet Security and New Jersey Multimedia Center.

The friendship of Jie Yang, Jian Ye, Zheng Zhang and Jun Li is much appreciated and has led to many interesting and good-spirited discussions relating to this research. I am also grateful to my colleagues Sheng Xu, Chengzhou Li, Jin Zhu and Didem Gozupek for their help.

Finally, I would like to thank my beloved wife, Yin Chen, for her understanding and love during the past few years. I express my special gratitude to my parents, Qisun Jiang and Yuehua Wang, for their dedicated and endless love. They supported me with their best in every aspect. Without them, I would not have achieved anything that I achieved today. I wish them the very best from the bottom of my heart.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure Page

CHAPTER 1

INTRODUCTION

Today's networks have become highly heterogeneous and vary greatly in the services they offer and the traffic they carry. While network heterogeneity provides more flexibility in utilizing the latest technologies and allows for customization by user applications, it also increases the risk of the occurrence of network anomalies [1, 2]. However if the next generation of network technology is to operate beyond the levels of current networks, it will require a set of well-designed tools for its management, that will provide the capability of dynamically and reliably identifying and correcting network faults and anomalies. It has been also demonstrated that network perform- ance monitoring is essential for managing a network efficiently and for ensuring reliable operation of the network [3, 4, 5].

As networks and their performance/utilization dynamics are becoming exponentially more complex and the sources and scope of network anomalies multiplying, human network operators find it increasingly difficult to analyze and recognize network faults and/or performance degradations in real-time, and to do it accurately and reliably. In the near future, given the stringent network reliability requirements, it may well be impossible, even in theory, to rely much on human operators for detection, diagnosis, and network planning. Thus, algorithms and methodologies that are capable of detecting and diagnosing performance degradations and/or network anomalies automatically are in high demand.

1.1 Network Anomalies

Network Anomalies mainly fall in two categories: Network/Service Faults and Network Intrusions. Throughout this dissertation we will use the term network anomalies to represent both network/service faults and network intrusions, since in most cases, if they persist, they all present abnormal and unexpected events that lead to significant performance degradations which eventually affect negatively the network behavior and the users' quality of service (QoS) requirements.

1.1.1 Network/Service Faults

In general an anomalous event is a surprising event [6] that may present some deviation of some performance metric from its corresponding expected value, and thus may be considered as anomalous. Network anomalies/faults are exceptional conditions in network protocol layers that impact negatively the performance and the functioning of networked applications and services, hence leading to service exceptional conditions, namely service faults. These exceptional conditions can be hard faults (severe outages and failures), or soft faults (performance degradations) such as broadcast storm in a shared Ethernet, etc. In this dissertation we refer to the detection of soft faults which are more difficult to be detected by traditional and conventional network management systems. Detection of hard faults, such as link cut and power failure, are usually engineered as element alarms/traps by equipment manufacturers, and are detected and diagnosed by fault diagnosis systems such as alarm correlation systems. Furthermore, issues associated with the network/service fault detection become even more critical in multiple service-class and wide-area communication infrastructures (such as next generation Internet supporting multimedia services), where performances of service classes are mutually dependent and strongly correlated.

1.1.2 Network Attacks/Intrusions

There are several types of attacks in the Internet that may range from information leakage, to routing table poisoning attacks, to packet mistreatment, to Denial of Service (DoS), etc. [7]. Some of these attacks may affect a single user, while others may affect the performance of a large group of users or classes of service. In this dissertation we mainly emphasize on the detection of attacks and/or intrusions that fall in the latter category, since in general they present an impact on the performance of the whole network, or of a significant part of it. Such an attack for instance is the DoS attack. These attacks become extremely dangerous and very hard to prevent, especially when a group of attackers coordinate in DoS [7, 8]. In addition to intentional direct DoS attacks against specific servers or hosts, it should also be noted that several other attacks against the transmission infrastructure, such as routing table poisoning and packet mistreatment, may result in massive DoS attacks against entire groups or whole portions of the Internet. Common types of DoS attacks include: a) UDP flooding where large number of UDP packets with spoofed return address are launched; b) TCP/SYN Flooding where the attacker may send a large volume of SYN packets to a victim with spoofed return address and thus the victim queues up SYN-ACKs but cannot continue its operation since it never receives ACKs from the spoofed addresses; and c) ICMP/Smurf where the hacker broadcasts ICMP ping requests with the returned address spoofed to show the victim's address [7, 8].

In recent years all the attacks have been significantly gaining in sophistication and power to harm. Attacks are increasingly automated, so that now the attack tools may initiate new attack cycles by themselves, with no person involved. Distributed attack tools are capable of coordinating the use of numerous attack platforms and scripts spread out through the Internet, thus launching truly devastating DoS attacks.

1.2 Detecting Network Anomalies

Currently known security techniques, such as authentication and encryption, although have significantly contributed to the data transfer protection, can not provide absolute security, neither can prevent users from misbehaving and thus degrading the network performance and security. A lot of attacks are simply based on software flaws and design errors [9]. Therefore security of computer systems and networks must be included and performed, at all levels, top-to-bottom. With the increasing complexity of the networks the task of detecting and preventing the network abuse becomes more and more difficult. Therefore additional countermeasures are required.

In figure 1.1 [10], the main approaches that have been proposed and developed for the detection of potential network anomalies, are presented.

One of the most common approaches is the misuse detection (or signature-based detection), which is based on the use of ”signatures” of known attacks (i.e. the patterns of attack behavior or effects) to identify a matched activity as an attack instance. Several systems (such as Network Security Monitor [11, 12], NSTAT [13] and BRO [14]) have been developed based on this approach due to its implementation simplicity and the low computational resources required. However, in practice, this approach may result in high false-alarm probability because it is difficult to select the appropriate signatures that can successfully detect the real anomalies without creating false alarms for normal traffic. More importantly, the signature-based detection techniques are not effective against new attacks which do not present well known signatures.

Therefore, in order to be able to detect new/novel anomalies in real networking environments, several strategies have been proposed. For instance, the bottleneck verification [15] approach has been introduced when there are only a few, well defined ways to transition between two groups of states. An illegal transition between the two groups of states, which did not exist when the bottleneck verification system was created, can be detected

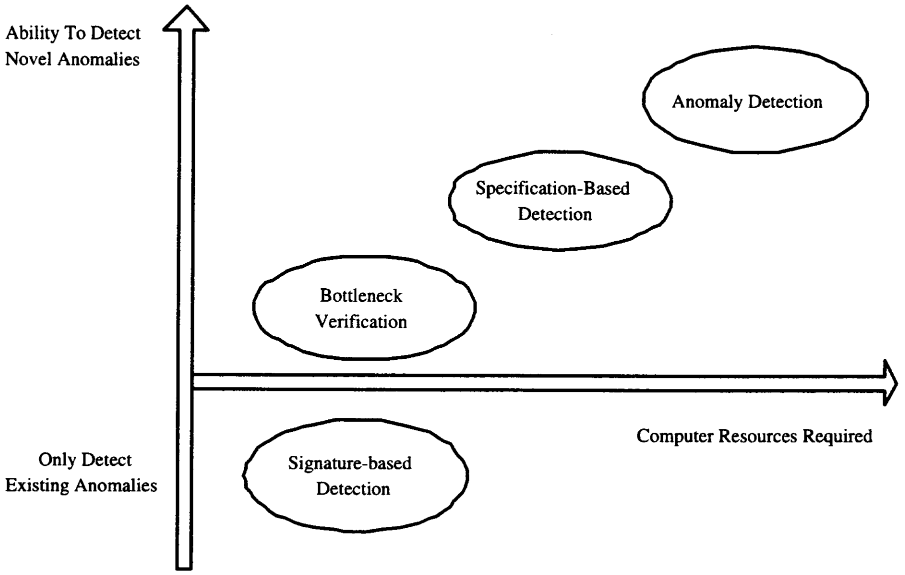

Figure 1.1 The strategies for anomaly detection

as potential anomaly. However this method is not scalable and can only be implemented in limited scenarios.

Specification-based anomaly detection [16] can detect anomalies that make improper use of system or application programs. This method requires the pre-defined security specifications that describe the normal intended behavior of programs. Once these pre-defined specifications are violated, potential anomalies can be detected. Although, this approach has the potential to provide very low false alarm probability and detect a wide range of anomalies, it is unfortunately very difficult to be implemented in practice, because the pre-defined security specifications have to be written for all monitored programs that are constantly updated.

Therefore, anomaly detection is one of the most frequently suggested methods to detect novel anomalies. Anomaly detection is based on the concept that perturbations of normal behavior suggest the presence of anomalies, faults, defects, attacks etc. It develops and identifies normal behavior patterns and profiles (i.e. the expected behavior), in order to identify any unacceptable and significant deviations from the usual behavior, as possibly a network anomaly. Some of the advantages of this technique is that in principle it is not restricted to any specific environment, and that can provide a way of detecting both network faults as well as network attacks, especially in cases where novel attacks are present where the nature of the intrusion is unknown.

1.3 Research Challenges and Motivations

The main objective of this dissertation is to develop a framework, the corresponding algorithms and a novel model that provide accurate statistical network traffic normality prediction methodologies, and based on them to detect network anomalies or performance degradations proactively and adaptively. Such framework and techniques can be used to improve the operational and functional capabilities of the Internet by facilitating many processes such as network management, network security, network data analysis, etc.

In our research work we adopt the anomaly detection approach. However, there are still a lot of issues and challenges associated with the development of efficient and effective anomaly detection methodologies. As mentioned before, one main component of these techniques is the capability of accurately predicting, analyzing and identifying the normality in the system and user behavior. This task is complicated by the following factors:

a) Users may slowly change their behavior with the system and time evolution (e.g. the traffic in a network may present changes and variations), and therefore any associated algorithm should be capable of dynamically and adaptively evaluating and determining the network normality, therefore minimizing and if possible eliminating the possibility of falsely identifying as an anomaly a new legitimate behavior;

b) The analysis and prediction about the expected system normality should be based on normal data, while all the data that may be anomalous and may affect the accuracy and correctness of the normal behavior prediction/identification should be excluded. In this case an anomaly-tolerant methodology is required;

c) In current communication networks, traffic has shown the self-similarity features and characteristics of black and brown noise [17]. The non-stationery signals originate from the complex dynamic behavior of underlying dynamic systems, which should be best described in terms of some nonlinear differential equation. However, in most cases, the particular differential equation is either completely unknown or very difficult to estimate. Therefore the accurate network traffic prediction is further complicated by the network noise and the occurrence of anomalous traffic.

Furthermore in order to properly execute anomaly detection it is required to introduce and define some threshold values that denote the network traffic normality, and based on these thresholds the corresponding anomaly violation conditions may be defined. The challenge in this case concerns the definition of the corresponding threshold values. This is further complicated by the well-known fact that performance variables (such as utilization) of networks undergo cyclic evolution and temporal fluctuation [18]. Therefore the use of dynamic and adaptive thresholds are required, so that based on them anomaly violation conditions may be defined as combined functions of magnitude and duration of the traffic deviations.

1.4 Dissertation Contributions and Outline

In this dissertation motivated by the aforementioned observations, we propose a framework and a novel network anomaly detection approach based on a statistical traffic normality prediction. Specifically, first in Chapter 2 an anomaly detection algorithm, namely the Extended Resource- Allocating Network (ERAN), is proposed based on the Resource-Allocating Network (RAN) [19]. The proposed approach presents several improvements and enhance- ments, including the development of an anomaly-tolerant non-stationary traffic prediction algorithm and the introduction of dynamically defined thresholds. The designed algorithm can provide accurate traffic prediction by removing both single pulse noise anomalies as well as continu- ous anomalies. We also introduce dynamic thresholds and threshold violation conditions that can be used to detect the anomaly traffic resulting from an attack, timely and accurately. Our main objective is to use such a framework and techniques in order to improve the security related operational and functional capabilities of the Internet, especially in the presence of novel attacks. In this chapter, we also provide an extensive evaluation of our proposed framework and its effectiveness for intrusion detection, under various test case scenarios, such as the mail bombing attack and the UDP flooding attack.

Furthermore analyzing various Internet traffic samples, we can observe that in many cases there are some underlying trends regarding the corresponding traffic patterns (e.g. daily, weekly patterns, etc.), which are mainly caused by different temporal and in some cases physical phenomena. These traffic components are mainly located in the low frequency area in the "frequency domain", while the dynamic part of the Internet traffic is mainly located in the high frequency. In addition, it was demonstrated in [20, 21] that the ERAN algorithm can provide accurate prediction for non-stationary time series traffic, mainly in cases with low burstiness. However, in the high burstiness scenarios (such as the expected real traffic in Internet) the corresponding prediction errors may increase due to the very dynamic characteristics of the traffic.

Therefore, in chapter 3 of the dissertation, we emphasize on the design and development of enhanced strategies that can be used to further improve the accuracy of the prediction of the network traffic normality, and as a result of the overall anomaly detection methodology, especially in cases where high burstiness is present. We first propose a methodology that provides effective traffic separation based on "frequency domain" data analysis and filtering. Specifically we separate the traffic into two main components: the baseline component and the short-term component. The baseline component includes most of the low frequency and non-stationary traffic and presents low burstiness, while the short-term component includes the most dynamic part. Since the baseline traffic is a mean non-stationary periodic time series, the ERAN algorithm described in chapter 2, can be used for the accurate prediction of this part of the traffic. The short-term traffic is shown to be a time-dependent series, and the Autoregressive Moving Average (ARMA) model is proposed to be used for the accurate prediction of this component.

One of the key principles of our proposed methodology is that most of the non-stationary traffic is separated and included into the baseline component, and therefore the short-term traffic, as shown in this dissertation, can be well and accurately modeled using the ARMA model. Numerical results presented in this chapter demonstrate that the proposed methodology of separating the traffic based on the "frequency domain" analysis and predicting each component separately by the appropriate method, improves significantly the prediction accuracy of the total combined Internet traffic, which in turn improves the performance of the network anomaly detection as well.

Finally Chapter 4 concludes the dissertation.

CHAPTER 2

NETWORK ANOMALIES DETECTION VIA STATISTICAL NETWORK TRAFFIC NORMALITY PREDICTION

In this chapter, we propose and evaluate a new Network Anomaly Detection Algorithm, based on a statistical traffic normality prediction approach. The proposed algorithm consists of a novel anomaly-tolerant non-stationary traffic prediction mechanism and the introduction of dynamically defined thresholds and adaptive violation conditions. The proposed algorithm can provide accurate traffic prediction by removing single pulse noise anomalies as well as continuous anomalies. The remaining of this chapter is organized as follows. In section 2.1 we present some background work related to the proposed algorithm. In section 2.2 we describe the anomaly-tolerant non-stationary traffic prediction mechanism and in section 2.3 we introduce dynamically defined thresholds and anomaly detection conditions, in order to adaptively detect potential anomalies in the network. In section 2.4 we introduce some performance metrics that are used in order to evaluate the accuracy of the traffic prediction mechanism as well as the effectiveness of the overall anomaly detection methodology, while in sections 2.5 and 2.6 we provide the corresponding numerical results under different traffic patterns and anomalies.

2.1 Background and Related Work

One of the challenges and difficulties in developing efficient and accurate anomaly detection statistical based approaches is related to the non-stationary characteristics that the Internet traffic presents. Non-stationary time series are properly encountered in a variety of real world situations ranging from engineering to economics. For instance forecasting is common in several diverse areas such as weather prediction [22] and projections of future economic performances [23].

In current communication networks, traffic has shown the self-similarity features and characteristics of black and brown noise [17]. The non-stationery signals originate from the complex dynamic behavior of underlying dynamic systems, which should be best described in terms of some nonlinear differential equation. However, in most cases, the particular differential equation is either completely unknown or very difficult to estimate. Moreover the accurate network traffic prediction is further complicated by the network noise and the occurrence of anomalous traffic. These uncertain properties frequently restrict the traditional stochastic analysis to deliver a realistic estimation.

In the past decades modeling of non-stationary multivariate time series has always been a challenging topic [24, 25]. In order to model or predict the non-stationary signals, traditional statistical methods such as parzen windows approximation [26], and Artificial Neural Networks (ANN) such as the Radial Basis Function Network (RBFN) [27], have been proposed and implemented. The parzen windows approximation allocates a new unit for every learned sample to approximate the underlying distribution. The problem with such methods is their high computation complexity when the number of samples is very large. The artificial neural networks have been successfully used in the multivariate and non-linear time series prediction [28, 29]. However, traditional ANNs perform very poorly in non-stationary time series prediction because of the deficiency of feedback mechanism to accommodate the input distribution changes. Moreover, capacity should be reserved for all existing and future observations. For example, the RBFN method uses the centers of each basis function as the prototype of a local region, to memorize an abstraction of the data set explicitly. The number of hidden units could be smaller than the number of training samples. One of the main drawbacks of the traditional RBFN is that the number of centroids of RBFN has to be determined in advance. If the number of centroids is not determined properly, the accuracy of RBFN is seriously affected.

In summary, due to the time-variable characteristic of the non-stationary signals, fixed size neural networks may not predict the signals properly. If a neural network is very simple, it cannot reflect the complexity of real-world problems, and it has low accuracy. If the neural network is too complex, the generalization capability may suffer if sufficient training samples are not available, while the corresponding computation is very timeconsuming. To overcome some of these problems several solutions have been proposed in the literature by the formulation of online adaptive learning procedure for a variety of ANNs [19, 30, 31, 32].

In [19] the authors introduced a Resource-Allocating Network that allocates a new computational unit whenever an unusual pattern is presented to the network. This algorithm is suitable for sequential learning and is based on the idea that the number of hidden units should correspond to the complexity of the underlying function as reflected in the observed signals. RAN [19] was developed to overcome the problem of NP-completeness in learning with fixed size network. In that network, the computation time may either linearly or exponentially increase due to the complexity of the data. But in RAN, by allocating new resources, learning could be achieved in polynomial time. In RAN similar skills as in parzen window estimation are used, except that it improves this method by storing fewer observations which makes the size of RAN grow sub-linearly and eventually saturate. A detail description of the sequential learning procedure in the ANN which the RAN is derived from, is presented in appendix A.

2.2 Anomaly Tolerant Non-Stationary Traffic Prediction

In this section we introduce the RAN algorithm and describe in detail an anomaly-tolerant non-stationary traffic prediction algorithm based on the RAN algorithm.

2.2.1 Resource-Allocating Network

RAN can be simplified as a single hidden layer network whose output response to sequential inputs is a linear combination of the hidden unit responses (figure 2.1).

The overall network response function can be expressed as follows:

$$f(X) = \alpha_0 + \sum_{j=1}^{J} \alpha_j \phi_j(X) \tag{2.1}$$

where X is an M-dimensional input vector, α_0 is the bias term, α_j are the weights between the hidden and the output layers, and parameter J is the hidden unit number. Each hidden unit possesses a unit center. A unit passes the Euclidean distance between its center and the network input vector through a nonlinear function. In general in literature, the nonlinear function is selected as a Gaussian function [19]. The RAN hidden unit responses $\phi_j(X)$ are given by:

$$\phi_j(X) = \exp\left\{-\frac{1}{\sigma_j^2}\left\|X - u_j\right\|^2\right\} \tag{2.2}$$

where u_j is the unit center or mean of the Gaussian and σ_j is the spread of the neighborhood or width of the Gaussian, while $\|\|$ denotes the Euclidean norm. Here we represent the network response at time frame n as $f(X_n)$, while the corresponding actual observation output scalar is denoted by y_n. Here the input vector X_n is being defined as $X_n = [y_{n-1}, ..., y_{n-M}]$.

We start the prediction procedure without hidden units. The first observation is (X_0, y_0) , where y_0 is the target output scalar, and in the initial stage the coefficient α_0=y_0.

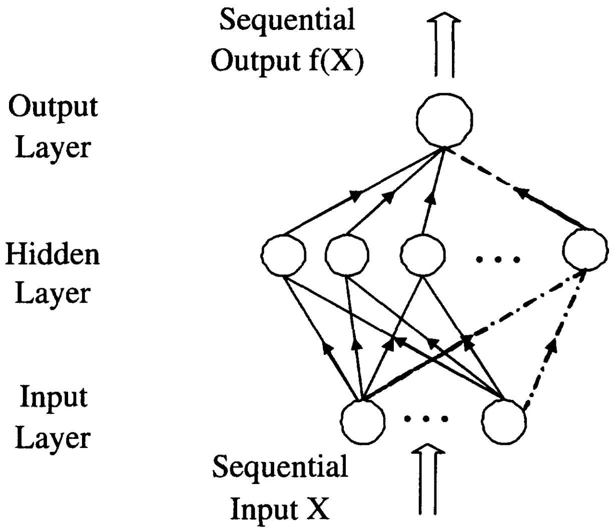

Figure 2.1 Logical diagram for Resource-Allocating Network

As observations are received the network grows by storing some of them and adding new hidden units. The decision to store an observation (X_n, y_n) depends on its novelty, as presented by the following two conditions:

$$\|X_n - u_{nj}\| > \epsilon_n; \quad e_n = y_n - f(X_n) > e_{\min} \tag{2.3}$$

where u_{nj} is the nearest stored pattern to X_n in the input space, threshold (distance) ϵ_n represents the scale of resolution in the input space, and the value $e_{\min}$ is chosen to represent the desired accuracy of the network output. The first condition says that the input must be far away from stored patterns and the second one says that the error between the network output and the target must be significant. When a new hidden unit is added to the network, the parameters and weights associated with this unit are assigned as follows:

$$\alpha_{J+1} = e_n; \quad \mu_{J+1} = X_n; \quad \sigma_{J+1} = \varphi \, \|X_n - u_{nj}\| \tag{2.4}$$

where φ is the overlap factor that determines the overlap of the responses of the hidden units in the input space. The value for the width σ_{J+1} is based on a nearest-neighbor heuristic. When the observation (X_n, y_n) does not satisfy the novelty criteria, we need to update the network parameters in order to minimize the prediction error. The RAN algorithm uses a Least Mean Square (LMS) algorithm to update the unit parameters.

