International Scholarly Research Network
ISRN Computational Mathematics
Volume 2012, Article ID 103509, 10 pages
doi:10.5402/2012/103509

Research Article

Study of Stationary Load Increase of Computer-Network Traffic via Dynamic Principal-Component Analysis

Shengkun Xie[1,2] and Anna T. Lawniczak[2,3]

[1] *Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON, Canada M5B 2K3*
[2] *The Fields Institute for Research in Mathematical Sciences, 222 College Street, Toronto, ON, Canada M5T 3J1*
[3] *Department of Mathematics and Statistics, University of Guelph, Guelph, ON, Canada N1G 2W1*

Correspondence should be addressed to Shengkun Xie, shengkun.xie@ryerson.ca

Received 24 July 2012; Accepted 16 August 2012

Academic Editors: D. S. Corti, L. S. Heath, and R. Tuzun

Many network monitoring applications and performance analysis tools are based on the study of an aggregate measure of network traffic, for example, number of packets in transit (NPT). The simulation modeling and analysis of this type of performance indicator enables a theoretical investigation of the underlying complex system through different combination of network setups such as routing algorithms, network source loads or network topologies. To detect stationary increase of network source load, we propose a dynamic principal component analysis (PCA) method, first to extract data features and then to detect a stationary load increase. The proposed detection schemes are based on either the major or the minor principal components of network traffic data. To demonstrate the applications of the proposed method, we first applied them to some synthetic data and then to network traffic data simulated from the packet switching network (PSN) model. The proposed detection schemes, based on dynamic PCA, show enhanced performance in detecting an increase of network load for the simulated network traffic data. These results show usefulness of a new feature extraction method based on dynamic PCA that creates additional feature variables for event detection in a univariate time series.

1. Introduction

The dynamics of many complex systems such as computer networks, financial systems, transportation systems, or power systems are mathematically intractable due to their complexity ([1–3]). Better understanding of states of the complex systems and how these states change is accomplished by analyzing the data coming from the underlying complex systems [4]. In network system performance analysis, the traffic data is measured over time and statistical quality control techniques such as process control are often applied to detect whether thresholds are exceeded based on the standard deviations of observed variables. Statistical process control is the application of statistical methods such as principal component analysis (PCA) to the monitoring and control of a process to ensure that it operates at its full potential to produce conforming product. Monitoring the changes of traffic load is a practical issue to ensure that network systems are not overridden by users [5], in particular, when the load increase is stationary. We define the stationary load increase as a state of network traffic that is before the phase transition. The phase transition is due to a large increase of network source load so that the amount of network traffic appears to be increasing upward. That is, the stationary load increase results in an increase of network traffic volatility, that does not lead to an onset of network congestion immediately.

In studying network traffic performance, besides analysis of aggregate network traffic, load estimate of link traffic is another useful measure. When using this technique, the link-traffic data are sampled and analyzed to make inference from a subnetwork to a global network. The inference problem based on the study of a subdomain of the entire network system leads to an accuracy requirement problem for network traffic estimates. Some sampling techniques for traffic-load estimation are proposed in [6] as a way to

limit the measurement overhead and to meet the required accuracy. In [7], a packet-probing technique is described to detect the presence of a competing network load in a cluster environment and it distinguishes between the loads caused by network transmission and by computational operation. Our work is different from the link-traffic-load estimate. We focus instead on an aggregate measure of network traffic, the number of packets in transit (NPT), and illustrate the usefulness of the proposed statistical methods by applying them to data generated by a packet-switching network model. Study of NPT performance indicator leads to an overall control and management of network traffic, ignoring the detailed spatial packet traffic dynamics in the network. Using this aggregate measure of network performance, we aim to detect a stationary increase of traffic load in the network. This technique means identifying a small increase in a network load that leads to an increase of network traffic flow, but does not lead to an onset of congestion immediately.

Although an increase of network-source load will lead to increaseing of both the mean level and the network traffic volatilities, focusing on the volatilities is more important than focusing on the mean level because the fluctuations of network-packet traffic reflect the behaviors of the uncertainty of network performance. The traditional method of testing the increase of data variance, one of the measures of network volatility, is by F-test. However, the construction of this test statistic is based on the normal assumption and ignores the time-dependent structure of the data. Also, F-test was diagnosed as being extremely sensitive to nonnormality ([8, 9]). PCA is an another technique for analyzing the data variance. It transforms a number of variables into a number of uncorrelated principal components. Because of the uncorrelatedness of principal components, using the principal components leads to better identification of the change of variance-covariance structure. Therefore, PCA has been broadly used for monitoring link traffic of a network to detect anomalous events (e.g., [10, 11]). In such applications, the extracted principal components of a set of test data predict an anomalous event. In this paper, we apply dynamic PCA as a feature extraction method and use the PC classifier in the dynamic framework to detect the change of fluctuations of the network traffic in a feature-extracted subspace. Our approach is different from the existing ones (e.g., [10, 11]) as we analyze univariate time series data. We use a nonoverlapping moving window technique to extract a set of features from univariate network traffic data. The obtained features are treated as the observations of a multidimensional feature variable. As a result, each coordinate of the multidimensional feature variable is spatially correlated, but less autocorrelated when the size of a moving window is large. To detect the load increase, first, we extract feature information of a set of NPT-training data with a reference level of network-traffic load and then we detect the load increase of network traffic in the extracted features of a set of test data, using the proposed detection schemes based on the hypothesis testing method.

The work is a theorical investigation that focuses on analysis of simulated data, from both sythetic and a network simulator. The main contribution of this paper is the proposal of dynamic PCA coupled with nonoverlapping moving window technique for applications to data analysis of complex network systems. The paper is organized as follows: in Section 2 we provide a brief description of the network simulator, its experimental setup, and the simulated NPT data. In Section 3 we present the methodologies proposed for analyzing NPT data. Section 4 provides a justification of the appropriateness of using the proposed method to a set of synthetic data and shows the results of our application to the simulated network traffic data. Section 5 reports our conclusions and outlines the future work.

2. Packet-Switching Network Model and Simulation Data

2.1. Packet-Switching Network Model. We briefly describe the PSN model, developed in [12, 13], and its C++ simulator, called Netzwerk-1 [14] that we use in our study. The PSN model is an abstraction of the ISO OSI Network Layer Reference Model. The PSN model focuses on packets and their routings. It is scalable, distributed in space, and time discrete. It avoids the overhead of protocol details present in many PSN simulators designed with different aims in mind. In the PSN model each node performs the functions of host and router and maintains one incoming and one outgoing queue which is of unlimited length and operates according to a first-in, first-out policy. At each node, independently of the other nodes, packets are created randomly with probability λ, called source load. In the PSN model all messages are restricted to one packet carrying only the following information: time of creation, destination address, and number of hops taken.

The PSN model connection topology is represented by a weighted directed multigraph $\mathcal{L}$ where each node corresponds to a vertex and each communication link is represented by a pair of parallel edges oriented in opposite directions. To each edge is assigned a cost of packet transmission. For a given PSN model setup, all edge costs are computed using the same type of edge cost function (ecf) that is either the ecf called ONE (ONE), or QueueSize (QS), or QueueSizePlusOne (QSPO). The ecf ONE assigns a value of "one" to all edges in the lattice $\mathcal{L}$. The ecf QS assigns to each edge in the lattice $\mathcal{L}$ a value equal to the length of the outgoing queue at the node from which the edge originates. The ecf QSPO assigns a value that is the sum of a constant "one" plus the length of the outgoing queue at the node from which the edge originates. The edge costs assigned by ecf ONE do not change during a simulation run, thus this results in a static routing. Since the routing decisions made using the ecf QS or QSPO rely on the current state of the network simulation this implies adaptive or dynamic routing. In the PSN model, each packet is transmitted via routers from its source to its destination according to the routing decisions made independently at each router and based on a minimum least-cost criterion. During a simulation of the PSN model using dynamic routing packets have the ability to avoid congested nodes, they do not have this ability when the static routing is used instead.