When the observation (X_n, y_n) does not satisfy the novelty criteria, the LMS algorithm is implemented to adapt the network parameter $W = \left[\alpha_0, \alpha_{1,}\mu_1^T, ..., \alpha_J, u_J^T,\right]^T$, and is updated for the n-th time frame as follows:

$$W^{(n)} = W^{(n-1)} + \eta e_n a_n \tag{2.5}$$

where η is the adaptation step size and $a_n = \nabla_w f(X_n)$ is the gradient of the function f(x) with respect to the parameter vector W evaluated at $W^{(n-1)}$ which is given by the following expression:

$$a_n = [1, \phi_1(X_n), \phi_1(X_n)\tfrac{2\alpha_1}{\sigma_1^2}(X_n - \mu_1)^T, \ , ..., \phi_J(X_n), \\ \phi_J(X_n)\tfrac{2\alpha_J}{\sigma_J^2}(X_n - \mu_J)^T]^T \tag{2.6}$$

2.2.2 Extended Resource-Allocating Network

The updating of the LMS algorithm in RAN is not accurate under anomaly status since it simply updates every incoming traffic. Moreover, LMS algorithm converges slowly [33]. In the following we propose the Extended RAN network (ERAN) in order to avoid erroneous updates during the anomaly status. We design an updating window of size $(2 * T_{Pers} + 1)$, where T_{Pers} is the time required for anomaly detection. Only the corresponding parameters during "normal" network status will be updated within the updating window. It should be noted here that undetected anomalous traffic may still be included, while detected anomalies are excluded from the updating process. This is achieved by choosing the median value of the corresponding parameters every time that the window is updated. Thus, this method can remove anomaly effects up to T_{Pers} time frame, including the single pulse noise anomaly. Continuous anomalies larger than T_{Pers} time frame, will be detected and the window will not be updated during the anomaly status.

In the following we introduce the Median Extended Kalman Filter (MEKF) Algorithm for the ERAN updating, where the Kalman Filter algorithm converges faster than LMS [34]. The MEKF Algorithm is used to update the ERAN parameters in order to minimize the prediction errors when no new hidden unit is being introduced. Here we define: $W = \left[\alpha_0, \alpha_{1,}\mu_1^T, \sigma_1, ..., \alpha_J, u_J^T, \sigma_J\right]^T$, and is updated for the n-th time frame as follows:

$$W^{(n)} = W^{(n-1)} + Med\left\{\acute{k}_1\acute{e}_1, ..., \acute{k}_{2T_{Pers}+1}\acute{e}_{2T_{Pers}+1}\right\} \tag{2.7}$$

The 'median' function Med is applied element by element while we choose the median value from the update vectors: $\acute{k}_l\acute{e}_l, 1 \leq l \leq (2 * T_{Pers} + 1)$, where $\acute{k}_l$ is the kalman gain vector and $\acute{e}_l$ denotes the error scalar. The median function is very important for accurate prediction because it tends to remove both the single occurrences of large spikes of noise and the continuous anomaly traffic during prediction. In relation (2.7), $\acute{k}_{2T_{Pers}+1}$ is the latest updated kalman gain vector, and it equals to $\acute{k}_n$ at time frame n. The kalman gain vector $\acute{k}_n$ at time frame n is given by:

$$\acute{k}_n = P_{n-1}a_n[R_n + a_n^T P_{n-1}a_n]^{-1} \tag{2.8}$$

where $a_n = \nabla_w f(X_n)$ is the gradient vector given by the following expression:

$$\begin{aligned} a_n = [1, \phi_1(X_n), \phi_1(X_n)\tfrac{2\alpha_1}{\sigma_1^2}(X_n - \mu_1)^T \\ , \phi_1(X_n)\tfrac{2\alpha_1}{\sigma_1^3} \|X_n - \mu_1\|^2 \ , ..., \phi_J(X_n), \\ \phi_J(X_n)\tfrac{2\alpha_J}{\sigma_J^2}(X_n - \mu_J)^T, \phi_J(X_n)\tfrac{2\alpha_J}{\sigma_J^3} \|X_n - \mu_J\|^2]^T \end{aligned} \tag{2.9}$$

where R_n is the variance of the measurement noise. The error covariance matrix is updated as follows:

$$P_n = [I - \acute{k}_n a_n^T]P_{n-1} + Q_0 I \tag{2.10}$$

Here parameter Q_0 is a scalar that determines the allowed random step in the direction of the gradient vector. The error covariance matrix P_n is a P X P positive definite symmetric matrix, where P is the number of parameters being adapted. Whenever a new hidden unit is added the dimension of P increases and hence the new rows and columns must be initialized. Here we choose the error covariance matrix $P_n = \begin{pmatrix} P_{n-1} & 0 \\ 0 & P_0 I \end{pmatrix}$ where P_0 is an estimate of the uncertainty in the initial values assigned to the parameters, which in this algorithm is also the variance of the observations X_n and y_n . The dimension of identity matrix I is equal to the number of new parameters introduced by adding a new hidden unit.

The algorithm begins with $\epsilon_n = \epsilon_{\max}$, the largest scale of the size of the entire input space of nonzero probability density. The distance e_n decreases exponentially since $\epsilon_n = \max\{\epsilon_{\max}\gamma^n, \epsilon_{\min}\}$, where $0 < \gamma < 1$ is a constant [19]. The value for ϵ_n decreases until it reaches $\epsilon_{\min}$. At the beginning, the system creates a coarse representation of the function, which is refined gradually by allocating units with smaller and smaller widths. Finally, when the system has learned the entire function to the desired accuracy and length scale, it stops allocating new units.

2.2.3 Algorithm Stabilization

In order to avoid the anomaly traffic and the noise effect on adding a new unit we use a root mean square (RMS) [34] prediction error sliding window method to store the "normal" traffic prediction error with sliding window size W_{pe} such that: $0 \leq W_{pe} \leq (2{*}\theta{*}T_{Pers}{+}1)$. Here θ is an integer larger than 1 in order to minimize the effect of the undetected anomaly traffic. The RMS value of the network output error at n-th observation is given by

$$\breve{e}_{nRMS} = \sqrt{\frac{\sum_{h=1}^{W_{pe}} e_h^2}{W_{pe}}} \tag{2.11}$$

where $\breve{e}_h$ is the prediction error in the RMS sliding window. Therefore, in addition to the two criteria presented in relation (2.3), we need a third criterion which must be satisfied before adding a new hidden unit as follows:

$$\breve{e}_{nRMS} > e'_{\min} \tag{2.12}$$

Moreover, we should remove those units which have very small contribution to the output for a long period of time. Therefore the output of each hidden unit is:

$$O_j = \alpha_j \phi_j(X) = \alpha_j \exp\left\{-\frac{1}{\sigma_j^2}\left\|X - u_j\right\|^2\right\} \tag{2.13}$$

We can also define the normalized ratio $r_j = \left\| \frac{O_j}{O_{\max}} \right\|$. Here $O_{\max}$ represents the largest output value of the hidden unit. If r_j is lower than some prespecified threshold η for a prespecified number of successive inputs, we will remove the hidden unit from the ERAN network. In this way we can limit the size of the network and increase the efficiency of the algorithm.

2.3 Detection of Anomalies Based on ERAN

Adaptive anomaly detection can be summarized as composing of three sets of methods and algorithms. Firstly, performance related data of the network are measured. Secondly temporal based performance thresholds for specific network traffic parameters that are identified as important and are monitored, are built for baselining performance characteristics, based on the traffic prediction algorithm described in the previous section. Thirdly anomaly detection procedure is executed by comparing the measured data and the baselines to detect anomalies and diagnose the occurrence of anomalies. In order to demonstrate how the anomaly-tolerant non-stationary traffic prediction algorithm developed in the previous section can facilitate the detection of anomalies in the network, in this section we introduce and design dynamic thresholds, and based on them we define adaptive anomaly violation conditions as a combined function of both magnitude and duration of the traffic deviations.

2.3.1 Definition of Dynamic Thresholds

During our prediction procedure, instead of seeking a single prediction value which is predicted according to the ERAN network, we can attempt to find the specific interval of the predicted values, that is highly likely to contain the actual value. In particular we begin by specifying some high probability, say $1-\beta$, and then we pose the following problem: Find an interval $[L_{th}^n, U_{th}^n]$ such that $P[L_{th}^n \leq y_n \leq U_{th}^n] = 1 - \beta$ where y_n is the real-time traffic at time frame n. Then we define the dynamic lower and upper thresholds respectively as L_{th}^n and U_{th}^n. For accurate prediction, we define the new interval by using the probability of the prediction errors. First, we define the threshold sliding window. This window will store the prediction errors under "normal" network status. The size of this sliding window can be determined according to the experiment. Let us assume that the minimum and maximum prediction error in the threshold sliding window is e_{min} and e_{max} respectively. Although we do not have apriori knowledge of the distribution of the random variables in the threshold sliding window, we can determine the probability density function (pdf) according to the random variables in the sliding window. In the following let us assume that the pdf of the random variable in the threshold sliding window is $g(s)$. Then the interval (S_1, S_2) which gives confidence that the prediction errors will have probability $1-\beta$ can be calculated by the following expression:

$$\int_{S_1}^{S_2} g(s)ds = 1-\beta \tag{2.14}$$

equation 2.14 can also be written as

$$P[S_1 \leq \frac{y_n - f(X_n)}{y_n} \leq S_2] = P[\frac{f(X_n)}{1-S_1} \leq y_n \leq \frac{f(X_n)}{1-S_2}] = 1-\beta \tag{2.15}$$

Here $f(X_n)$ is the predicted traffic at time frame n. For instance in figure 2.2 we provide an example how to define the interval according to the pdf of prediction errors. It should be noted that the area under the pdf curve, for the interval $(-S, S)$, occupies $1-\beta$ of the total area, and based on that we can easily obtain the desired value S. Therefore the dynamic lower and upper thresholds can be defined respectively as follows:

$$L_{th}^n = f(X_n)/(1-S_1); \;\; U_{th}^n = f(X_n)/(1-S_2) \tag{2.16}$$

Here $S_1 = max(-S, e_{min}); S_2 = min(S, e_{max})$. Please note that the thresholding sliding window is being updated as time involves.

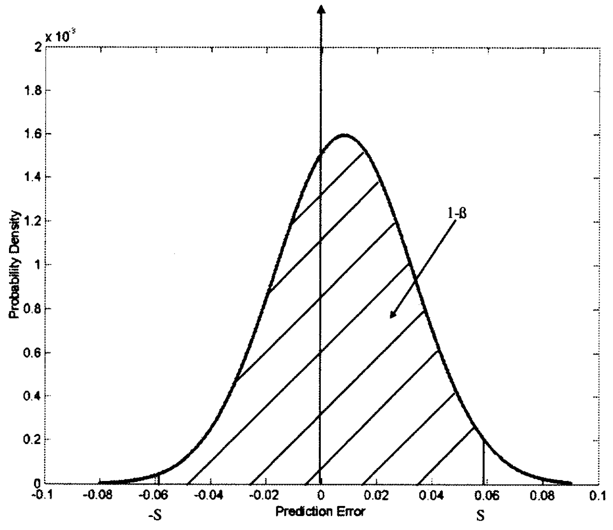

Figure 2.2 Probability calculation based on the pdf of prediction errors

2.3.2 Anomaly Violation Conditions and Detection

Since network anomalies may present different characteristics that vary both in amplitude as well as time duration, in the following we define the anomaly detection based on the above thresholds, as a combined function of both magnitude and duration. The reason of utilizing such a combination of both magnitude and duration, is to highlight the occurrence of anomalies that present a potential degradation in the network performance and behavior, while minimize the flagging of temporary anomalies that are "non-threatening" for the network's performance. We also define an anomaly violation sliding window with size T ($0 \leq T \leq T_{pers}$), which will record the anomaly status if the traffic crosses the dynamic threshold. Thus we define the anomaly violation condition as follows:

$$\left|\sum_{t=0}^{T} E(t) * t\right| \geq \sum_{t=0}^{T} k\sigma(t) f(X) * t \tag{2.17}$$

$$E(t) = \begin{cases} y_n - U_{th}^n & if \quad y_n \geqslant U_{th}^n \\ L_{th}^n - y_n & if \quad y_n \leqslant L_{th}^n \\ 0 & otherwise \end{cases} \tag{2.18}$$

where $\sigma(t) = \sqrt{var[Z(t)]}$, $Z(t)$ is the data in the prediction error window at time t and k is a constant that may be determined experimentally. When the traffic crosses the dynamic threshold, the violation magnitude and the corresponding time will be recorded and an anomaly detection alarm will be sent out if relation (2.17) is satisfied. This indicates that the network is under "anomaly" status and all time frames within the anomaly violation sliding window that are judged as being in anomaly status will be marked. Then the corresponding prediction errors marked with anomaly status will be removed from the threshold sliding window and RMS prediction error sliding window, the hidden unit added during anomaly status will also be removed from the hidden layer, while the anomaly violation sliding window will be reset to "empty".

2.4 Performance Evaluation Metrics

In order to evaluate and characterize the prediction accuracy of the ERAN methodology we utilize the Mean Absolute Percentage Error (MAPE) as a performance metric, which is defined as follows:

$$MAPE = \frac{\sum_L \frac{|l^n - l^{n^*}|}{l^n} X100\%}{L} \tag{2.19}$$

Here l^n and l^{n^*} is the actua! and predicted network traffic load at time instance n, and L represents the length of the evaluation window.

Furthermore three additional performance metrics of interest are utilized in order to better evaluate the effectiveness of the overall anomaly detection algorithm: detection probability P_b, false alarm probability P_f and miss probability P_m. Here the detection probability is defined as the probability that the abnormal traffic is being detected and recognized. The false alarm probability is defined as the probability that the normal traffic events are being classified as anomalies, while the miss probability is defined as the probability that an anomaly traffic is failed to being recognized. A successful anomaly detection algorithm should achieve high P_b, low P_f and low P_m. Since $P_b + P_m$=1, we usually evaluate the detection algorithm performance by the detection probability and the false alarm probability. In this dissertation, we use the Receiver Operating Characteristic to express the relationship of these two probabilities. Receiver Operating Characteristic (ROC) curves are a way of visualizing the trade-offs between detection and false alarm probabilities.

2.5 Performance Evaluation for Chaotic Non-stationary Time Series in the Presence of Anomaly Traffic

In this section we evaluate the proposed algorithm's operational efficiency and the corresponding anomaly detection capabilities in the presence of network anomaly traffic. The real traffic considered is based on a non-stationary chaotic time series: $X(n+1) = (1-b)X(n) + a\frac{X(n-\tau)}{1+X(n-\tau)^{10}}$. The constant parameters in the algorithm are chosen as follows: a = 0.2, b = 0.1, $\tau = 17, \epsilon_{\max}$ =1, $\epsilon_{\min}$ =0.01, $\varphi = 0.84, \gamma = 0.999$, $e_{\min}$ =0.02, $e'_{\min}$ =0.1, M=60, η =0.001, $T_{Pers} = 4, \beta = 0.03,\ Q_0 = 0.0002$.

In figure 2.3, we present the traffic index (vertical axis) that shows the traffic magnitude, versus the index that presents the prediction data counting number as time evolves (horizontal axis). The continuous line corresponds to the predicted traffic based on our proposed strategy, while the dotted line depicts the actual traffic. From this figure we can clearly see that the ERAN algorithm predicts the traffic proactively and accurately, when there are no network anomalies. For the case under consideration the corresponding MAPE is 3.245% between time index 4500 to 5000.

In figure 2.4, we present the hidden unit number (vertical axis) that shows the number of hidden units required in the original RAN algorithm, and in the ERAN network respectively, versus the index that presents the counting number as time evolves (horizontal axis). Comparing the hidden unit number in figure 2.4, we can see that not only the number of hidden units in the ERAN network is considerably lower, but the network converges to this value very

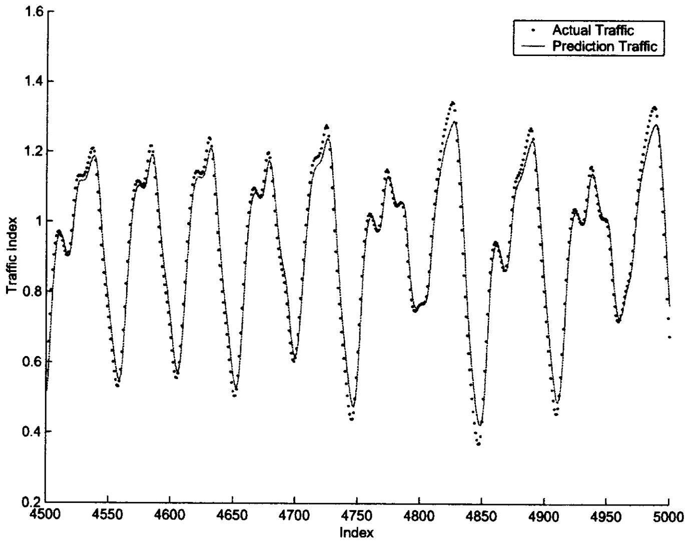

Figure 2.3 Predicted traffic vs. real traffic under normal status.

fast. These features make the ERAN network a very efficient methodology for detecting real-time anomalies.

In the following, we demonstrate the operation of ERAN algorithm under two different traffic anomaly scenarios. In the first one (scenario 1) we tested the algorithm's operation under the occurrence of single pulse anomaly traffic, while in the second one (scenario 2) we injected some continuous traffic anomalies in the system.

For scenario 1 we set the anomaly pulse traffic $APT = |X + 0.1| * Mod(n/80)$ where X follows the normal distribution with parameters (0,0.35) and n is the prediction data counting number. We can see from figure 2.5, that despite of the large pulse anomaly of the real traffic, our prediction algorithm removes the pulse anomaly effect on the traffic prediction, which confirms our claim that our proposed methodology is anomaly-tolerant. In figure 2.6 we present the actual traffic as it evolves, as well as the dynamic upper and lower thresholds as they are defined by our strategy. As can be seen by this figure the algorithm can effectively detect the anomaly traffic that crosses the dynamic upper or lower thresholds.

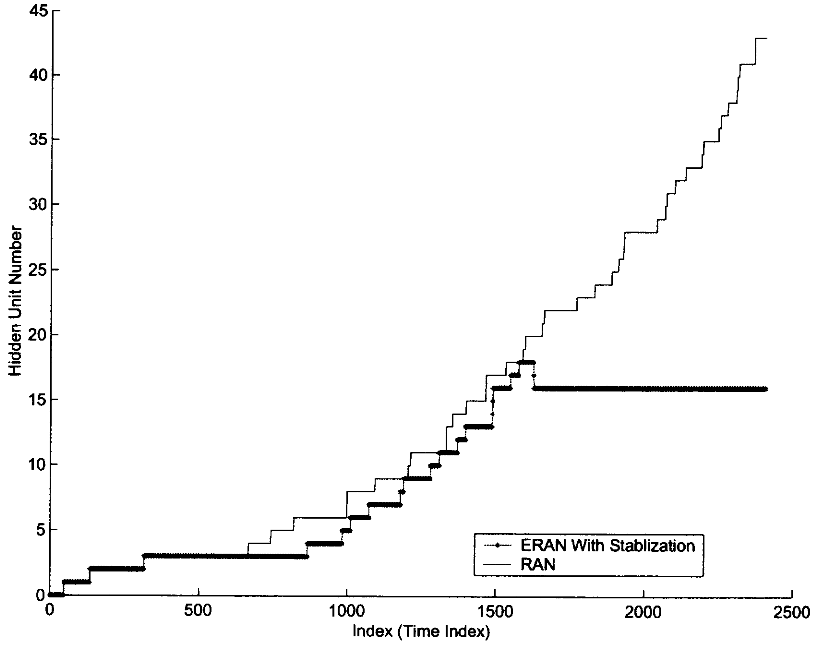

Figure 2.4 Hidden unit number in RAN and stabilized ERAN algorithm.

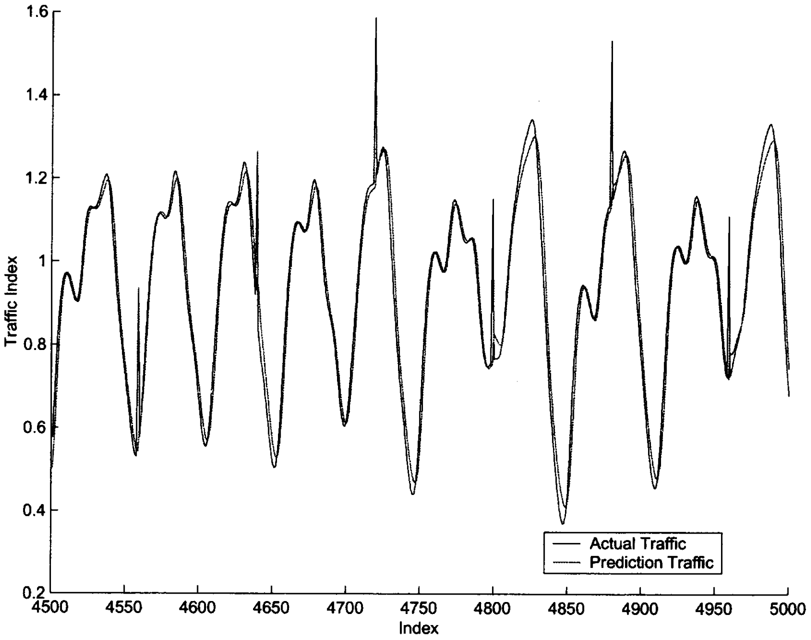

Figure 2.5 Anomaly-tolerant prediction under pulse anomaly.

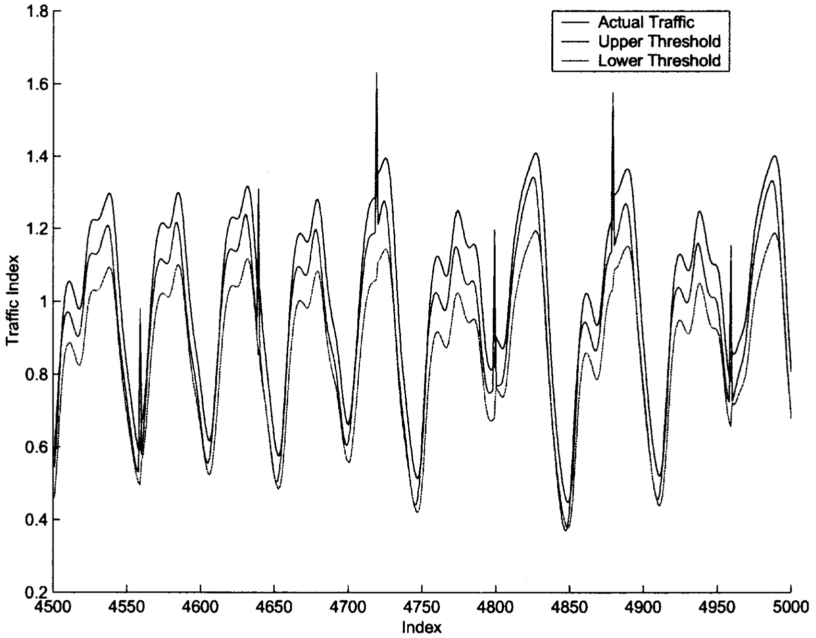

Figure 2.6 Anomaly detection based on ERAN algorithm for scenario 1

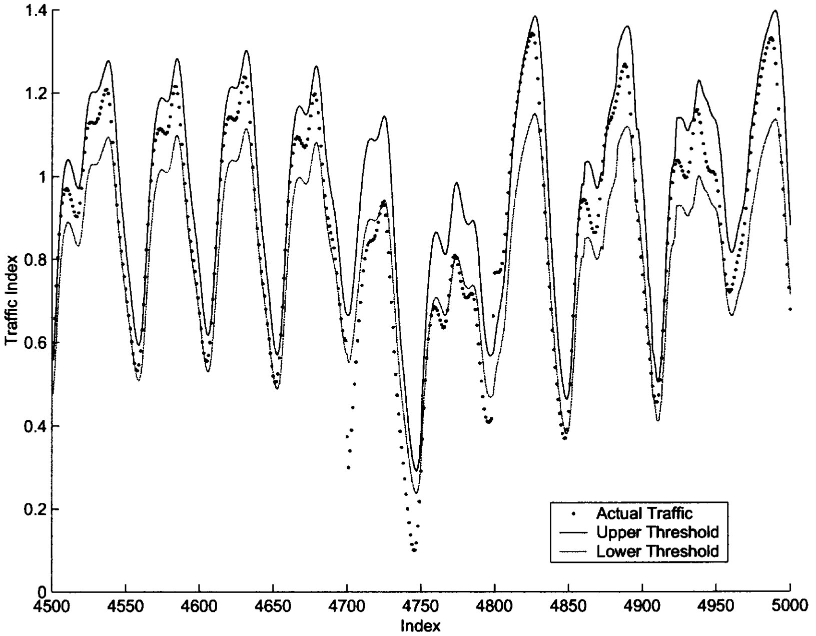

Figure 2.7 Anomaly detection based on ERAN algorithm for scenario 2

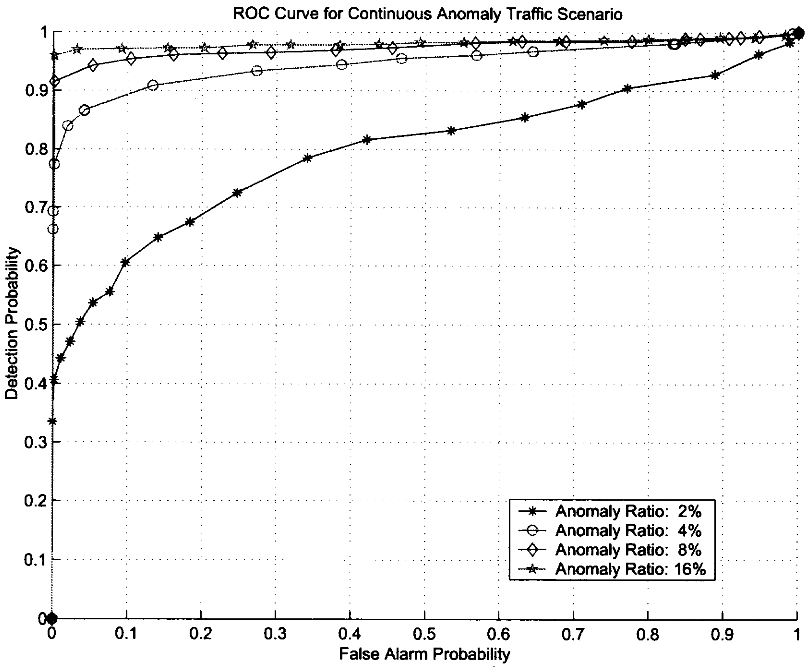

Figure 2.8 ROC Curve for continuous anomaly traffic scenario.

For scenario 2 we added some continuous anomaly traffic with normal distribution with parameters (0,0.4), from data set (indices) 4700 to 4850. We can see from figure 2.7 that in this case anomalous traffic crosses the dynamic thresholds for long period of time, while the algorithm can detect the anomaly timely and accurately. Moreover when the traffic is back to normal (after index 4850) the actual traffic is included very well within the predicted dynamic thresholds, due to the anomaly-tolerant feature of the traffic prediction technique we developed.

Figure 2.8 displays the plot of the detection probability vs. false alarm probability using the Receiver Operation Characteristic curves under different anomaly ratios. Here the anomaly ratio is the ratio between the mean anomaly traffic and the background traffic. The x-axis of this figure presents the false alarm probability (which is the probability of the typical traffic events being classified as anomalies), while the y-axis of the figure depicts the detection probability which is calculated as the ratio between the number of correctly detected faults/anomalies to their total number. For each curve, the point at the upper left corner represents the optimal detection, with high detection probability and low false alarm probability. From figure 2.8, we can observe that the detection performance improves as the anomaly ratio increases. Also, we should note that there is a close relationship between the detection probability and the dynamic thresholds which are based on the prediction errors. If the anomaly ratio is very small, for example, 2%, we can see from ROC curves in Figure 2.8 that the detection probability is low as well. In this case the anomaly ratio is below the MAPE which is 3.245%. However the detection probability greatly improves once the anomaly ratio becomes larger that the MAPE.

2.6 Performance Evaluation under Network Attacks/Intrusions

In order to validate our statistical models and evaluate the operational effectiveness and performance of our methodology, we carried out an extensive set of experiments of the network performance evolution in the presence of intrusion traffic, and studied how the proposed methodology achieved accurately and timely to detect the presence of attacks. This evaluation was achieved via modeling and simulation using the Optimized Network Engineering Tool (OPNET). In this section we present the performance evaluation process and some numerical results under typical background traffic and two different types of attacks and anomaly traffic characteristics. The attacks that were modeled and simulated in these experiments include the Internet ”mail-bombing” attack and the UDP flooding attack. Specifically in section 2.6.1 we present the network model architecture that was utilized and describe the background network traffic characteristics as well as the details of the attack models that were used throughout these experiments, while section 2.6.2 contains the corresponding numerical results along with relevant discussions and observations.

2.6.1 Model and Assumptions

Network Architecture

The corresponding network architecture that was used throughout this evaluation is shown in Figure 2.9. It consists of three subnetworks connected by 3 routers. The various clients are located in all the three subnetworks, i.e. Subnet_1 (Ethernet Network - Figure 2.10), Subnet_2 (Fast Ethernet Network) and Subnet_Server (FDDI Network - Figure 2.11). Each of the three subnetworks supports several clients that can establish communication with the Main Server located in Subnet_Server, which can support several applications such as http, ftp and e-mail applications.

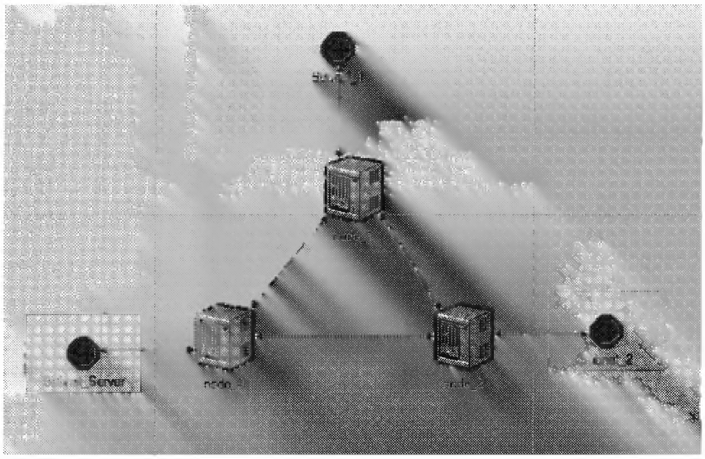

Figure 2.9 Network model

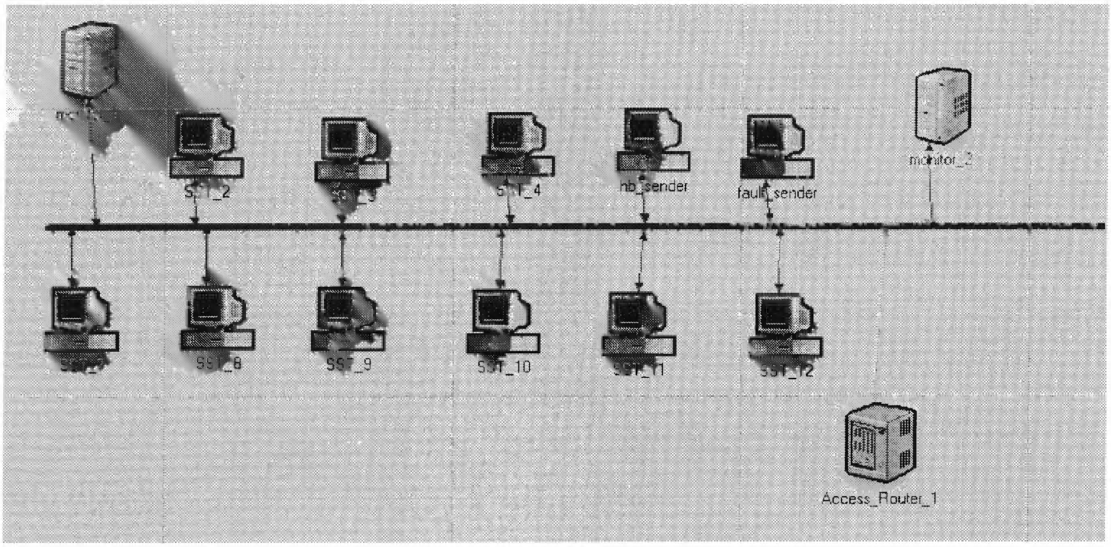

Figure 2.10 Diagram for subnet_1

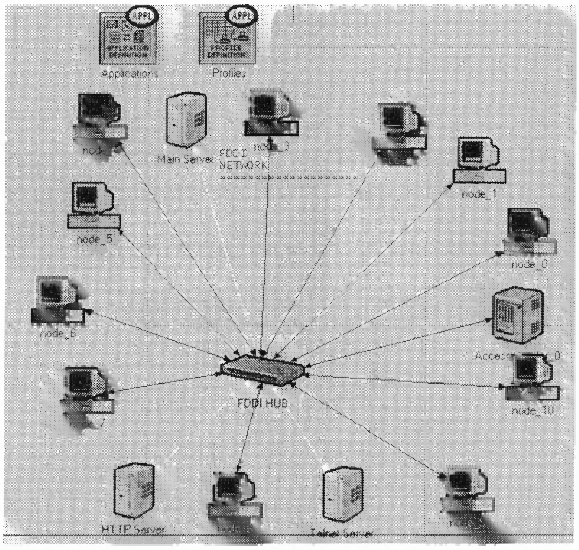

Figure 2.11 Diagram for subnet_server

Background Traffic Configuration

In our experiments the background traffic is modeled as a mix of http, ftp and e-mail applications. The background traffic is collected at the MAC layer of the Main Server.

In order to model and simulate an http conversation, we modeled the following characteristics: Number of bytes per object (for client to server and for server to client), active off time, inactive off time, and number of objects per page. Specifically the number of bytes per object from client to server accounts for the size of an http request message in the client to server direction, while the number of bytes per object from server to client accounts for the size of each object (e.g., an image or a main body in HTML format) transferred from the server to the client. The active off time refers to the time between the transfers of the components of a single web page, while the inactive off time corresponds to the time between the transfer of two web pages; this is the user ”thinking time”. The number of objects per page corresponds to the number of objects embedded in the web page plus the main body of the page. In our simulation, we assume that HTTP1.1 is used, which means that all the objects, including the main body of the web page, will be transferred in a single connection.

An ftp communication is modeled and simulated as follows. First the client establishes an ftp control connection to the server. After that, a series of Put/Get commands are sent to the server through this connection. The inter-arrival time between two successive commands is distributed according to Pareto distribution. On receiving an ftp command from the client, the server will respond with a control packet. Following that, a separate ftp data connection will be established for the file transfer. The size of the file is also distributed according to Pareto distribution

Finally, an e-mail communication is simulated as follows. First, the e-mail client makes a request to send e-mail data to the e-mail server; then the server responds according to the request of the client. No connection is established between client-server. The inter-arrival time between Send (Receive) commands is distributed according to lognormal distribution, while Send (Receive) group size follows a normal distribution. On receiving an UDP request from the client, the server responds with a control packet. Following that, a separate UDP data port will be opened for the data to be transmitted. The size of the data portion of the e-mail is distributed according to pareto distribution.

In this dissertation we used 24 clients (8 clients in each subnet) to simulate the daily Internet activities. These clients are connected to the network periodically to simulate the real network traffic. Figure 3.5 presents the corresponding simulated data for four weeks of network background traffic, where the vertical axis represents the traffic intensity in kbits/sec while the horizontal axis represents the time evolution (in hours). For presentation purposes, the time resolution in this simulation is set to 5 minutes. It should be noted that the data used here are for demonstration purposes only, in order to mainly represent the trend and the behavior of the expected Internet traffic, while the actual values could be different and usually depend on several parameters including the time resolution, the number of users, the traffic density, the applications etc.

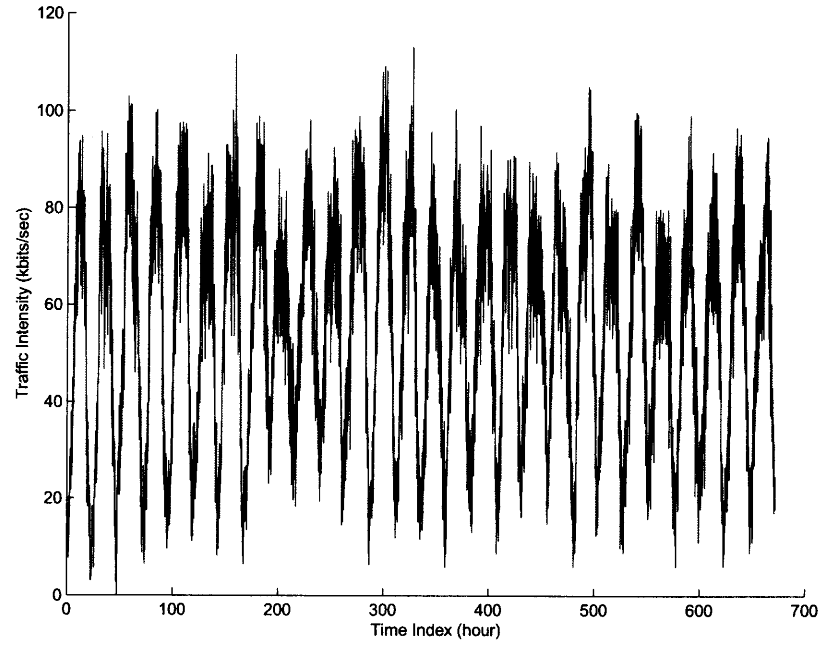

Figure 2.12 Background traffic

Attack/Anomaly Models

Two different attacks, namely the mail-bombing attack (scenario 1) and UDP flooding attack (scenario 2) were modeled and used throughout our evaluation. Internet ”mail-bombing” is the act of sending an extraordinarily large number of duplicate or random messages via Internet electronic mail to a target whose account typically resides on a system other than the one the messages were originated from. The characteristics of the anomalous injected traffic for the scenario 1 attack (”mail-bombing” attack) are as follows: the packet length of injected traffic follows exponential distribution with mean 1000 bits; The send and receive interarrival time follow pareto distribution with parameters of (30,1.8); The target group size follows exponential distribution with mean 200 users.

An UDP flooding attack repeatedly transmits a UDP packet to a specified port at the target machine for a specified period of time. The characteristics of the anomalous injected traffic for the scenario 2 attack (”UDP flooding” attack) are as follows: the repetition interval follows lognormal distribution with parameters (0.37,2.25) and the packet length is exponentially distributed with mean 1000 bits. In our experiments, a client (workstation) in the subnet_1 is assumed to generate and send the mail-bombing attack or UDP flooding attack packets to Subnet_Server.

2.6.2 Numerical Results and Discussion

In this section we provide some numerical results to evaluate the proposed algorithm's operational efficiency and the corresponding anomaly detection capabilities. The constant parameters of the algorithm are chosen as follows: $\epsilon_{\max}$ =1, $\epsilon_{\min}$ =0.01, $\varphi = 0.84, \gamma = 0.999$, $e_{\min}$=0.02, $e'_{\min}$ =0.1, η =0.001, $T_{Pers} = 4$, $\beta = 0.05$, Q_0=0.0004.

In figure 2.3, we present the traffic index (vertical axis) that shows the traffic intensity (kbits/sec), versus the time index (hours) as the system evolves. The curve with the asterisks corresponds to the predicted traffic based on our proposed strategy, while the solid line depicts the real traffic. From this figure we can clearly see that the ERAN algorithm predicts the traffic proactively and accurately, when there are no network anomalies. For the case under consideration the corresponding MAPE is 7.927% between time indices 500 to 548.

In order to demonstrate the operational efficiency of the proposed methodology, in figure 2.14, we present the hidden unit number (vertical axis) that shows the number of hidden units required in the original RAN algorithm and in the ERAN algorithm, versus the time index (horizontal axis). We can see that not only the number of hidden units in the ERAN network is considerably lower than the RAN algorithm, but the network converges to this value very fast. These features make the ERAN network a very efficient methodology for detecting real-time anomalies.

In the following, we demonstrate the operation of ERAN algorithm under the two different attack scenarios that we described above: scenario 1 ("mail-bombing attack) and scenario 2 ("UDP flooding attack"). For scenario 1 we injected anomaly "mail-bombing" traffic in the network at two different time instances, time index 515 and time index 530,

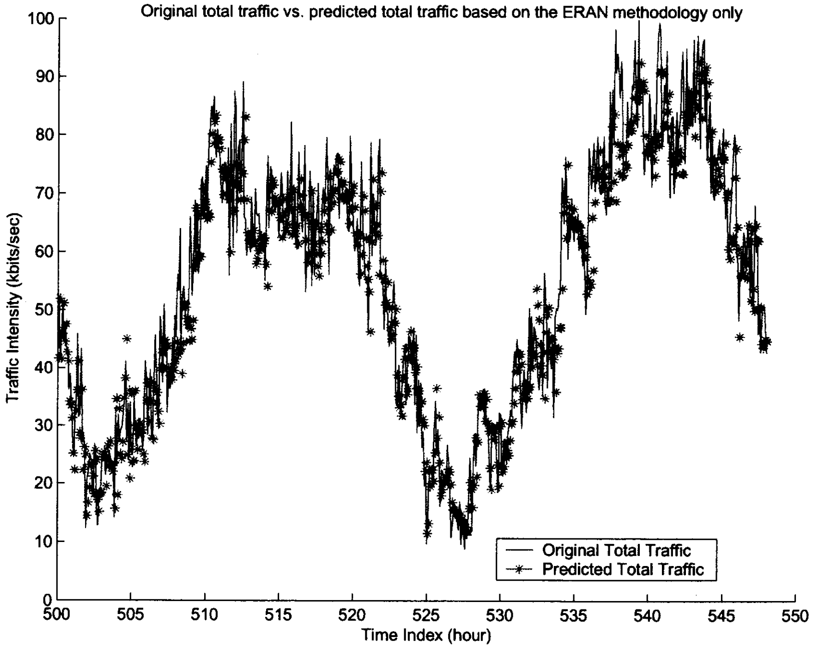

Figure 2.13 Predicted traffic vs. real traffic under normal status.

while the duration of each such occurrence was one hour. We can see from figure 2.15, that despite of the large pulse anomaly of the real traffic (represented by the solid line), our prediction algorithm removes the pulse anomaly effect on the traffic prediction (represented by the line with asterisks), which confirms our claim that our proposed methodology is anomaly-tolerant. In figure 2.16 we present the real traffic as it evolves, as well as the dynamic upper and lower thresholds as they are defined by our strategy. As can be seen by this figure the algorithm can effectively detect the anomaly traffic that crosses the dynamic upper thresholds (represented by the line with stars) or lower thresholds(represented by the line with squares). Therefore based on relations (2.17) and (2.18) we can configure the system to automatically detect the anomaly and alarm the corresponding condition, if required.

For scenario 2 we injected some UDP flooding attack traffic from time index 515 to time index 525. We can see from figure 2.17 that in this case the anomalous traffic crosses the dynamic upper thresholds(represented by the line with triangles) for a long

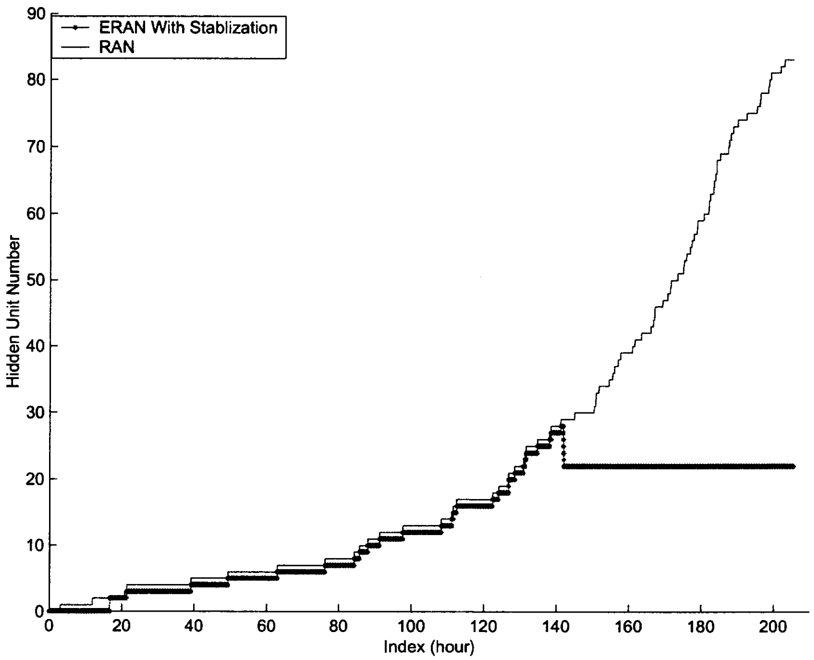

Figure 2.14 Hidden unit number in RAN and stabilized ERAN algorithm.

period of time, and the algorithm can again detect the anomaly timely and accurately, based on conditions (2.17) and (2.18). Moreover when the traffic is back to normal (after index 525) the actual traffic is again included very well within the predicted dynamic thresholds, due to the anomaly-tolerant feature of the traffic prediction technique we developed.