In the PSN model, time is discrete, and we observe the network state at the discrete times $k = 0, 1, 2, \ldots, T$, where T is the final simulation time. At time $k = 0$, the setup of the PSN model is initialized with empty queues, and the routing tables are computed. The time-discrete, synchronous and spatially distributed PSN model algorithm consists of the sequence of five operations advancing the simulation time from k to $k + 1$. These operations are: (1) update routing tables, (2) create and route packets, (3) process incoming queue, (4) evaluate network state, and (5) Update simulation time. The detailed description of this algorithm is provided in [12, 13].

A PSN-model setup is defined by a selection of: a type of network connection topology, a type of ecf, a type of routing table and its update algorithm, a value of source load, seeds of two pseudorandom number generators, and a final simulation time T. The first pseudorandom number generator provides the sequence of numbers required for packets generation and routing. The second one is used for adding extra links to a regular network connection topology. The details of PSN model setup are provided in [12].

2.2. Experimental Setups of PSN Model and Network Performance Indicators. The simulation experiments were conducted for the PSN model setup with a network connection topology that is isomorphic to $\mathcal{L}^p_\square(16)$ (i.e., a two-dimensional periodic square lattice with 16 nodes in the horizontal and vertical directions), full-table routing, and distributed routing table update, and we used the default value for the second pseudorandom number generator, as we do not add extra links. During each simulation run the incoming packet traffic was generated at each network node independently of the other nodes and times by Bernoulli random variables with expected value λ, that is, source-load value. In our simulation experiments, we varied the values of the following setup variables: ecf, source load and seed of the first pseudorandom number generator. We use, respectively, the following conventions $\mathcal{L}^p_\square(16, \text{ecf})$ and $\mathcal{L}^p_\square(16, \text{ecf}, \lambda)$, where ecf = ONE, or QS, or QSPO, when we want to specify with what type of ecf and additionally with what λ value of source load the PSN model is setup.

In the PSN model, for each family of network setups, which differ only in the value of the source load λ, values of $\lambda_{\text{sub-}c}$ for which packet traffic is congestion-free are called subcritical source loads, while values $\lambda_{\text{sup-}c}$ for which traffic is congested are called supercritical source loads. The critical source load λ_c is the largest subcritical source load. Thus, λ_c is a very important network performance indicator because it is the phase transition point from free-flow to congested state of a network. Details about how we estimate the critical source load are provided in [12]. For the PSN-model setups considered here, the estimated critical source load (CSL) values are, respectively, $\lambda_c = 0.115$ for $\mathcal{L}^p_\square(16, \text{ONE})$, $\lambda_c = 0.120$ for $\mathcal{L}^p_\square(16, \text{QS})$, and $\lambda_c = 0.120$ for $\mathcal{L}^p_\square(16, \text{QSPO})$.