Figure 2.18 displays the plot of the detection probability vs. false alarm probability using Receiver Operation Characteristic curves. The x-axis depicts the false alarm probability while the y-axis presents the corresponding detection probability. For each curve, the point at the upper left corner represents the optimal detection, with high detection probability and low false alarm probability. Here the anomaly ratio is the ratio between the mean intrusion/anomaly traffic and background traffic. From this figure, we can observe that the detection performance improves as the anomaly ratio increases, as it was also demonstrated in the previous sections. We also observe here as well that there is a relationship between the detection probability and the dynamic thresholds which are based on the prediction errors. If the anomaly ratio is very small, for example, 4%. We can see from the

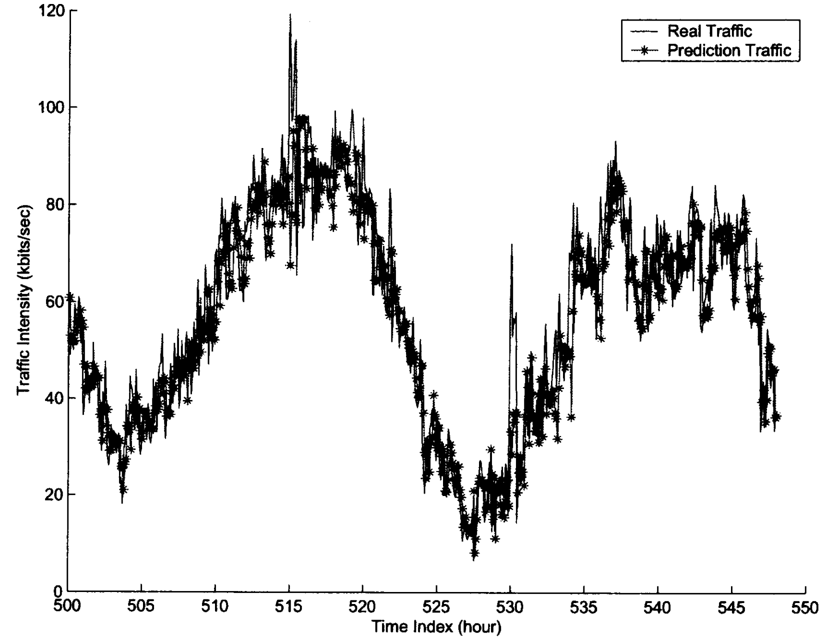

Figure 2.15 Anomaly-tolerant prediction under pulse anomaly (scenario 1)

ROC curves in Figure 2.18, that the detection probability is low as well. In this case the anomaly ratio is below the MAPE which is 7.927% for this scenario.

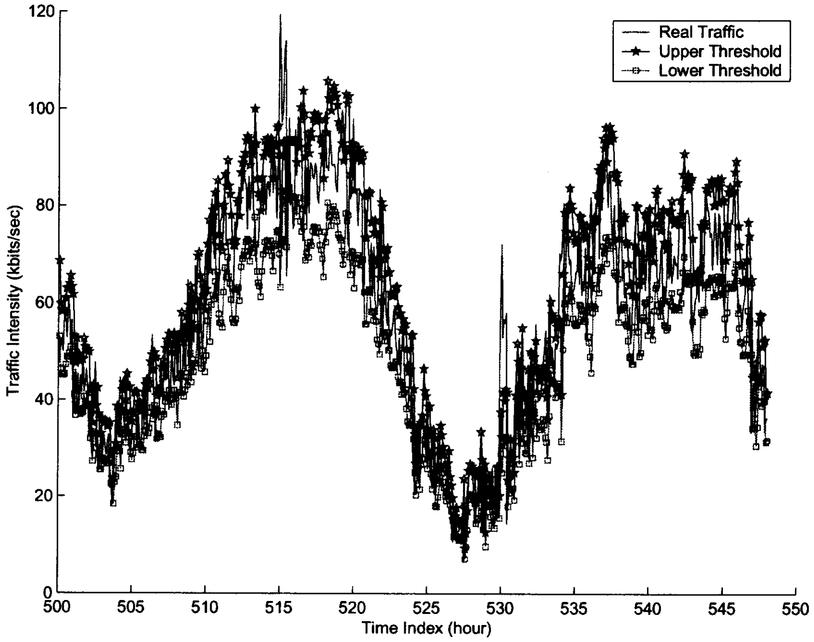

Figure 2.16 Anomaly detection based on ERAN algorithm for scenario 1

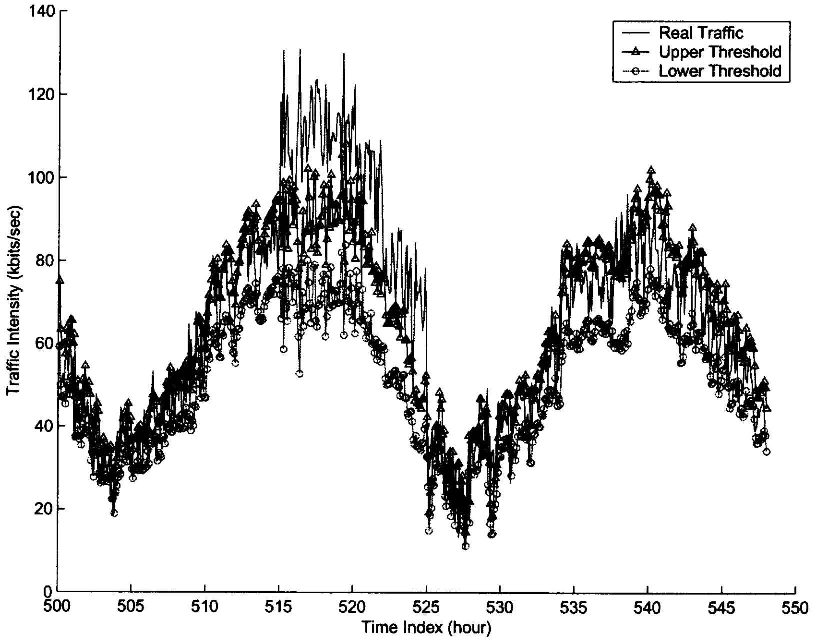

Figure 2.17 Anomaly detection based on ERAN algorithm for scenario 2

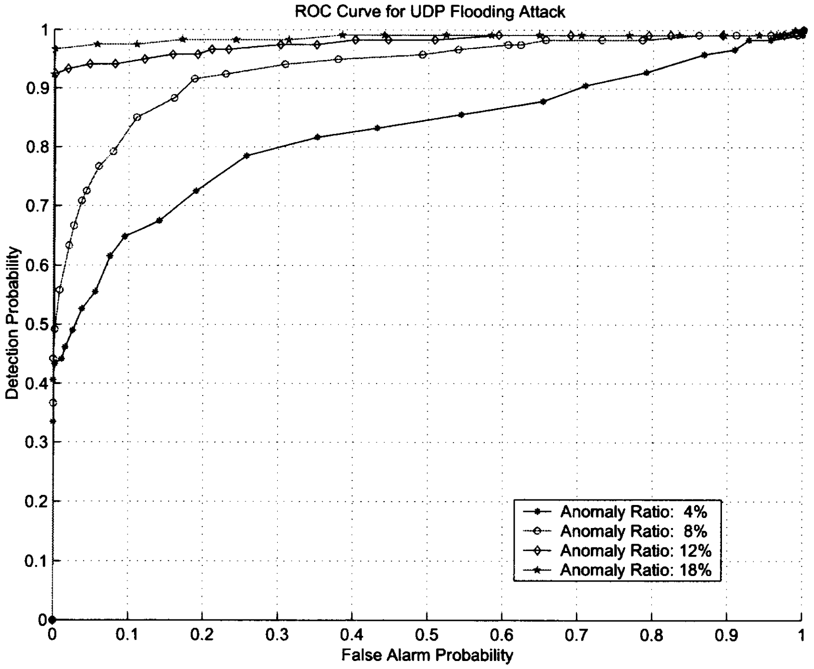

Figure 2.18 ROC curve for UDP flooding attack.

CHAPTER 3

ACCURATE TRAFFIC SEPARATION-COMBINATION AND PREDICTION STRATEGIES

3.1 Introduction

This chapter, emphasizes on the design and development of enhanced strategies that can be used to further improve the accuracy of the prediction of the statistical network traffic normality, and as a result of the overall anomaly detection methodology, especially in cases where high burstiness is present. In the previous chapter we presented the ERAN algorithm and we demonstrated that it can provide accurate prediction for non-stationary time series traffic, mainly in cases with low burstiness. In the high burstiness scenarios (such as the expected real traffic in Internet) the corresponding prediction errors may increase due to the very dynamic characteristics of the traffic. Furthermore it was demonstrated that the anomaly detection performance is directly correlated with the traffic prediction accuracy in the proposed anomaly detection algorithm.

Analyzing various Internet traffic samples, we can easily observe that there are some underlying trends regarding the corresponding traffic patterns (e.g. daily, weekly patterns, etc), that are mainly caused by different temporal and in some cases physical phenomena. These traffic components are mainly located in the low frequency area in the "frequency domain", while the dynamic part of the Internet traffic is mainly located in the high frequency. Therefore, in this chapter we first propose a new methodology that analyzes the traffic in the frequency domains. Specifically we separate the traffic into a baseline component that includes most of the low frequency traffic and presents low burstiness, and the short-term traffic that includes the most dynamic part. Since the baseline traffic is a mean non-stationary periodic time series, the methodology developed in the previous chapter will be used for its accurate prediction. The short-term traffic is shown to be a time-dependent series, and the Autoregressive Moving Average (ARMA) model is proposed to be used for the accurate prediction of this component.

One of the key principles of our proposed methodology is that most of the non-stationary traffic is separated and included into the baseline component, and therefore the short-term traffic, as shown here, can be well and accurately modeled using the ARMA model. Numerical results presented in this chapter demonstrate that the proposed methodology of separating the traffic based on the "frequency domain" analysis and predicting each component separately by the appropriate method, improves significantly the prediction accuracy of the total combined Internet traffic, which in turn improves the performance of the network anomaly detection as well.

3.2 Background Information

In order to gain some insight about the various characteristics of the Internet traffic, and present the observations that motivated our proposed strategies based on the traffic separation in the "frequency domain", we first provide some examples regarding typical Internet traffic patterns from two datasets available in the literature and from simulated traffic that

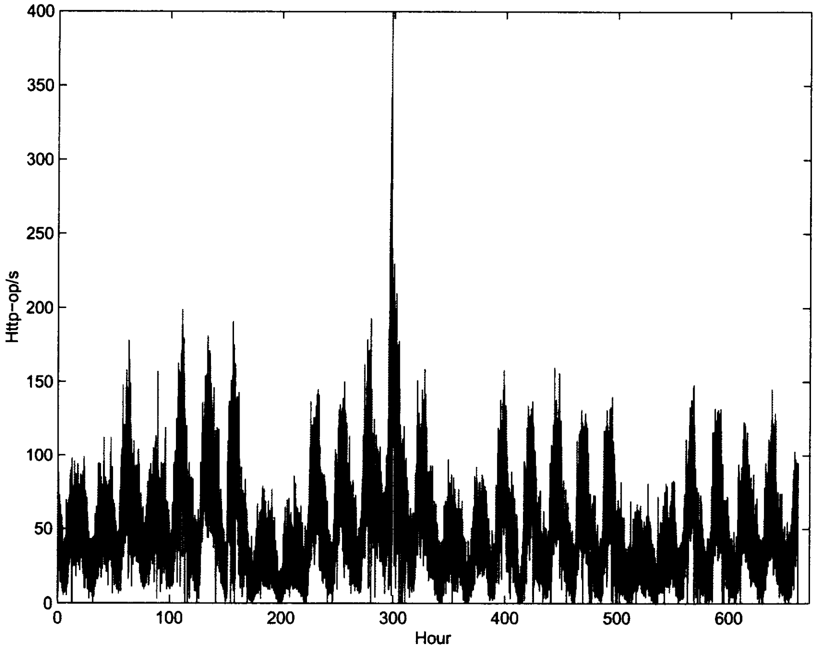

Figure 3.1 Traffic intensity on a HTTP server in dataset 1

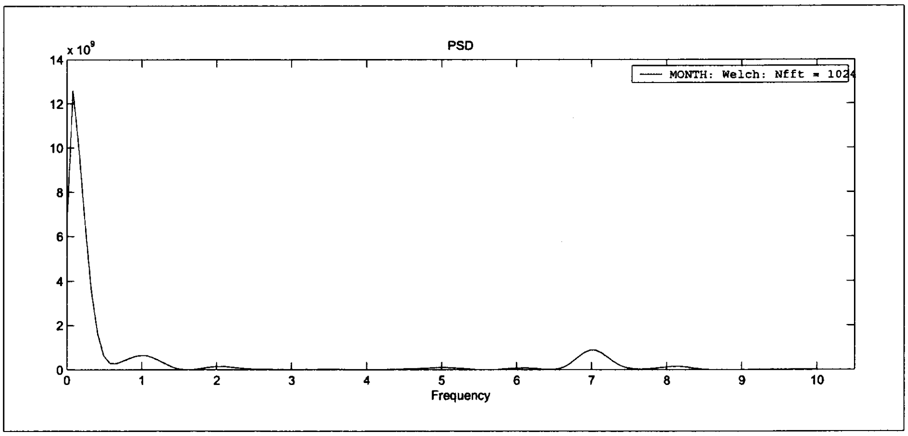

Figure 3.2 Power spectral density for dataset 1 (with frequency cycles/week)

was generated using the Optimized Network Engineering Tool (OPNET) Modeling and Simulation Tool. The two datasets we use in this section have been collected by The Internet Traffic Archive [35]. The variables that are chosen to represent the corresponding data are the number of HTTP operations per second in the two datasets, and the traffic intensity in Kbits per second for the simulated traffic.

Dataset 1 contains one-month's worth of all HTTP operations to the NASA Kennedy Space Center WWW server in Florida. In figure 3.1, we plot the corresponding HTTP operations' intensity. We can observe from this figure that there are some underlying trends with respect to the daily and weekly traffic patterns. For instance, in each day the traffic intensity starts to increase at the beginning of the business hours, while the HTTP intensities reach the peak and remain there for most of the business hours. In figure 3.2 we present the corresponding Power Spectral Density (PSD) of the traffic in the spectral domain (frequency unit is cycles/week). We can easily observe from this figure that the daily and weekly spectrums are very strong, while most of the energy is located in the low frequency area.

Dataset 2 contains 7-month's worth of data of all the HTTP operations to the University of Saskatchewan's (Canada) WWW server. Figure 3.3 presents the corresponding HTTP operations' intensity, while in figure 3.4 we present the corresponding PSD of the traffic

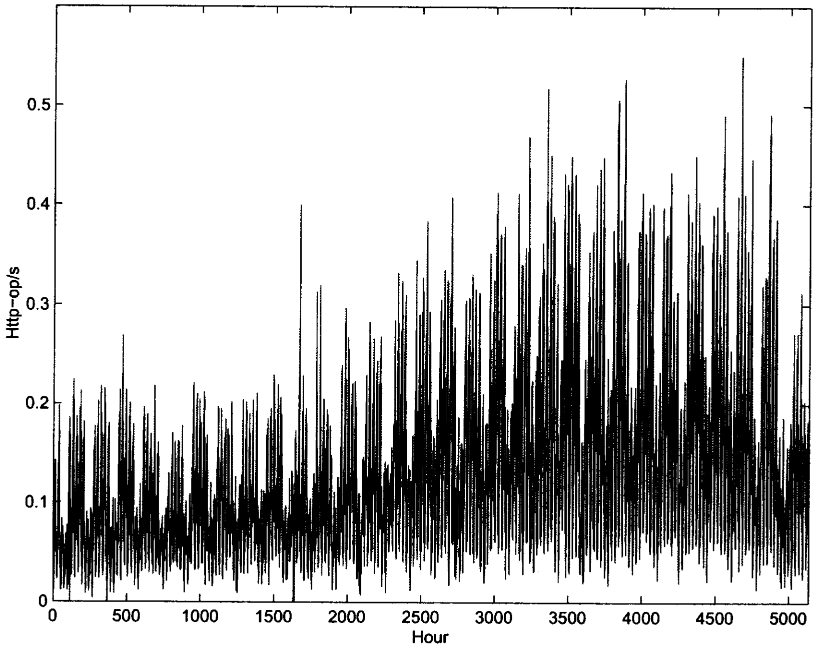

Figure 3.3 Traffic intensity on a HTTP server in dataset 2

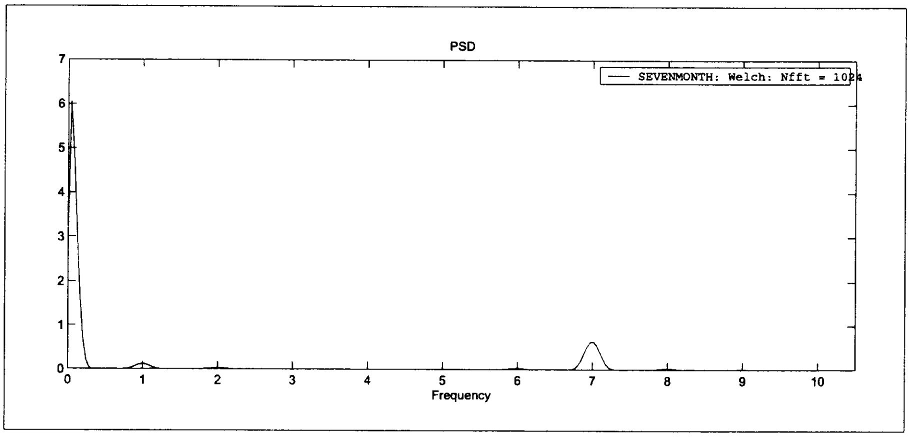

Figure 3.4 Power spectral density for dataset 2 (with frequency cycles/week)

in the spectral domain (frequency unit is cycles/week). We can observe from figure 3.4 that there is a strong presence of frequency components at the seven cycles per week (daily pattern), while the components at the weekly level (one cycle/week) are relatively weaker. However we do not find any identifiable monthly or seasonal patterns from the PSD of dataset 2 (Figure 3.4), though the corresponding dataset includes data for seven months.

In the following figure 3.5 we present the data for a 4-week OPNET simulated network traffic, while figure 3.6 depicts the corresponding PSD. It can be seen from the latter figure that there exists a strong daily pattern, while most of the energy concentrates within the low frequency area.

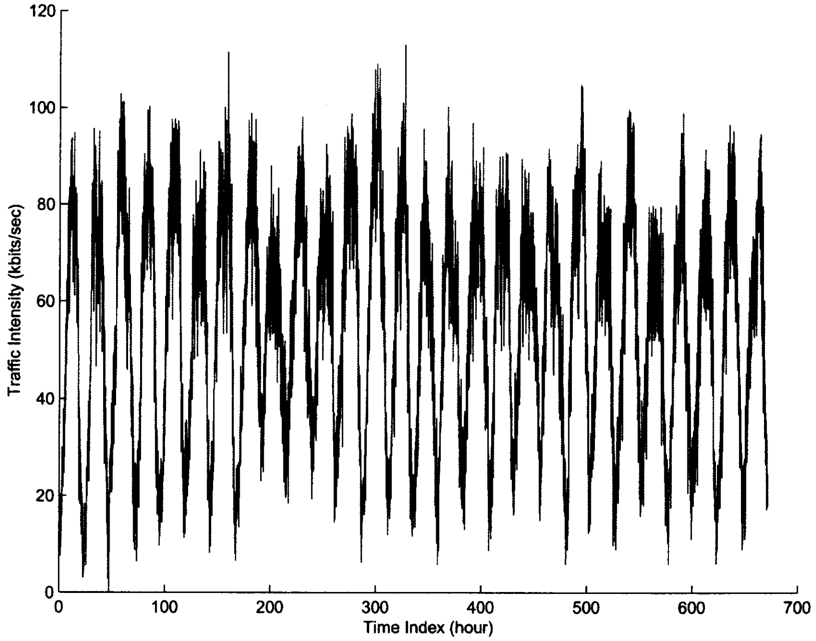

Figure 3.5 OPNET simulated traffic

Based on the above observations, we conclude that some strong patterns can be identified in the network traffic and most of the energy concentrates with the low frequency domain, while the bursty network traffic mainly resides in the high frequency. The traffic within different frequency ranges may represent different phenomena, and therefore in order to achieve high prediction accuracy it would be more efficient to apply appropriate

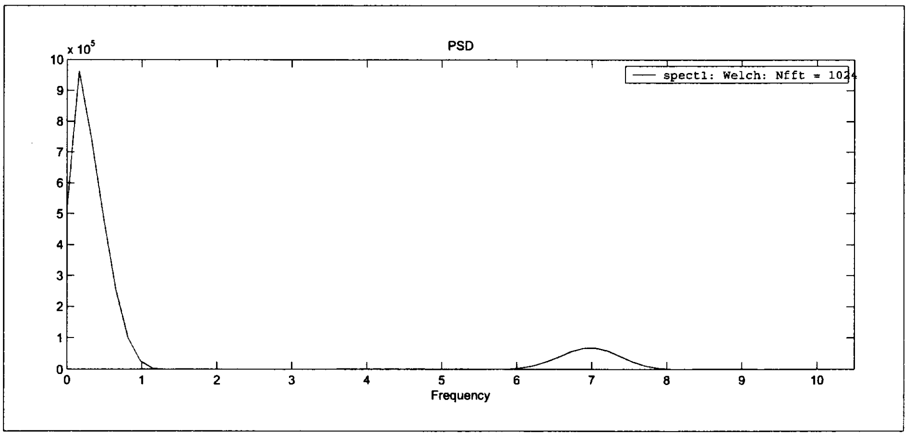

Figure 3.6 Power spectral density for simulated traffic (with frequency cycles/week)

strategies on the separated traffic components. In the next section we introduce a new methodology that separates the network traffic into a low frequency baseline traffic component, which is non-stationary periodic traffic that contains most of the energy, and the short-term traffic component that includes the high bursty traffic.

3.3 Proposed Traffic Separation Strategies

Techniques presented in the literature (e.g. [36, 37]) based on daily and weekly mean average separations and in the [38] based on given cut-off frequency separation, although can be applied in many cases, in general are not sufficient, because the energies of the corresponding separated components are mixed and attributed to each other [39]. Such traffic separation mechanisms may reduce the possibility of drawing accurate inferences and make accurate predictions about the expected traffic. Therefore, as mentioned before, motivated by the characteristics observed in the PSD figures presented above, in this chapter we separate the Internet traffic based on the spectral domain. The most general applied strategies for frequency separation are based on the use of the spectral filters. Among the several high-resolution filters available for frequency separation in spectral domain, we choose to apply here the Kolmogorov-Zurbenko (KZ) filter [40] to separate the high frequency and the low frequency traffic. The underlying principle that motivated the use of KZ filter is, that it is distinguished by its simple algorithm and the preservation of true information when applied to a nonequally spaced and/or missing data environment [39, 41].

The KZ(m,k) filter is defined as k applications of a simple moving average of m points. The KZ filter can be expressed as the following equations:

$$Y^l(t) = \frac{1}{m} \sum_{s=-(m-1)/2}^{(m-1)/2} Y^{l-1}(t+s) \qquad l = (1, ..., k) \tag{3.1}$$

$$Y^0(t) = \frac{1}{m} \sum_{s=-(m-1)/2}^{(m-1)/2} X(t+s) \tag{3.2}$$

where X is the original time series and t denotes the time. The time series produced by l iterations of the filter described by equation (3.1) is denoted by $Y^l(t)$. From the above equations we can see that the KZ filter is an iterated moving average filter, since the output from the simple moving average is again subjected to the same moving average operation for a specified number of iterations. The transfer function of the KZ filter can be expressed as follows [39]:

$$\left|\phi_{m,k}(\omega)\right|^2 = \left[\frac{1}{m}\frac{\sin(\pi m \omega)}{\sin(\pi \omega)}\right]^{2k} \tag{3.3}$$

Please notify that when m=1, the KZ filter becomes an all-pass filter and it is not able to separate the traffic. Therefore we define m≠1 in this dissertation. Moreover, according to equation 3.1 and 3.2, m can not be an even number because parameter s has to be an integer for time consistent. Based on equation (3.3) and the corresponding figure 3.7, we observe that the KZ filter is a low-pass filter [42]. Parameter k controls the level of noise suppression. For example, if a value of k is to be chosen such that the height of the additional peaks in the squared transfer function is less than 10^{-5}, the resulting value for k will be ≥ 4 (see figure 3.7). Once k is fixed, m is chosen such that:

$$\omega_0 \approx \frac{\sqrt{6}}{\pi}\sqrt{\frac{1-(1/2)^{1/2k}}{m^2-(1/2)^{1/2k}}} \tag{3.4}$$

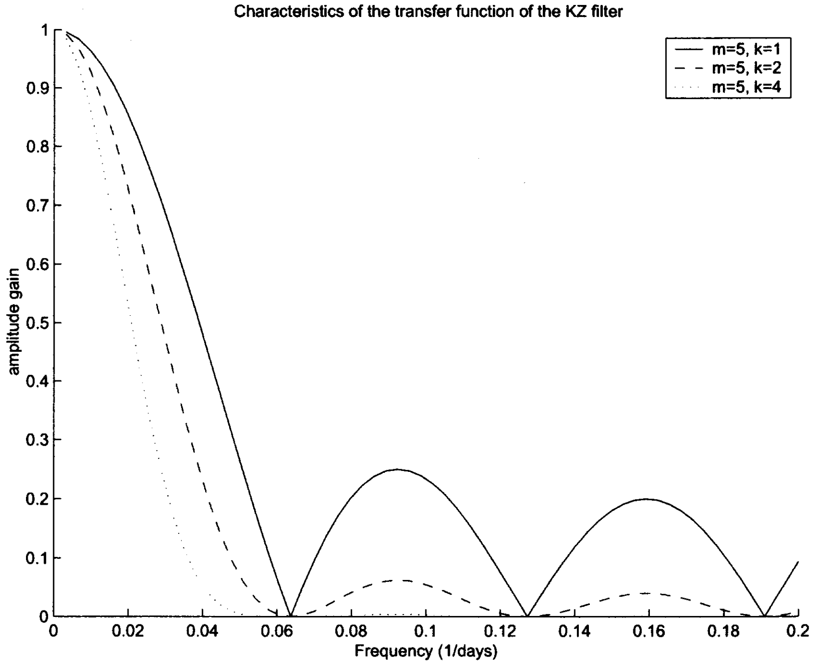

Figure 3.7 Characteristics of the transfer function of the KZ filter

where ω_0 is the desired separating frequency. This is the approximate solution for

$$|\phi_{m,k}(\omega)|^2 = 1/2 \qquad (3.5)$$

In order to analyze the Internet traffic based on the KZ filter, we first build a conceptual model for the Internet traffic. We define:

$$R(t) = L(t) + W(t) + D(t) + S(t) = Baseline(t) + S(t) \qquad (3.6)$$

where R(t) is the original traffic time series, L(t), W(t) and D(t) represent the long-term, weekly and daily traffic components respectively, while S(t) is the short-time variation. In the following we assume that the baseline traffic consists of (L(t)+W(t)+D(t)). Therefore each part of the traffic can be estimated as follows:

$$Baseline(t) = KZ(R(t)); \quad S(t) = R(t) - KZ(R(t)) \qquad (3.7)$$

where KZ( ) denotes the operation that the corresponding signal passes through the KZ filter. Please note that based on this separation the baseline traffic is a mean non-stationary periodic signal with low burstiness, and as a result the ERAN algorithm described in the chapter 3 provides an efficient and accurate prediction for this traffic.

In the remaining of this section we describe how to define the separating frequency and the KZ filter parameters m and k. The effective filter width of the KZ(m,k) filter is approximately m$\sqrt{k}$ [39], which results in an approximate separation frequency 0.5/(m$\sqrt{k}$). After we determine the separating frequency from the spectral domain and the level of noise suppression which is determined by parameter k, we can calculate the corresponding m for each separating frequency according to equation (3.4). Moreover, given the level of noise suppression, we can also find the most appropriate separating frequency by calculating the minimum covariance ratio with the corresponding parameter m.

The energy of the separated processes should be concentrated at different frequencies (spectral domain) and the information in the natural physical processes that cause these energies should be independent of each other. The filter's squared transfer function (gain) shows the transfer of energy to each component affected by a separation technique [39]. A good separation technique is characterized by a gain function that concentrates the energy at the timescale of interest and does not mix energies from different timescales. Although the squared transfer function (gain) provides information about the transfer of energy to the resulting components, it does not define whether these energies are mixed. More precisely, we want to obtain a measure of the correlation between the resulting components of the KZ filter. The traffic can be successfully separated only if the correlation between the separated components is small enough.

The covariance between the result of the KZ filtration and the residual [1- KZ], is:

$$R_{cov} = \int f\,(\omega)\,\phi_{m,k}(\omega)[1 - \phi_{m,k}\,(\omega)]d\omega \tag{3.8}$$

and therefore, the kernel for the covariance between KZ and [1-KZ] is given by:

$$K(\omega) = \phi_{m,k}(\omega)[1 - \phi_{m,k}(\omega)] \tag{3.9}$$

Usually there is high correlation between the filtered data and the residuals for small values of k; also, the width of the kernel gets smaller as m and k increase. Therefore we need to find appropriate values of m and k to guarantee that the separated traffic is not correlated and the power is mainly located in the kernel.

To verify the effectiveness of the separation, the covariance of the different parts of the traffic should be calculated and evaluated. The total variance of R(t) can be written as a sum of the variances and covariances between the corresponding traffic components. Therefore:

$$\sigma^2(R(t))=\sigma^2(Baseline(t) + S(t)) = \sigma^2(Baseline(t)) + \sigma^2(S(t)) + R_{cov} \tag{3.10}$$

The covariance is

$$R_{cov} = \sigma^2(R(t)) - \sigma^2(Baseline(t)) - \sigma^2(S(t)) \tag{3.11}$$

If the covariance R_{cov} is small compared to the total variance (e.g. less than 2% of the total variance as demonstrated in [39]), it indicates an effective separation of the traffic components. Here we define the covariance ratio as follows:

$$R_{cov\_ratio} = \frac{R_{cov}}{\sigma^2(R(t))} \tag{3.12}$$

In the following, we use the OPNET simulated traffic presented in the previous section, as an example, for the demonstration of the network traffic separation strategy. Specifically, figure 3.8 shows the ratio between the covariance of the separated traffic and the total variance versus parameter m. We can observe from figure 3.8 that the ratio reaches the lowest point when parameter m is 5. The ratio 1.83% indicates a good separation of the baseline and the short-term traffic components. Here we can estimate the baseline

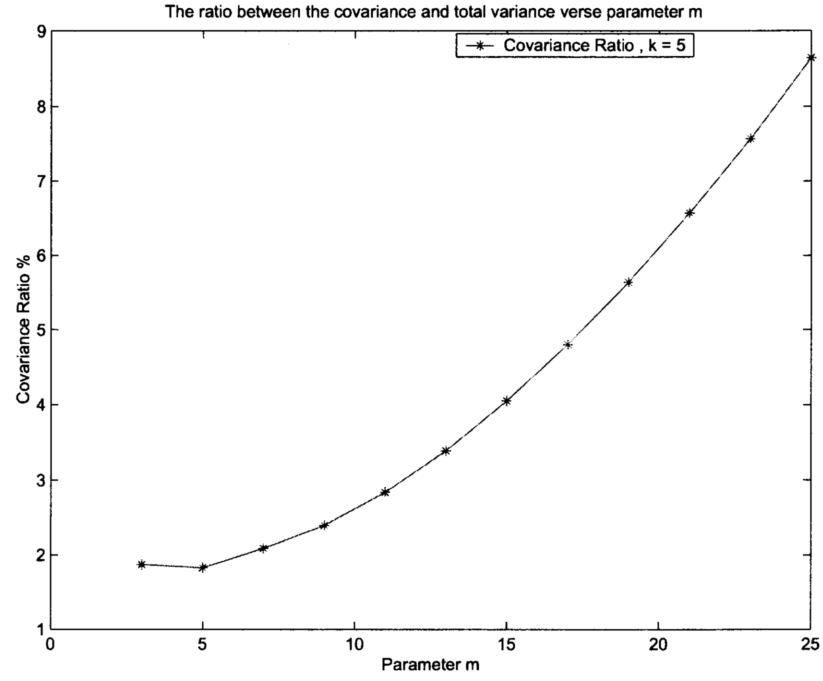

Figure 3.8 The ratio between the covariance and the total variance of the traffic versus parameter m

traffic by using a KZ filter with parameters(5,5). This KZ filter can be written as $KZ_{5,5}$. Here $KZ_{5,5}$ refers to five passes of a simple moving average of width 5. It should be noted that parameter k should be small enough to reduce the number of iterations, while at the same time should be sufficiently large for effective noise suppression in KZ filtering. The effective filter width can then be approximated, as explained before, as: $5(5^{1/2})$. Considering that the sampled data contain 5-minute interval data, the approximate separation frequency is $0.5/[5(5^{1/2})5]$ =0.008945, or separation time of approximately 111.8 minutes. Therefore the baseline contains phenomena that have a period longer than 111.8 minutes, and the short-term traffic contains the high-frequency processes.

Figure 3.9 and Figure 3.10 present the separated baseline traffic and short-term traffic respectively. From these figures we can observe that the baseline traffic is a mean non-stationary periodic traffic, which inherits most of the energy from the original traffic and includes most of the non-stationary components. On the other hand, the separated short-

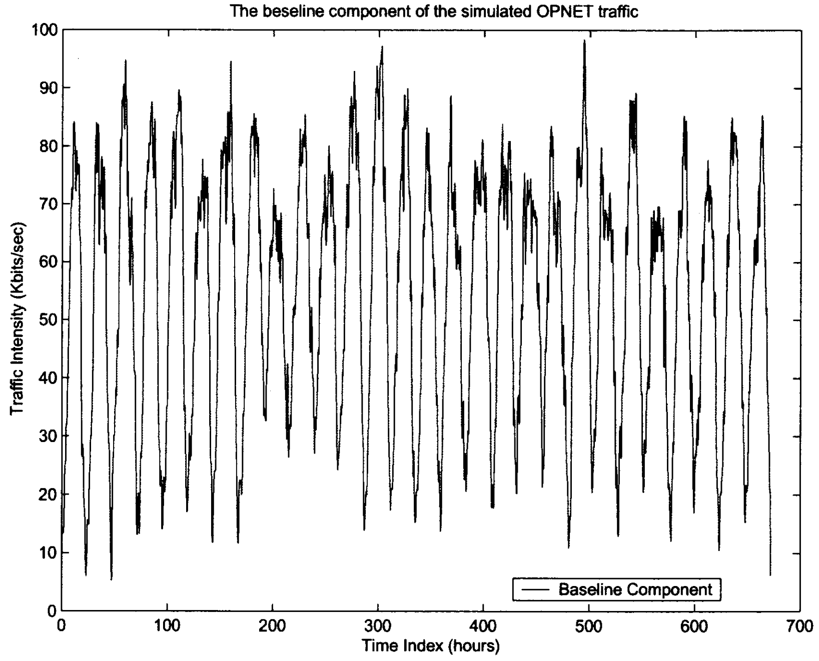

Figure 3.9 Baseline component traffic of the simulated OPNET traffic

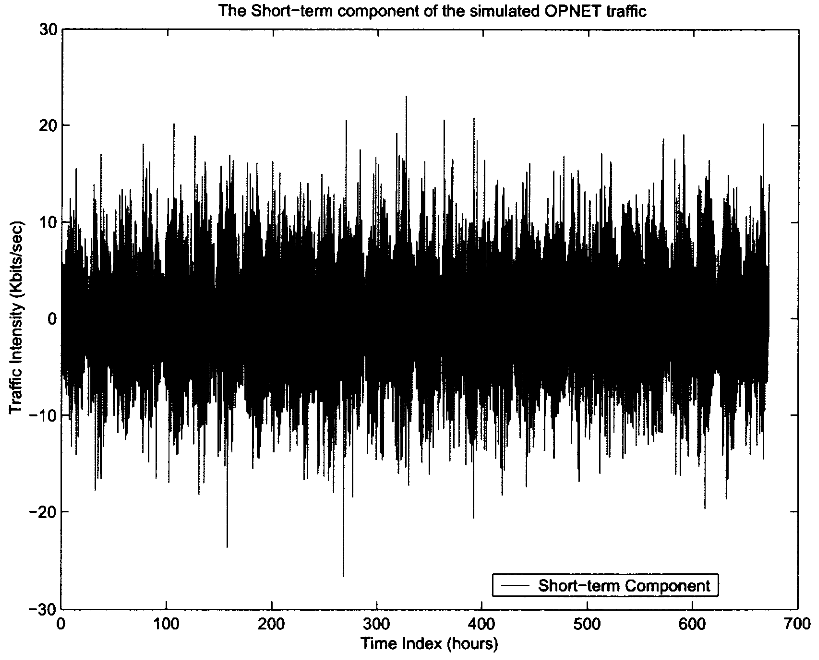

Figure 3.10 Short-term component traffic of the OPNET simulated traffic

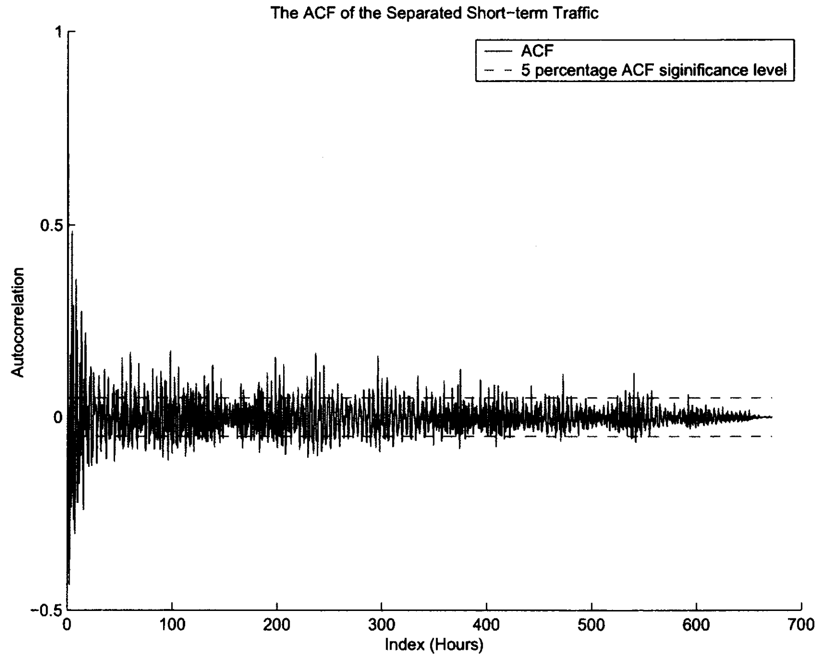

Figure 3.11 Autocorrelation function of the short-term component of the raw data

term traffic is a bursty high-frequency process with high variance, and it only includes a smaller part of the original traffic. Analyzing the Autocorrelation Function (ACF) of the short-term time series in figure 3.11, we can observe that all the autocorrelations are significantly outside the range defined by the two dashed-lines (which define a significance level of 5%). Here we assume that the autocorrelation function values that lie between the dashed-lines are statistically identical to zero (with a significance level of 5%). This suggests that the data contain significant time-dependency and therefore the short-term traffic is time-serial dependent. The prediction of this part is further complicated by the bursty characteristics of the short-term time series.

3.4 Accurate Network Traffic Prediction Strategies

3.4.1 Prediction of Short-term Traffic

In this subsection, we propose the application of the Autoregressive Moving Average (ARMA) model [43] to predict the short-term traffic, mainly due to the fact that ARMA model is distinguished by its ability to effectively predict time-dependent series. The following equation (3.13) presents the expression that characterizes the ARMA model.

$$S(t) = \sum_{i=1}^{p} \phi(i)S(t-i) + \sum_{j=0}^{q-1} \psi_j e(t-j) \qquad (3.13)$$

The first and second terms on the right-hand side of equation (3.13) correspond to the AR and MA models respectively. Here $S(t)$ denotes the short-time series, p and q represent the autoregressive and moving average orders of the AR and MA models respectively. $\phi(i), i = 1, ...p$ and $\psi(j), j = 1, ..., q$ are ARMA parameters, while we assume that the noise e(t) is a sequence of Independent and Identical Distributed(IID) Gaussian random variables with a mean of zero and a variance of σ^2, (i.e. IID(0, σ_e^2)). Without loss of generality, in the following we assume that $\phi(i) \neq 0$, $\psi(i) \neq 0$ and $\psi(0) = 1$. In order to predict the short-term traffic accurately, the AR model's order p should be optimized. In general, the optimum order p_{opt} of an AR model is chosen as the optimizer of a predetermined order selection criterion. Several order selection criteria, such as Akaike's Final Prediction Error Criterion [44] and Schwarz's Bayesian Criterion [45], have been developed and proposed in the literature. It is demonstrated in [46] that Schwarz's Bayesian Criterion, in general chooses the correct model order in most of the cases, and achieves the smallest mean-squared prediction error of the corresponding fitted AR models. For a detailed discussion regarding the demonstration of the order optimization, the interested readers may refer to [46, 47].

In the following, we present the methodologies used for the ARMA parameters estimation. Specifically, equation (3.13) can be rewritten as follows:

$$\Phi_p(A)S(t) = \Psi_q(A)e(t) \qquad (3.14)$$

where e(t) is a IID(0, σ_e) Gaussian white noise,

$$\Phi_p(A) = (1 - \phi_1 A - \phi_2 A^2 - ... - \phi_p A^p) \qquad (3.15)$$

and

$$\Psi_q(A) = (1 - \psi_1 A - \psi_2 A^2 - ... - \psi_q A^q) \tag{3.16}$$

Here $\sigma_e, \bar{\phi}_p = (\phi_1, \phi_2, ..., \phi_p)$ and $\bar{\psi}_q = (\psi_1, \psi_2, ..., \psi_q)$ are all unknown parameters that need to be estimated. In order to decide the autoregressive order p and the moving average order q in the ARMA(p,q) model, Schwarz [45] introduced a Bayesian criterion based on Laplace's asymptotic approximation. In the following, we provide an overview of the Schwarz's Bayesian criterion (SBC) criterion and describe its applicability in our proposed methodology.

3.4.2 Applying SBC Criterion

Let S_n denote the sequence of the separated short-term traffic. We assume that S_n can be described using a suitable model M_w which is selected from a sequence of candidate models $M_1, ..., M_L$, which are not necessarily nested. Here each model's parameters can be expressed using a vector $\bar{\delta} = (\sigma_e, \bar{\phi}_p, \bar{\psi}_q)$. This vector lies in a parameter space $\Delta(w) \subseteq \Re^w$. The predictive density of $\bar{S} = (S_1, S_2, ..., S_n)$ under the ARMA(p,q) can be defined as

$$d_{(p,q)}(\bar{S}) = \int L_{(p,q)}(\bar{\delta}|\bar{S}) \cdot \pi_{(p,q)}(\bar{\delta}) d\bar{\delta} \tag{3.17}$$

where $L_{(p,q)}(\bar{\delta}|\bar{S})$ is a likelihood function and $\pi_{(p,q)}(\bar{\delta})$ is a prior density under the ARMA(p,q) model.

Under Laplace's asymptotic approximation, the predictive density of $\bar{S}$ can be approximated as follows [48]:

$$d_{(p,q)}(\bar{S}) \approx \{L_{(p,q)}(\hat{\bar{\delta}}|\bar{S})|\hat{I}_{(p,q)}|^{-\frac{1}{2}}\} \cdot \{(2\pi)^{\frac{p+q+1}{2}} \pi_{(p,q)}(\hat{\bar{\delta}}) \tag{3.18}$$

Here $\hat{I}_{(p,q)}$ is the observed information matrix and $\hat{\bar{\delta}}$ is the maximum likelihood estimation of $\bar{\delta}$ under ARMA(p,q) model. As shown in [49], SBC criterion ignores the term in the second brace of equation (3.18) since this part is irrelevant for the purpose of model selection. Then, ignoring this term and applying (-2ln) function on relation (3.18), we can approach the SBC criterion as follows:

$$SBC(p,q) = -2\ln L_{(p,q)}(\hat{\bar{\delta}}|S_n) + D_w \cdot \ln n \tag{3.19}$$

Here $L_{(p,q)}(\hat{\bar{\delta}}|S_n)$ denotes the corresponding empirical likelihood, while D_w is the dimension of M_w and it actually represents the number of functionally independent parameters in $\bar{\delta}$, which here equals to (p+q+1). The basic idea of the SBC criterion is to find suitable values for p and q to maximize the predictive density in equation (3.17), while at the same time minimize the SBC(p,q).

After determining the optimized order of the ARMA model, we can estimate the corresponding parameters of the ARMA model. In order to improve the efficiency of the ARMA model, we define an upper bound $p_{\max}$ for the AR model order. We use the stepwise least squares algorithm [50] to calculate the optimum orders. The least squares estimates of AR parameters can be obtained by implementing the AR model in the form of an ordinary regression model, and using the method of least squares to estimate the regression model parameters [47]. In [51], Bjorck provides a standard method that involves the factorization of a data matrix for the least squares problem. We calculate the fitting model for different orders from 1 to $p_{\max}$, by stepwise downdating a regularized QR factorization of a data matrix for the model with order of $p_{\max}$. A QR factorization of matrix A is A=QR, where Q has orthogonal columns (i.e. Q^TQ=I) and R is upper triangular with non-zero entries on the diagonal. For an $a \times b$ matrix A with linearly independent columns (which means $a \geqslant b$), a QR factorization A, is defined as a full QR factorization, if Q is $a \times a$ and R is $a \times b$, (here Q is square and hence orthogonal, while R is not invertible unless a=b). Accordingly, a QR factorization of matrix A, is defined as a reduced QR factorization, if Q is $a \times b$ and R is $b \times b$ (here R is square and hence orthogonal, while Q is not invertible unless a=b). Therefore, given a full factorization, we can get a reduced factorization by discarding unwanted columns and rows. As result we first calculate the full QR factorization with order p_{max}, while then we can calculate the reduced QR factorization with order p_{opt}. Furthermore let us define:

$$p_{opt} = min(p_{opt}, p_{max}) \tag{3.20}$$

Then the approximate least squares estimates of parameters $\hat{\sigma}_e, (\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_{opt})$ can be computed for the model with order p_{opt} that optimizes the order selection criterion [50].

3.4.3 Evaluation Framework for Model Adequacy

Before the structure of the fitted AR model is analyzed or used for prediction, it is necessary to evaluate whether the fitted model provides an adequate representation of the given time series [52, 53]. In the following for simplicity in the representation, we choose one order MA model. In order to evaluate the model adequacy we need to determine whether the statistics of the residuals are consistent with the assumptions intrinsic to the ARMA model. The residual traffic can be expressed as follows:

$$\hat{e}(t) = S(t) - \sum_{i=1}^{p} \hat{\phi}(i) S(t-i) \qquad t = 1, ..., n \tag{3.21}$$

If we denote the estimated parameters by $\hat{\phi}(i)$, the predicted short-term traffic can be expressed as follows:

$$\hat{S}(t) = \sum_{i=1}^{p} \hat{\phi}(i) S(t-i) \tag{3.22}$$

The correlation of the residuals is tested using the estimate $\hat{R}(i)$ of the lag i correlation matrices that consist of the elements

$$\hat{R}(i) = \frac{\hat{c}(i)}{\hat{c}(0)} \qquad i = 1, ..., r \tag{3.23}$$

where r is the maximum lag and

$$\hat{c}(i) = \sum_{t=i+1}^{n} (\hat{e}(t-i) - \hat{u})(\hat{e}(t) - \hat{u})^T \tag{3.24}$$

contains the lagged residual cross-products. The mean of the residuals can be expressed as follows:

$$\hat{u} = \frac{1}{n}\sum_{t=1}^{n}\hat{e}(t) \tag{3.25}$$

It is shown in [54] that under the null hypothesis of model adequacy, for Gaussian noise, and for sufficiently large r, the quantity

$$Q_r = n\sum_{i=1}^{r} x^T_{\hat{R}(i)}(\hat{R}(0)^{-1} \otimes \hat{R}(0)^{-1})x_{\hat{R}(i)} + \frac{d^2r(r+1)}{2n} \tag{3.26}$$

is asymptotically χ^2-distributed with $f = d^2(r-p)$ degrees of freedom. Here $x_{\hat{R}(i)}$ includes the components of the matrix $\hat{R}(i)$, parameter d is the dimension of the short-term time series, superscript ^{T} indicates the transposition, while the operation $U \otimes V$ denotes the Kronecker product of U and V. Assuming that U is an (a x c) matrix and V is a (b x d) matrix, the Kronecker product of U and V, which is an (ab x cd) matrix can be expressed as follows:

$$U \otimes V = \begin{bmatrix} u_{1,1}V & u_{1,2}V & \cdots & u_{1,c}V \\ u_{2,1}V & u_{2,2}V & \cdots & u_{2,c}V \\ \vdots & \vdots & \vdots & \vdots \\ u_{a,1}V & u_{a,2}V & \cdots & u_{a,c}V \end{bmatrix} \tag{3.27}$$

According to the asymptotic distribution of the Li-McLeod statistic [54], with respect to Q_r, we have the following: the hypothesis that the residual components are uncorrelated will be rejected with approximate significance level ρ, if $Q_r > \chi^2_{1-\rho}(f)$, where $\chi^2_{1-\rho}(f)$ is a solution of

$$\rho = 1 - \acute{\Gamma}(\frac{f}{2}, \frac{\chi^2_{1-\rho}(f)}{2}) \tag{3.28}$$

and

$$\acute{\Gamma}(\alpha, \beta) = \frac{1}{\Gamma(\alpha)} \int_0^{\beta} t^{\alpha-1} e^{-t} dt \tag{3.29}$$

is the incomplete gamma function. Here

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} dt \tag{3.30}$$

Equivalently, if the significance level $\rho_{Q_r} \leq \rho$, the hypothesis that the residual components are uncorrelated will be rejected. Here we define $\rho_{Q_r} = 1 - \acute{\Gamma}(\frac{f}{2}, \frac{Q_r}{2})$ and ρ denotes the probability that the hypothesis that the residual components are uncorrelated will be rejected. Typically, we choose $\rho = 0.05$ which indicates a 5% significance level. This means that, if the correlation of the residual components is below 5%, they can be treated as statistically uncorrelated.

Figure 3.12 presents the residual component of the short-term traffic after the AR model has been applied, while Figure 3.13 shows the corresponding autocorrelation function (ACF). From the latter figure, we observe that the ACF of the residual traffic is located within the two dashed-lines (that represent a significance level of 5%), which is consistent with our assumption that the e(t) is a sequence of independent random variables, and clearly demonstrates that the time-dependent model is well fitted.

3.4.4 Baseline Traffic Prediction

As it was explained in chapter 2, the Extended Resource-Allocating Network (ERAN) algorithm [20] is a noise-tolerant non-stationary traffic prediction technique, which is capable of removing both single pulse noise and continuous anomalies, and allows for accurate and effective prediction of non-stationary traffic with low burstiness. Since, as it was demonstrated before, the baseline traffic presents these characteristics, ERAN algorithm will be applied for the prediction of this component. In addition to its prediction accuracy, the ERAN algorithm is used here, due to its simplicity and low complexity, especially when

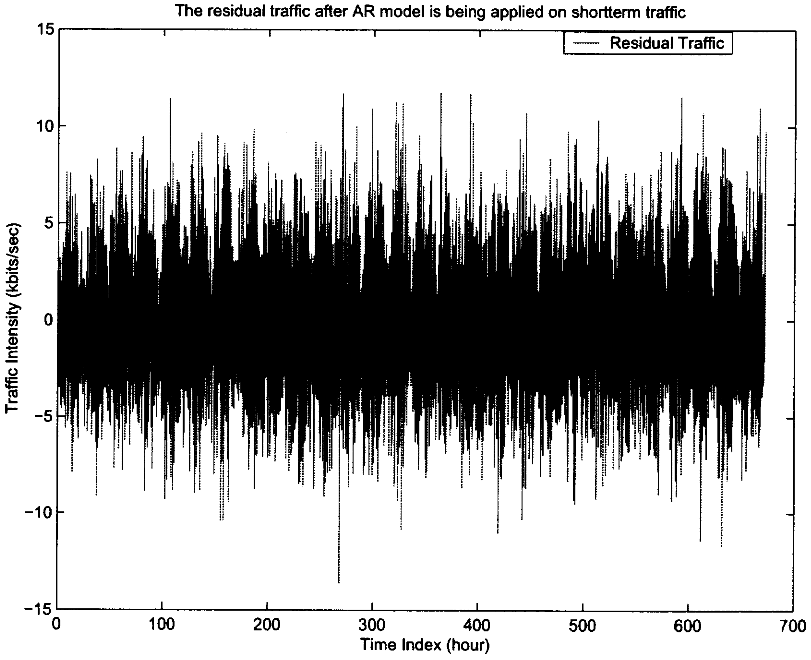

Figure 3.12 The residual component of the short-term traffic

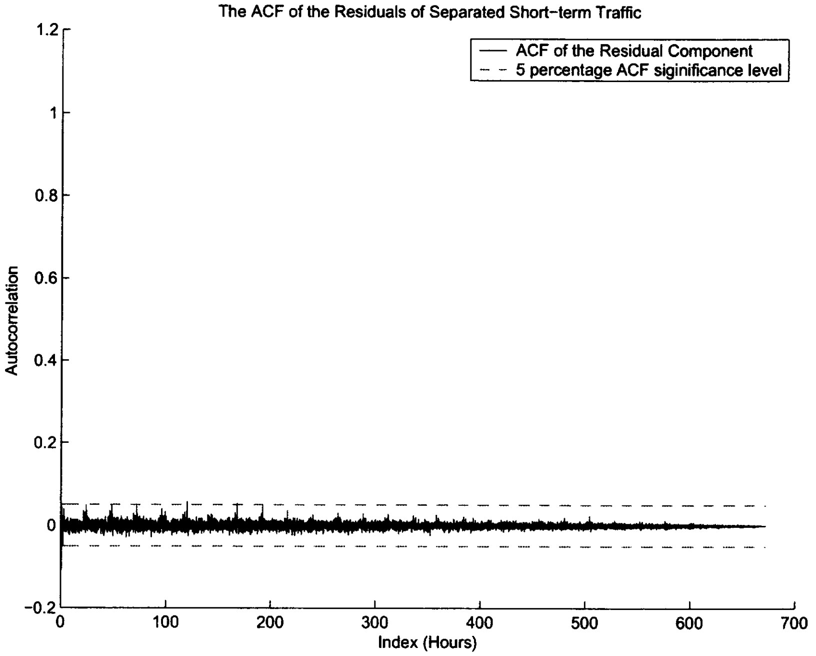

Figure 3.13 Autocorrelation function of the residual component of the short-term traffic

compared to other schemes which present high complexity when the number of samples is large.

3.5 Performance Evaluation and Discussion

In this section we evaluate the operational effectiveness and performance of our proposed methodology, via modeling and simulation using the Optimized Network Engineering Tool (OPNET). To achieve that, we carried out two sets of experiments. In the first one, the objective is to evaluate the performance of the proposed strategy in terms of the achievable accuracy and effectiveness in predicting the network traffic, especially in realistic traffic scenarios where high burstiness may be present. In the second one, the emphasis is placed on the evaluation of the corresponding improvement that can be achieved with respect to the network anomaly detection capabilities, and therefore the network performance evolution in the presence of intrusion traffic is studied.

3.5.1 Models and Assumptions

Network Architecture and Simulated Traffic

A high-level diagram of the corresponding network architecture that was used throughout this evaluation is shown in Figure 2.9. It consists of three subnetworks connected by 3 routers. The various clients are located in all the three subnetworks, i.e. Subnet_1 (Ethernet Network), Subnet_2 (Fast Ethernet Network) and Subnet_Server (FDDI Network). Each of the three subnetworks supports several clients that can establish communication with the Main Server located in Subnet_Server, which can support several applications such as http, ftp and e-mail applications.

The corresponding simulated data for four weeks of network traffic was presented in figure 3.5. In this figure, the vertical axis represents the traffic intensity in kbits/sec while the horizontal axis represents the time evolution (in hours). It should be noted that the data used here are for demonstration purposes only, in order to mainly represent the trend and the behavior of the expected Internet traffic, while the actual values could be different and usually depend on several parameters including the number of users, the traffic density, the applications etc.

Attack/Anomaly Model

The UDP flooding attack was modeled and used throughout our evaluation, for the second set of the experiments. It should be noted, that the UDP flooding attack represents one of the most common Denial of Service (DoS) types of attacks, and was used for demonstration mainly purposes here. Additional types of attacks such as TCP/SYN flooding, Internet mail-bombing attack, etc. have also been modeled. For comparison purposes, the same UDP flooding attack described in chapter 2, has been used here as well.

3.5.2 Accurate Network Traffic Prediction with the Proposed Separation-Combination Strategy

In this subsection, we first present some numerical results regarding the prediction of each traffic component separately (baseline and short-term traffic), while next the capabilities of the proposed methodology in predicting the total combined traffic are evaluated. In order to quantify and obtain a better understanding of the improvements achieved by the proposed traffic separation and combination strategy, we also compare its performance with the corresponding performance obtained when only the ERAN methodology is utilized to predict the total traffic (without using the proposed separation strategy). Figure 3.14 presents the original baseline traffic (depicted by the solid line) as well as the predicted baseline traffic based on the ERAN methodology (curve with asterisks). The vertical axis shows the corresponding traffic intensities (in Kbits/sec), while the horizontal axis depicts the time index (hours), as the system evolves. From these results we observe that the baseline prediction is very accurate by utilizing the noise-tolerant ERAN algorithm. The main reason for the high prediction accuracy of this part of the traffic, is due to fact that the bursty high-frequency traffic has already been removed from the original traffic, as described in section 3, and therefore the remaining (baseline) traffic has low burstiness, which can be well predicted by ERAN algorithm. The main advantage here is that we can predict the pattern-mixed low frequency baseline traffic with high prediction accuracy. In order to predict the bursty short-term high-frequency traffic, we implemented the proposed ARMA model. Figure 3.15 is the predicted short-term traffic based on the ARMA model for all data. In figure 3.16 we compare the predicted short-term traffic based on the ARMA model, with the original separated short-term traffic, between the time indices 500 and 550. Although throughout that time interval, the prediction errors are still high in some instances, even by utilizing the ARMA model, from figure 3.13 we observed that the ACF of the residual traffic is below 5%, which indicates that the time-dependent AR model is well fitted.

The combined predicted traffic $\hat{R}(t)$, based on the prediction of the individual components, can be expressed as follows:

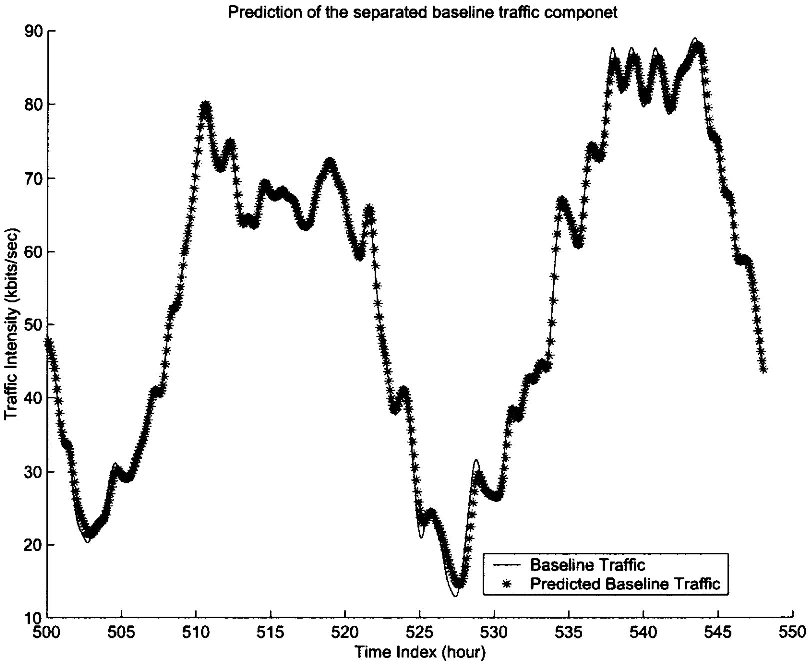

Figure 3.14 Prediction of the separated baseline traffic component

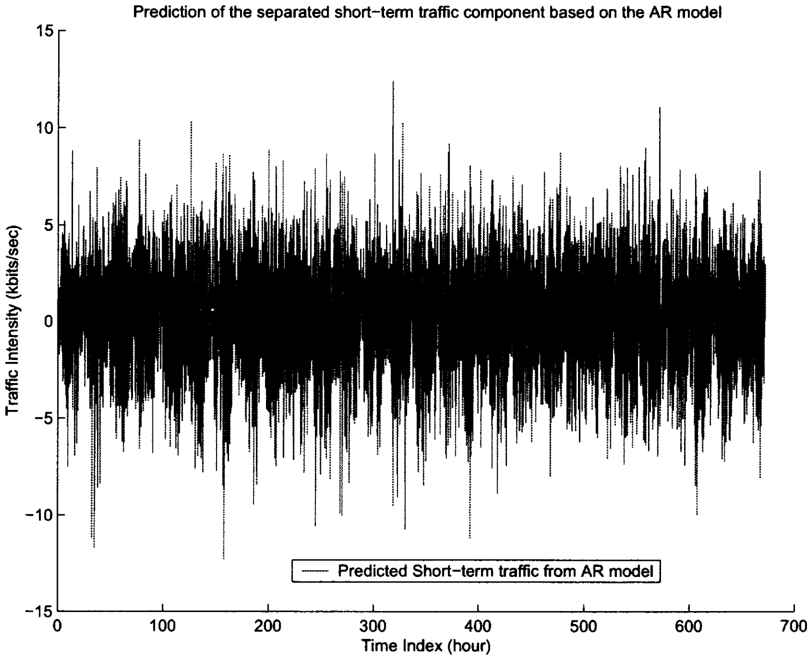

Figure 3.15 Separated short-term traffic prediction based on the AR model

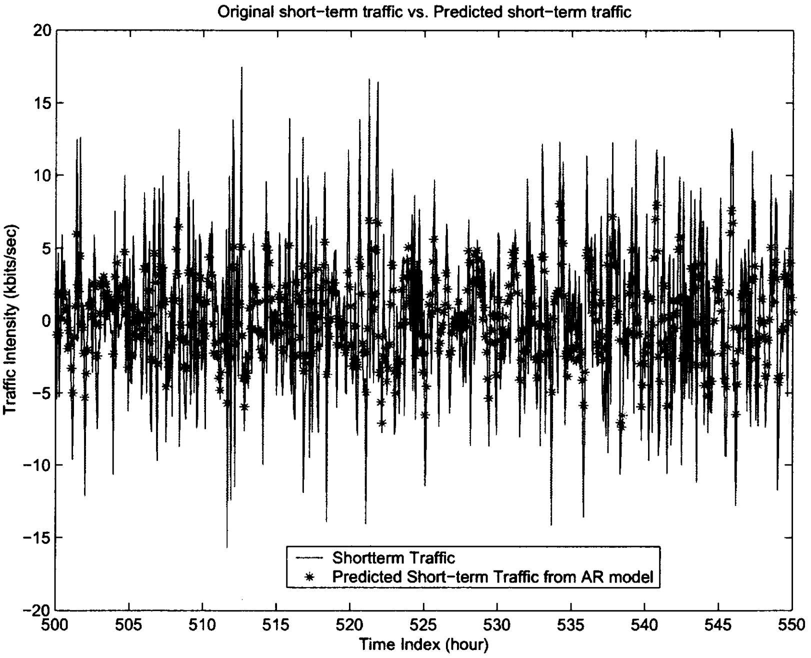

Figure 3.16 Original short-term traffic vs. predicted short-term traffic

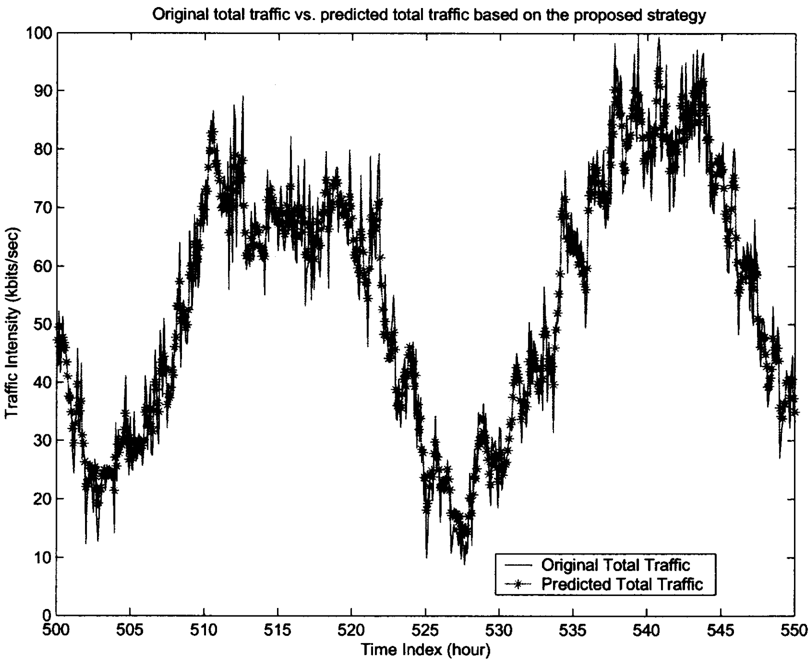

Figure 3.17 Original total traffic vs. predicted total traffic based on the separation and combination proposed strategy

$$\hat{R}(t) = Ba\hat{s}eline(t) + \hat{S}(t) \tag{3.31}$$

In figure 3.17, we present the combined total traffic intensity in Kbits/sec (vertical axis), versus the time index in hours (horizontal axis), as the system evolves. The curve with asterisks corresponds to the predicted combined traffic based on our proposed strategy, while the solid line depicts the original traffic. From this figure we can clearly see that our proposed algorithm predicts the traffic proactively and accurately. For comparison purposes, we also present in figure 3.18 the corresponding results obtained when only the ERAN methodology is utilized to predict the total traffic (without using the proposed separation strategies) [20].

Comparing figures 3.17 and 3.18, we observe that the prediction accuracy and corresponding effectiveness have been significantly improved by the use of the proposed

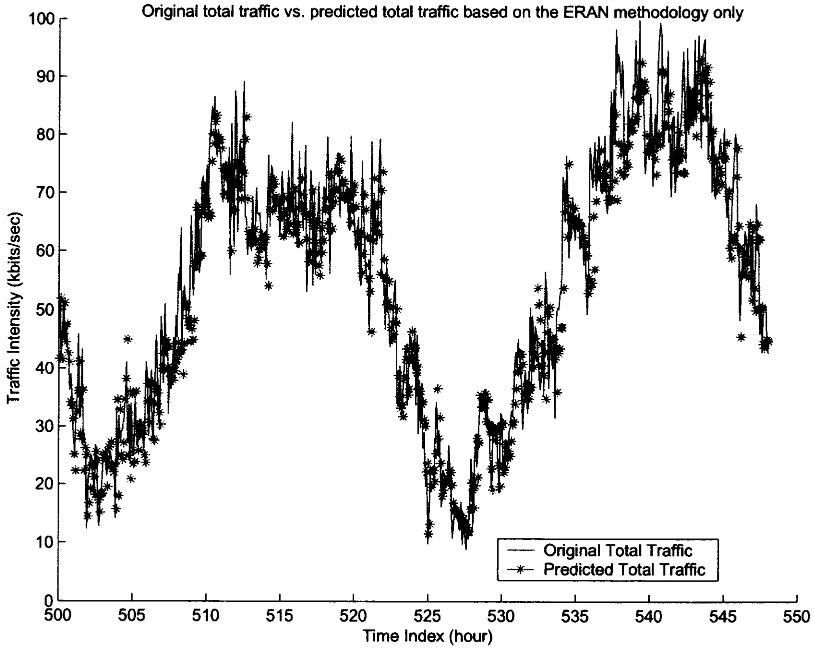

Figure 3.18 Original total traffic vs. predicted total traffic based on ERAN algorithm only

separation and combination strategy. In fact, the Mean Absolute Percentage Error (MAPE) drops from 7.927% to 6.205% when the separation-combination strategy is implemented, which indicates a 21.7% improvement in the prediction accuracy. Also, the corresponding variances of the absolute percentage error drops from 0.009802 to 0.006006, which indicates an improvement of approximately 33%. The reason for this improvement is two fold. First by separating the baseline traffic we removed the high burstiness traffic component, and therefore the prediction accuracy of the baseline traffic is considerably improved, when the ERAN algori- thm is applied to this part. This is very important and its impact on the overall prediction accuracy is very critical, since as explained before, the baseline traffic inherits most of the energy of the original traffic, while the bursty, short term traffic, contains only a smaller part of the original traffic. Second, as demonstrated above, the introduction and use of the ARMA model can further improve the prediction of the burst, high-frequency time-dependent short-term traffic. Therefore, the overall prediction accuracy improvement is

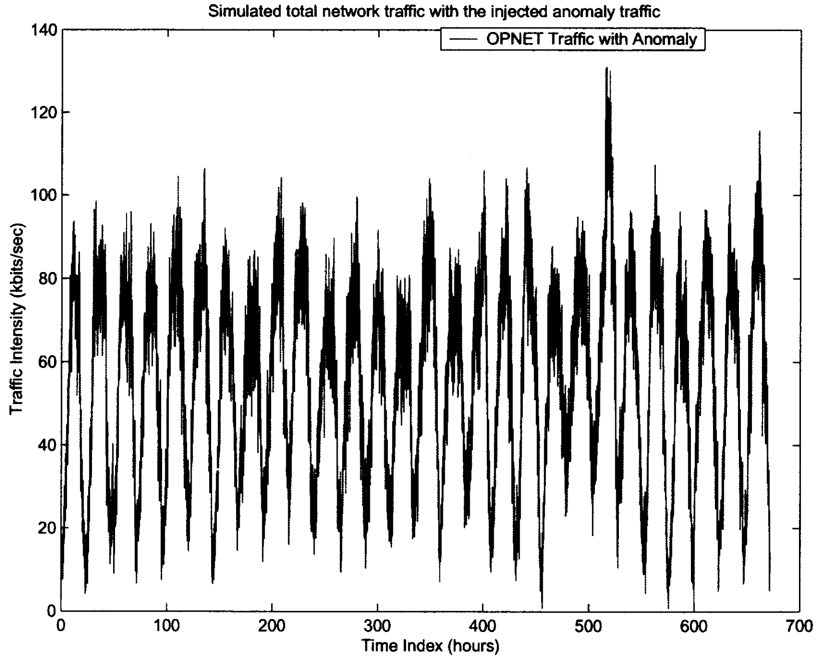

Figure 3.19 Simulated total network traffic with the injected anomaly traffic

due to the combined effect and outcome of both these interrelated factors and observations.

3.5.3 Network Anomaly Detection

In this subsection, we evaluate the operation of the proposed strategies and demonstrate the improved network anomaly detection capabilities, under a UDP flooding attack scenario. As a result, for this experiment we injected into the network some UDP flooding attack traffic from time index 515 to time index 525. Specifically, figure 3.19 presents the network traffic data, including the UDP Flooding attack traffic, while figures 3.20 and 3.21 depict the corresponding separated baseline traffic and short-term traffic, respectively, when the $KZ_{5,5}$ filter (described in section 3) is applied. We can observe from figure 3.20 that the baseline traffic contains most of the energy, while the majority of the anomaly traffic has been separated into the baseline traffic component.

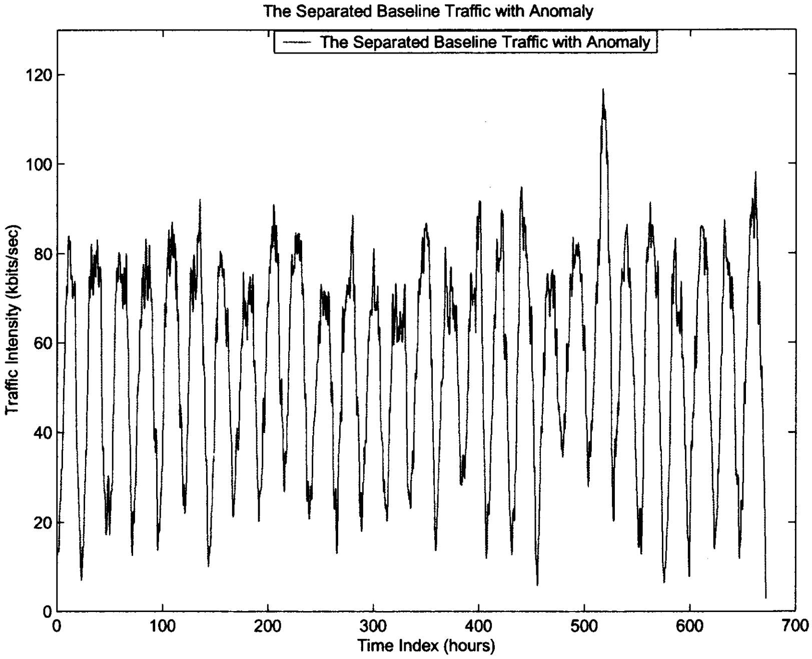

Figure 3.20 Separated baseline traffic with anomaly

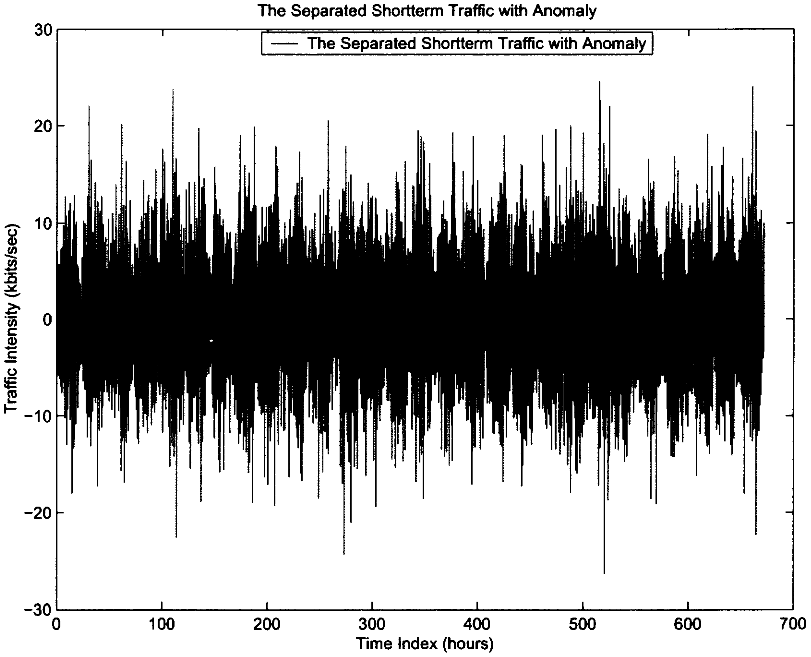

Figure 3.21 Separated short-term traffic with anomaly

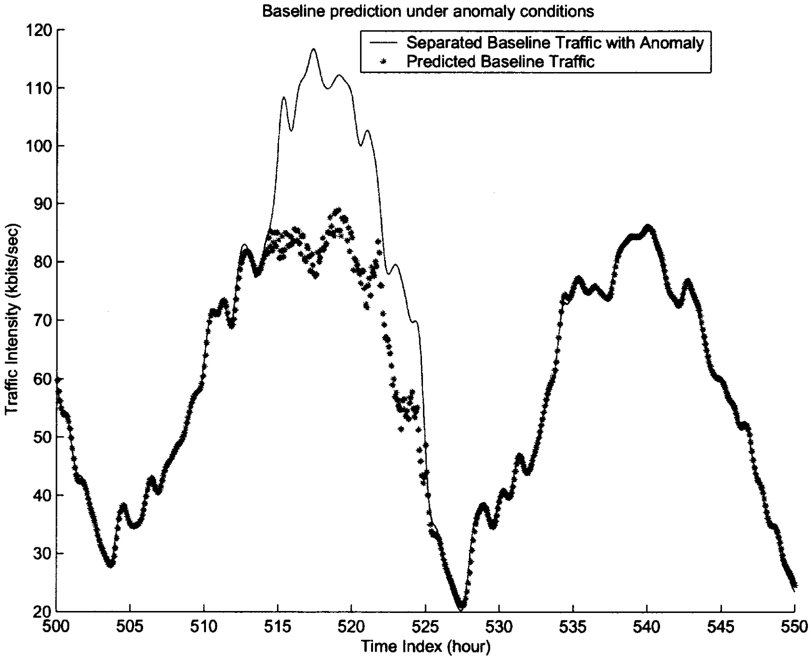

Figure 3.22 Baseline prediction under anomaly conditions

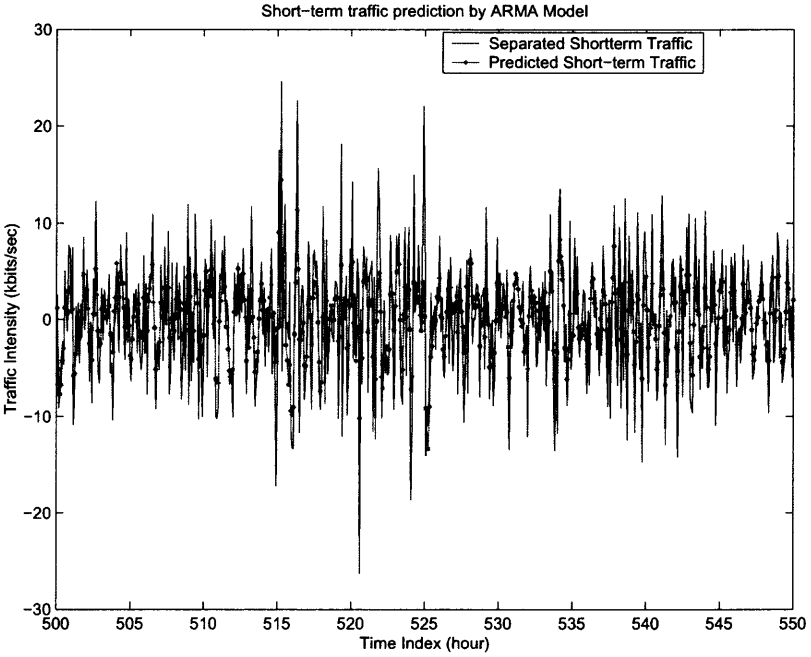

Figure 3.23 Short-term traffic prediction by ARMA model

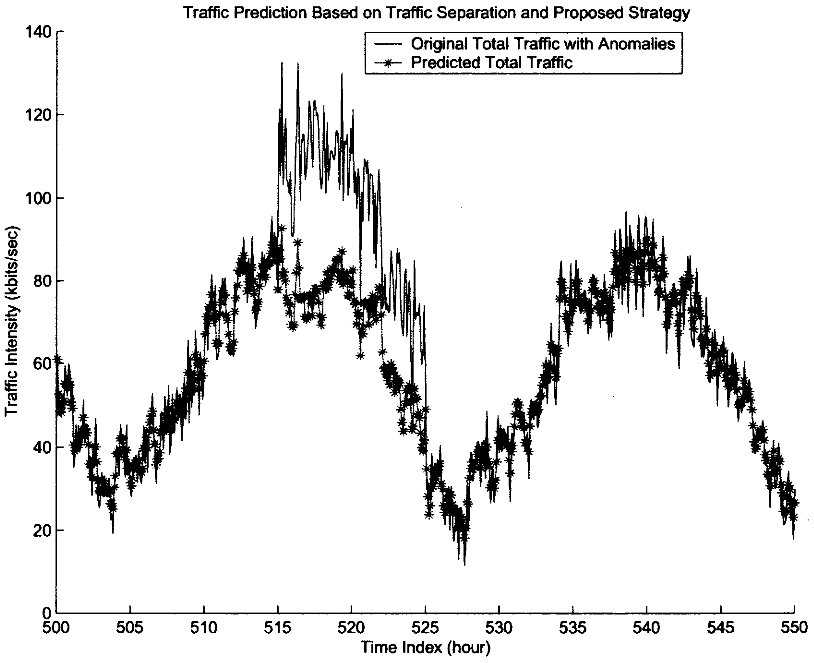

Figure 3.24 Traffic prediction based on traffic separation and proposed strategy under anomaly status

Figure 3.22 presents the original (solid line) and predicted (with asterisks) baseline traffic component, between the time indices 500 to 550. As we observe from this figure, the predicted baseline traffic does not follow the anomaly injected traffic, and therefore our proposed strategy provides an accurate and anomaly-tolerant strategy for predicting this type of traffic. The corresponding results, with respect to the short-term traffic where the ARMA model is used for prediction, are presented in figure 3.23 (the curve with diamonds corresponds to the predicted short-term traffic based on ARMA model, while the solid line depicts the separated original short-term traffic).

In figure 3.24 we present the total traffic as it evolves, as well as the total combined predicted traffic, while figure 3.25 depicts the total traffic and the dynamic upper and lower thresholds as they are defined by the anomaly detection strategy, based on the predicted total traffic. As can be seen from figure 3.25, the anomaly traffic that crosses the dynamic upper thresholds (represented by the line with stars) or lower thresholds (represented by the

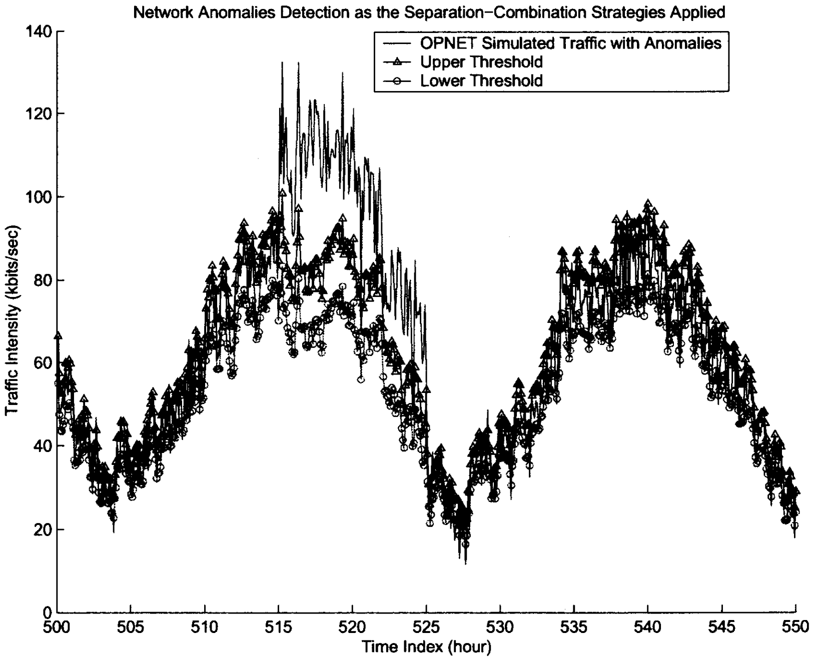

Figure 3.25 Network anomaly detection

line with squares), can be clearly detected, and therefore appropriate actions can be taken before the specific anomaly negatively affects the network performance.

Our results demonstrate that based on our approach, both the anomaly detection probability and false alarm probability can be improved. Specifically, figure 3.26 displays the detection probability vs. the false alarm probability, using Receiver Operation Characteristic (ROC) curves after the proposed separation-combination strategy has been applied on the traffic. Here the detection probability is defined as the probability that the abnormal traffic is being detected and recognized, while the false alarm probability is defined as the probability that some normal traffic events are classified as anomalies. The different curves correspond to different scenarios with respect to the ratio between the mean intrusion/anomaly traffic and background traffic (i.e. anomaly ratio). The x-axis depicts the false alarm probability while the y-axis presents the corresponding detection probability. For each curve, the point at the upper left corner represents the optimal detection, with high detection probability and low false alarm probability. From this figure we confirm, that the detection

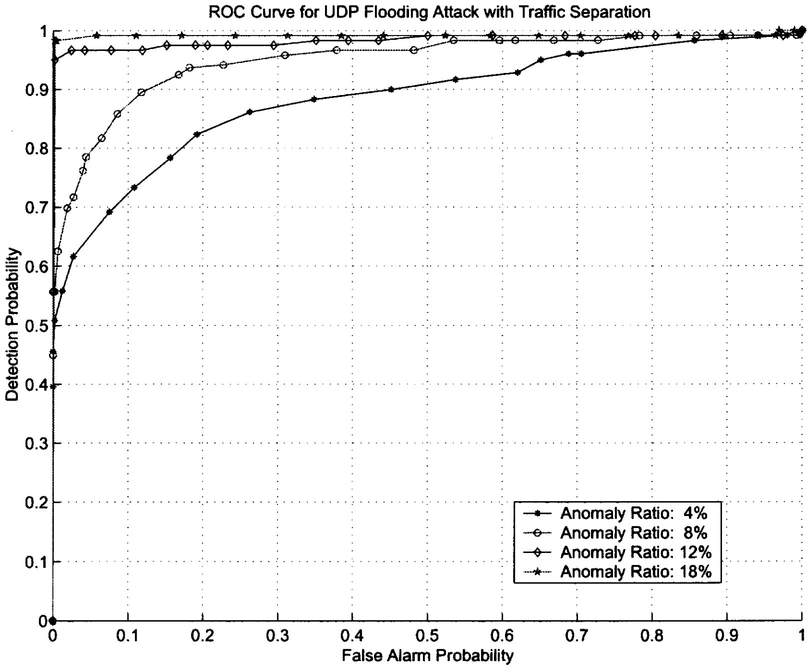

Figure 3.26 ROC curve for UDP flooding attack with traffic separation

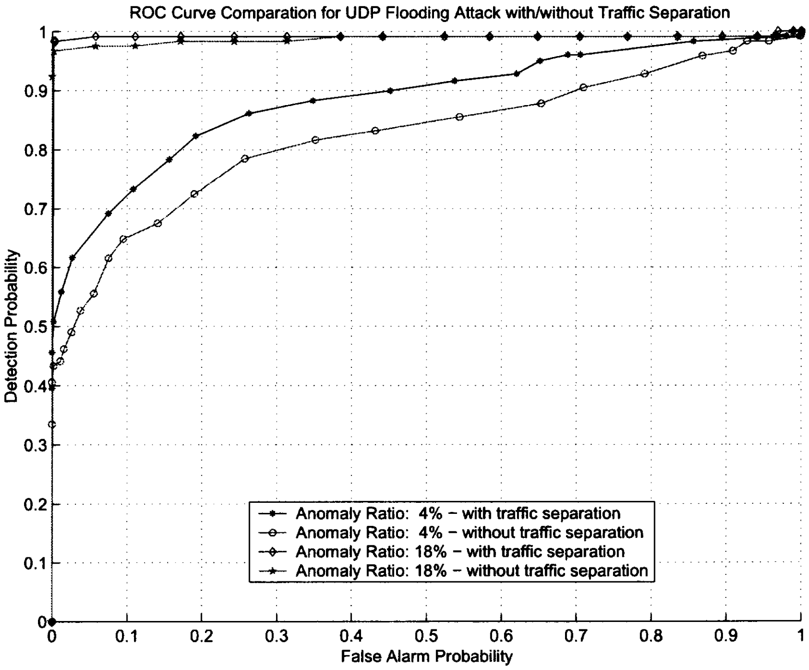

Figure 3.27 ROC curve comparison for UDP flooding attack (with/without traffic separation-combination strategy)

performance improves as the anomaly ratio increases. We also observe here, that the detection probability increases, as a result of the improvement of the prediction accuracy.

This is more clearly demonstrated in figure 3.27, where we compare the ROC curves under the proposed separation-combination strategy, and the corresponding anomaly detection results that are obtained when only the ERAN methodology is utilized to predict the total traffic (without using the proposed traffic separation strategies). We can observe from this figure that the anomaly detection probability increases, while the false alarm probability decreases when the separation-combination strategy is applied, which indicates a significant improvement of the anomaly detection performance.

CHAPTER 4

CONCLUSION

Efficient network and traffic management, and timely identification of potential performance degradation and anomalies, are required to increase the cost effectiveness and QoS support of multimedia applications in next generation Internet. As we increasingly rely on information systems, computers and networks, to support critical operations in telecommunication, banking, electronic commerce, defense and other systems, intrusions present serious obstacles and threats on the deployment of various computing systems and networks. The information technology advances that provide new capabilities to the network users and providers, also provide with new and powerful tools the network intruders that intend to launch attacks on critical information resources. Therefore, algorithms and methodologies that are capable of detecting and diagnosing performance degradations and/or network anomalies automatically are in high demand.

In this dissertation, we proposed, developed and evaluated network anomaly detection methodologies, which rely on the analysis of network traffic and the characterization of the dynamic statistical properties of traffic normality, in order to accurately and timely detect network anomalies. Specifically, in chapter 2, we proposed a new anomaly detection algorithm in order to detect novel anomalies in the Internet. Our approach is based on the concept that perturbations of normal behavior suggest the presence of attacks and anomalies. Therefore our methodology first develops and identifies normal behavior patterns and profiles (i.e. the expected behavior), in order to identify any unacceptable and significant deviation from the usual behavior, as possibly a network attack. Due to the observation that Internet traffic presents non-stationary characteristics, as part of our approach, we first proposed and developed ERAN, an anomaly-tolerant non-stationary traffic prediction technique, which is capable of removing both single pulse noise anomalies and continuous anomalies, and therefore allowing for accurate traffic prediction. Motivated by this approach, we also designed dynamic thresholds and created anomaly detection/violation conditions, in order to study and demonstrate how the developed anomaly-tolerant non-stationary traffic prediction algorithm can be used as the basis to effectively detect possible network attacks. The operational efficiency and effectiveness of the proposed methodology was tested and evaluated under different attack scenarios, such as mail-bombing attack and UDP flooding attack, and the corresponding numerical results demonstrated that it can accurately and proactively detect potential network attacks and traffic anomalies.

A new network traffic separation-combination strategy was also proposed in this dissertation in order to improve the network traffic prediction and network anomaly detection, especially in realistic traffic scenarios where high burstiness may be present. Our approach is based on the observation that the various network traffic components, that may be caused by different temporal and in some cases physical phenomena, are better identified, represented and isolated in the frequency domain. Therefore, a new methodology, that analyzes and filters the traffic data in the frequency domain, was first introduced and designed. As a result of this separation strategy, the traffic is divided into two main components: the baseline component and the short-term component. The baseline component includes most of the low frequency and non-stationary traffic and presents low burstiness, while the short-term component includes the most dynamic part. Furthermore, the relationship between the cut-off frequency and the residual covariance of the baseline traffic and the short-term traffic was discussed as well.

As it is demonstrated in chapter 3, the traffic components within different frequency ranges present different characteristics, and therefore in order to achieve high prediction accuracy and improve the operational effectiveness in the Internet, it would be more efficient to apply appropriate strategies on the separated traffic components. Specifically, since based on the traffic separation strategy proposed here, the baseline traffic component is a mean non-stationary periodic signal with low burstiness, the ERAN algorithm was used for the efficient and accurate prediction of this traffic part. With respect to the short-term traffic, which was shown to be a time-dependent series, the ARMA model is proposed to be used for the prediction of this component. Then, the overall predicted traffic can be obtained by the combination of the predicted individual separated traffic components.

The performance evaluation process and the corresponding numerical results presented here, demonstrated that the proposed methodology of separating the traffic based on the "frequency domain" analysis and predicting each component separately by the appropriate method, improves significantly the prediction accuracy of the total combined Internet traffic. Since, the effectiveness of the anomaly detection strategies is directly correlated with the traffic prediction accuracy, we also demonstrated that using our proposed strategies, and combining them with the use of dynamic thresholds and adaptive anomaly violation conditions, we can improve significantly both the anomaly detection probability and the false alarm probability, therefore enhancing the operational effectiveness of the Internet.

It should be noted that in the work presented in this dissertation the emphasis is placed on the accurate traffic prediction and the detection of unknown anomalies in Internet. In order to design and develop an integrated framework and novel system architecture that is capable of improving the overall management and security related operational capabilities of Internet, the proposed anomaly detection approach can be integrated with methods that intend to identify and track the source of an attack or fault when such an occurrence is detected.

Finally, although the application of the proposed anomaly detection algorithm was demonstrated for next generation Internet, we believe that our research and framework can be applied for anomaly detection in various networking environments, such as:

- Wide area networks where proactive detection and timely recovery of service performance degradations (or soft faults) are important (compared with generating and processing hard network alarms)

- Next Generation Internet, Wide area networks and E-commerce infrastructures

- Multiple service-class networks where resource sharing is significant and service-class performances are correlated (e.g., in a multiple service-class transaction network, or a virtual private networking infrastructure)

- Networks where the non-managed parts and environmental components (e.g., customer-site equipment, and attached networks) can fail and consequently degrade the performance of the managed network proper.

- Next generation wireless networks supporting multimedia services

Although the differences between the various networking environments (e.g. wireless networks and the wired networks) are known and well documented in the literature and practice, we believe that similar concepts and methodologies can be applied in both kind of networks, since in any case the objective is the same for the network/service anomaly detection and optimal use of resources. The main differences would lie in the observables to be measured and monitored, the objective functions to be constructed, the shared resources to be controlled, and the environment/infrastructure to be applied. Additionally, in wireless mobile networks the offered traffic may vary both temporally and spatially, with the spatial variation significantly higher than in wired networks.

APPENDIX

SEQUENTIAL LEARNING IN ARTIFICIAL NEURAL NETWORKS

Artificial neural networks estimate a sequential data mapped by a Gaussian Radial Basis Function (GaRBF) network that adds a hidden unit to each observation. It is basically a sequence learning approach in which the task is to estimate an underlying function sequentially. The learning procedure of the ANN is the mapping from a set of data in the form of input - output observation pairs(X_n, y_n), where X_n is an M-dimensional input vector and y_n is an output scalar. The input lies in a subset $\mathcal{S}$ of the space of all real valued M-dimensional vectors R^M . The n^{th} observation can then be described as:

$$\hat{O}^{(n)} = \{(X_n, y_n) : X_n \in \mathcal{S} \subseteq R^M; y_n \in R\} \tag{.1}$$

When the observations $\hat{O}^{(n)}$, n= 1,...,N are assumed to be free of noise and consistent with an underlying function f_*, we have

$$f_*(X_n) = y_n \quad n = 1, ..., N \tag{.2}$$

The difference between the ANN mapping and the underlying function is measured by some distance metric $D(f, f_*)$. The L^2-norm is one common and popular metric used with ANNs, which is given by,

$$D(f, f_*) = \|f - f_*\| \tag{.3}$$

where $\|\|$ denotes the L^2-norm. The squared-norm is given by,

$$\|f\|^2 = \int\limits_{X \in \text{Ş}} |f(X)|^2 \, dX \tag{.4}$$

Where $|.|$ is the absolute value of its argument. The L^2-norm describes a function space which contains all the square integrable real valued functions. Since an inner product can also be defined in this space, it is a Hilbert space, denoted by,

$$H = \{f : \|f\| < \infty\} \tag{.5}$$

The mapping described by an ANN satisfies the above requirement in general and hence all possible functions an ANN can describe lie in the function space H. The inner product between two functions is given by [55],

$$< f, g >= \int\limits_{X \in \text{Ş}} f(X) g(X) dX \tag{.6}$$

When the output of the network is not limited to the interval (0 1), the hidden-output layer transformation is linear [56]. Then the single hidden layer ANN linearly combines the output of the hidden units. Each of the hidden units construct a mapping and hence these mappings can be viewed as the basis functions $\phi_j \in H, j = 1, ..., J$, the total number of hidden units being J. The output is thus represented as,

$$f(X) = \sum_{j=1}^{J} \alpha_j \phi_j(X) \tag{.7}$$

In sequential estimation, an estimate is required at each time instant n. The ANN mapping after it has learned from the n^{th} observation $\hat{O}^{(n)}$ is denoted by $f^{(n)}$. This is known as the posterior estimate of the underlying function and $f^{(n-1)}$ as the prior estimate.

The sequential function estimation problem can be stated as follows: Given the prior estimate $f^{(n-1)}$ and the new observation $\hat{O}^{(n)}$, how can we obtain the posterior estimate $f^{(n)}$? Here we assume that the observations are free of noise. One approach to sequential estimation is to choose an optimal estimate at each step. The step-wise optimal estimate is given by the principle of the F-Projection[57], which states,

$$f^{(n)} = \arg\min_{f \in H} \|f - f^{n-1}\| \qquad f^{(n)} \in H_n \tag{.8}$$

where H_n is the set consisting of all the functions in H that satisfy the constraint $f(X_n) = y_n$. The posterior is a projection of the prior onto the space H_n. The principle is an analogue of the projection algorithm for linear models [58], where the prior parameter vector is projected onto the constraint hyperplane in the parameter space. The equality constraint $f(X_n) = y_n$ can be rewritten as an inner product in the function space,

$$< f, \delta_n >= y_n \tag{.9}$$

where $\delta_n = \delta(X - X_n)$ is the impulse function . The constrained minimization can be solved exactly to give,

$$f^{(n)} = f^{(n-1)} + e_n h_n \tag{.10}$$

where e_n is the prediction error, given by,

$$e_n = y_n - f^{(n-1)}(X_n) \tag{.11}$$

and $h_n = \frac{\delta_n}{\|\delta_n\|^2}$. This solution adds a spike at the point X_n to $f^{(n-1)}(X)$ such that $f^{(n)}(X)$ goes through the point (X_n, y_n). It discounts the fact that the underlying function is smooth and an observation has a bearing on its neighborhood in the input space $\mathfrak{S}$. Smoothness constraints must then be added to obtain a posterior estimate. The smoothness constraint must be imposed on which has the following property:

$h_n(X_n) = 1$ and $h_n(X_n + s) = 0$ for any $s \neq 0$. Smoothing this impulse like function subject to the constraint that $f^{(n)}(X) = y_n$, yields the Gaussian RBF , given by,

$$\phi_n(X) = \exp\{-\frac{1}{\sigma_n^2} \|X - u_n\|^2\} \tag{.12}$$

with $u_n = X_n$ and $\sigma_n$representing the required smoothness . Now the properties of σ_n are: $\phi_n(X_n) = 1$ and $\phi_n(X_n + s) \to 0$ as $\|s\| \to \infty$. The parameter σ_n is the spread of the GaRBF representing its span around in the input space. This view is similar to the method of the potential functions [26] where each observation in the input space contributes to its neighborhood via the potential of a charge placed on the observation, the span signifying the region of influence of the charge. Hence, from the principle of F-Projection and smoothing its solution, we have the posterior function estimate $f^{(n)}$, given by,

$$f^{(n)}(X) = f^{(n-1)}(X) + e_n\phi_n(X) \tag{.13}$$

Let us use the GaRBF network to map the function estimate $f^{(n-1)}$. Assume that there are J hidden units (basis functions) in the network that maps $f^{(n-1)}$. Then the posterior is given by,

$$f^{(n)}(X) = \sum_{j=1}^{J} \alpha_j\phi_j(X) + e_n\phi_n(X) = \sum_{j=1}^{J+1} \alpha_j\phi_j(X) \tag{.14}$$

The posterior estimate is mapped by the GaRBF network with a new hidden unit added and the parameters associated with it are assigned as follows:

$$\alpha_{J+1} = e_n, u_{J+1} = X_n, \sigma_{J+1} = \sigma_n \tag{.15}$$

The network we have arrived at grows with each new observation. The observations X_n are implicitly stored as the centers of the Gaussian hidden units and e_n (hence y_n) are implicit in their coefficients. This estimate is similar in spirit to the Parzen window density estimation procedure where the number of kernels are the same as the number of observations and are centered on the input observations [26]. The difficulty with using this network for estimation is that the network grows indefinitely as the observations are continually received.

BIBLIOGRAPHY

[1] Talpade, R.; Kim, G.; and Khurana, S.; "NOMAD: Traffic-based Network Monitoring Framework for Anomaly Detection", *In Proc. IEEE International Symposium on Computers and Communications*, pp. 442-451, 1999.

[2] Yemini, Y.; "A Critical Survey of Network Management Protocol Standards", *In Telecommunications Network Management Into the 21st Century, (editors: S. Aidarous and T. Plevyak), IEEE Press*, New York, NY 1994.

[3] Caceres, R.; et. al, "Measurement and Analysis of IP Network Usage and Behavior", *IEEE Communications Magazine*, Vol. 38, No. 5, pp. 144-151, May 2000.

[4] Adams, A.; Tian, Bu; Friedman, T.; Horowitz, J.; Towsley, D.; Caceres, R.; Duffield, N.; Presti, F.L.; Moon, S.B.; Paxson, V.; "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior", *IEEE Communications Magazine*, Vol. 38, No. 5, pp. 152-158, May 2000.

[5] Ho, L.L.; Cavuto, D.J.; Papavassiliou, S.; Zawadzki, A.G.; "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.

[6] Dawes, N.; Altoft, J.; Pagurek, B.; "Network Diagnosis by Reasoning in Uncertain Nested Evidence Spaces",*IEEE Transactions on Communications*, Vol. 43, pp. 466-467, 1995.

[7] Chakrabarti, A.; Manimaran, G.; "Internet Infrastructure Security: A Taxonomy",*IEEE Network*, Vol. 16, issue: 6, pp. 13-21, Nov./Dec. 2002.

[8] Chang, R.K.C.; "Defending against flooding-based distributed denial-of-service attacks: a tutorial", *IEEE Communications Magazine*, Vol. 40, issue: 10 , pp. 42-51, Oct. 2002.

[9] Buschkes, R.; Kesdogan, D.; Reichl, P.; "How to Increase Security in Mobile Networks by Anomaly Detection", *In Proc. 14th Annual IEEE Computer Security Applications Conference*, pp. 3-12, Dec. 1998.

[10] Kendall, Kris; "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", Master's Thesis, Massachusetts Institute of Technology, 1998.

[11] Heberlein, L.T.; Dias, G.V.; Levitt, K.N.; Mukherjee, B.; Wood, J.; Wolber, D.; "A Network Security Monitor", *In Proc. of the 1990 IEEE Symposium on Research in Security and Privacy*, pp. 296-304, Oakland, CA, May 1990.

[12] Heberlein, T. "Network Security Monitor (NSM) C Final Report", U.C. Davis, February 1995. http://seclab.cs.ucdavis.edu/papers/NSM-final.pdf

[13] Kemmerer, Rachard A.; "NSTAT: A Model-based Real-time Network Intrusion Detection System," Computer Science Department, University of California, Santa Barbara, Report TRCS97-18,1997.

[14] Paxon, V.; "Bro: A System for Detecting Network Intruders in Real-Time," *In Proc. of the 7th USENIX Security Symposium*, San Antonio, TX, Jan. 1998.

[15] Webster, Seth; "The Development and Analysis of Intrusion Detection Algorithms". Masters Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1998.

[16] Ko, C.; Ruschitzka, M.; Levitt, K.; "Execution Monitoring of Security-Critical Programs in a Distributed System: A Specifications-Based Approach," *In Proc. 1997 IEEE Symposium on Security and Privacy*, pp. 134-144, Oakland, CA: IEEE Computer Society Press, 1997.

[17] Popivanov, I.; Miller, R.; "Similarity Search Over Time Series Data Using Wavelets". *Proc. of the 18th International Conference on Data Engineering (ICDE'02)*, pp. 212-221, 2002.

[18] Maxion, R.; Feather, F.E.; Siewiorek, D.; "Fault Detection in an Ethernet network Using Anomaly Signature Matching",*ACM Computer Communication Review (SIGGOMM'93)*, Vol. 23(4), 1993.

[19] Platt, John; "A Resource-Allocating Network for Function Interpolation", *Neural Computation*, Vol. 3, pp. 213-225, 1991.

[20] Jiang, Jun; Papavassiliou, Symeon; "Detecting Network Attacks in the Internet via Statistical Network Traffic Normality Prediction", *Journal of Network and System Management*, Vol. 12, No. 1, pp. 51-72, Mar. 2004.

[21] Jiang, Jun; Papavassiliou, Symeon; "A Network Fault Diagnostic Approach Based on a Statistical Traffic Normality Prediction Algorithm", *In Proc. IEEE Global Communications Conference (GLOBECOM2003)*, Vol. 5 , pp. 2918-2922, Dec. 2003.

[22] Monin, Andrei S.; "Weather Forecasting as a Problem in Physics", *MIT Press*, Cambridge, MA, 1972.

[23] Dutta, M.; , Executive Editor, *Economics, Econometrics and the Links*, North Holland, 1995.

[24] Chan, Man-Chung; Fung, Chi-Chung; "Incremental Adaptation of Resource-Allocating Network for Non-stationary Time Series", *International Joint Conference on Neural Networks*, Vol. 3, pp. 1554-1559, 1999.

[25] Williams, Garnett P.; *Chaos Theory Tamed*, Taylor & Francis Limited, 1997.

[26] Duda, R.O.; Hart, P.E.; " Pattern classification and scene analysis", *Prentice-Hall*, New Jersey, 1984.

[27] Powell, M.J.D.; "Radial Basis Function for Multivariate Interpolation: A Review", in Mason, J.C.; and Cox, M.G. (Eds.): *Algorithm for approximation*, Clarendon Press, Oxford, pp. 143-168, 1987.

[28] Rape, R.; Fefer, D.; Drnovsek, J.; "Time Series Prediction with Neural Networks: A Case Study of Two Examples",*IEEE Instrumentation and Measurement Technology Conference*, May 1994.

[29] Jacobsen, Ben; "Time Series Properties of Stock Returns", *Amsterdam : Kluwer Bedrijfsinformatie*, 1997.

[30] Manikopoulos, C.; Papavassiliou, S.; "Network Intrusion and Fault Detection: A Statistical Anomaly Approach", *IEEE Communications Magazine*, Vol. 40, No. 10, pp. 76-82, October 2002.

[31] Schaal, Stefanand; Atkeson, Christopher G.; "Constructive Incremental Learning from only Local Information", *Neural Computation*, 1999.

[32] Cheung, Yiu-Ming; Leung, Wai-Man; Xu Lei; "Adaptive Rival Penalized Competitive Learning and Combined Linear Predictor Model for Financial Forecast and Investment", "International Journal of Neural Systems", Vol. 8, Nos 5&6, Oct./Dec. 1997.

[33] Mo, F.; Kinsner, W.; "Prediction and Modelling of Nonstationary Temporal Signals with Fractral characteristics",*IEEE Canadian Conference on Electrical and Computer Engineering*, Vol. 2, pp. 581-584, 1998.

[34] Fravolini, M.L.; Campa, G.; Napolitano, K.; Song, Yongkyu ; "Minimal Resource Allocating Networks for Aircraft SFDIA",*IEEE International conference on Advanced Intelligent Mechantronics Prodeedings*, 8-12 pp. 1251-1256, July 2001.

[35] http://ita.ee.lbl.gov/

[36] Shen, D.; Hellerstein, J.L., "Predictive Models for Proactive Network Management: Application to a Production Web Server", *Network Operations and Management Symposium, IEEE/IFIP* , pp. 833-846, 2000.

[37] Hellerstein, J.L.; Zhang, Fan; Shahabuddin, P.; "An Approach to Predictive Detection for Service Management, Integrated Network Management", *Proceedings of the Sixth IFIP/IEEE International Symposium on Distributed Management for the Networked Millennium*, pp. 309-322, 1999.

[38] Li, San-Qi ; Chong, Song; Hwang, Chia-Lin; "Link Capacity Allocation and Network Control by Filtered Input Rate in High Speed Networks" IEEE/ACM Trans. on Networking, Vol. 3, No. 1, pp. 10-25, Feb. 1995.

[39] Eskridge, R.E.; Ku, J.U.; Porter, P.S.; Rao, S.T.; Zurbenko, I.; "Separating Different Scales of Motion in Time Series of Meteorological Variables", *Bull. American Meteorological Society*, 78, 7, pp. 1473-1483, Jul. 1997.

[40] Zurbenko, I.G.,"The Spectral Analysis of Time Series." North Holland, pp. 241, 1986.

[41] Rao, S. T.; Zurbenko, I; "Detecting and Tracking Changes in Ozone Air Quality," *J. Air and Waste Manage. Assoc.*, 44, pp.1089-1092, 1994.

[42] Rao, S.; Zurbenko, I.; Neagu, R.,; Porter, S.; Ku, J.; Henry, R.; "Space and Time Scales in Ambient Ozone Data", *Bulletin of American Meteorological Society*, Vol. 78, No. 10, pp. 2153-2166, Oct. 1997.

[43] Chatfield, C.; "The Analysis of Time Series," *Chapman & Hall*, 1989.

[44] AKAIKE, H.; "Autoregressive Model Fitting for Control", *Annals Inst. Statist. Math.* 23, pp.163-180, 1971.

[45] Schwarz, G.; "Estimating the Dimension of a Model", *The Annals of Statistics*, Vol. 6, pp. 461-464, 1978.

[46] Lutkepohl, H.; "Comparison of Criterion for Estimating the Order of a Vector Autoregressive Process", *Journal of Royal Statistical Society*, B 43, pp. 231-239, 1985.

[47] Lutkepohl, H.; "Introduction to Multiple Time Series Analysis (2nd Edition)", *Springer Verlag*, Berlin, 1993.

[48] Son, Young Sook; "ARMA Model Identification Using the Bayes Factor", *JKSS*, Vol. 28, No. 4, pp. 503-514, 1999.

[49] Cavanaugh, J.E.; Neath, A.A.; "Generalizing the derivation of the Schwarz information criterion", Communications in Statistics Theory and Methods, 28, pp. 49-66, 1999.

[50] Neumaier, Arnold; Schneider, Tapio; "Estimation of Parameters and Eigenmodes of Multivariate Autoregressive Models", *ACM transactions on mathematical software(TOMS)*, Vol. 27, issue: 1, pp. 27-57, March, 2001.

[51] Bjorck, A; "Numerical Methods for least squares problems". *SIAM*, Philadelphia, PA, 1996.

[52] Tiao, G. C.; Box, G.E.P.; "Modeling Multiple Time Series with Applications", *J. Amer. Stat. Assoc.* 76, pp. 802-816, 1981.

[53] Brockwell, P. J.; Davis, A.; "Time Series: Theory and Methods (2nd Edition)", *Springer*, New York, 1991.

[54] Li, W.K.; Mcleod, A.I.; "Distribution of the Residual Autocorrelations in Multivariate ARMA Time Series Models", *J. Roy. Stat. Soc.* B 43, pp. 231-239, 1981.

[55] Kadirkamanathan, V.; Fallside, F.; "F-Projection: A Nonlinear Recursive Estimjation Algorithm for Neural Networks", Technical Report CUED/F-INFENG/TR.53, Cambridge University Engineering Department, 1990.

[56] Kadirkamanathan, V.; "Sequential Learning in Artificial Neural Networks", *Ph.D thesis*, Cambridge University Engineering Department, 1991.

[57] Kadirkamanathan V.; Niranjan, M.; and Fallside, F.; "Sequential Adaptation of Radial Basis Function Neural Network and Its Application to Time-series Prediction", In R.P.Lippmann, J.E.Moody and D.S. Touretzky, eds.,*Neural Information Processing System 3*, Morgan Kaufmann, 1991.

[58] Goodwin, G.C.; Sin, K.S.; "Adaptive Filtering Prediction and Control ", *McGraw-Hill*, New York, 1986.