Another very important "real-time" network performance indicator is an indicator called number of packets in transit (NPT) ([12, 15, 16]). This indicator, $N_\nu(\text{ecf}, \lambda, k)$, for a given PSN model with $\mathcal{L}^p_\square(16, \text{ecf}, \lambda)$ setup and ν value of the seed of the first pseudorandom number generator, is given by the total number of packets in the network at time k, that is, by the sum over all network nodes of the number of packets in each outgoing queue at time k. The NPT time series, that is, $N_\nu(\text{ecf}, \lambda, k)$, for $k = 0, \ldots, T$, is an important time-dependent, that is, dynamic, aggregate measure of network performance providing information on how many packets are in the network on their routes to their destinations at time k for a given PSN-model setup $\mathcal{L}^p_\square(16, \text{ecf}, \lambda)$ and ν value of the seed of the first pseudorandom number generator. Thus, $N_\nu(\text{ecf}, \lambda, k)$ is a "real-time" network performance indicator. We simulate the NPT-time series of PSN model with $\mathcal{L}^p_\square(16, \text{ecf}, \lambda)$ setups, respectively, for ecf = ONE, QS, and QSPO, and source load values $\lambda = 0.095, 0.100, 0.105$, and 0.110, called FreeFlow values as these values are smaller than the respective critical source-load values. For each PSN model with the setup $\mathcal{L}^p_\square(16, \text{ecf}, \lambda)$, where ecf = ONE, QS, QSPO, respectively, and λ belongs to FreeFlow set we run simulations with 24 different seed values ν, where $\nu = 1, \ldots, 24$, of the first pseudorandom number generator. Each simulation is run until the final simulation time $T = 8000$ time steps. Even though the final simulation time is $T = 8000$ only the data from $k = 2001$ is accounted for in our analysis in order to remove the initial transient effects caused by the setups of the PSN model that are always with empty queues. We denote this initial value as T_0. Thus, notice, in all the presented graphs time-axis scale goes always from 0 to 6000 to account for the discarded data.

2.3. Simulated NPT Data. The behaviors of the time variability of NPT data are the key characteristics of NPT data ([15, 16]). The time variability of the NPT data shifted by its time average, is denoted by

$$\overline{N}_\nu(\text{ecf}, \lambda, k) = N_\nu(\text{ecf}, \lambda, k) - \overline{N}_\nu(\text{ecf}, \lambda), \tag{1}$$

where

$$\overline{N}_\nu(\text{ecf}, \lambda) = \frac{1}{T - T_0} \sum_k N_\nu(\text{ecf}, \lambda, k) \tag{2}$$

is its time average. The volatility of NPT data increases with the increase of source-load value for each ecf ONE, QS, and QSPO. However, from our empirical studies [17], the changes of volatilities for ecf QS and QSPO are difficult to distinguish. To detect an increase of network source load, for each type of ecf, the simulated NPT data is categorized into two groups, normal traffic and normal-high traffic. By normal traffic we mean a traffic such that the NPT data has the same value of source load as the network-traffic training data or a value smaller than the one that the network-traffic training data has. Normal-high traffic means traffic such that the NPT data correspond to a source-load value larger than the one that the network-traffic training data has. To detect an increase of the network-source load, we choose the NPT data simulated using the model setup $\mathcal{L}^p_\square(16, \text{ONE}, 0.100)$, $\mathcal{L}^p_\square(16, \text{QS}, 0.100)$, and $\mathcal{L}^p_\square(16, \text{QSPO}, 0.100)$, respectively, to be the training data of the considered network type. The NPT data simulated using the model

setup $\mathcal{L}_{\square}^{p}(16, \text{ecf}, 0.095)$ for each ecf = ONE, QS, and QSPO, respectively, are treated as the normal-traffic test data. The network-traffic data simulated using the model setup $\mathcal{L}_{\square}^{p}(16, \text{ecf}, 0.105)$ and $\mathcal{L}_{\square}^{p}(16, \text{ecf}, 0.110)$, respectively, for each ecf = ONE, QS, and QSPO, are treated as the normal-high traffic test data.

3. Methodology

3.1. Principal Component Extraction by Dynamic PCA. Extraction of additional time-dependent variables from time series data was originally accomplished by introducing dynamic PCA ([18, 19]). The method considers observations, taken at times $1, 2, \ldots, n$, that is, $\{\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)\} \subseteq R^p$ from a p-variate time series $\mathbf{x}(k) = [x_1(k), x_2(k), \ldots, x_p(k)]^\top$, where n is the number of observations and $l + 1 \leq k \leq n$. In the present work, $p = 1$. Using dynamic PCA, the input data matrix $\mathbf{X}$ to be analyzed is arranged as follows:

$$\mathbf{X} = \left[\mathbf{x}(k-l)^\top, \mathbf{x}(k-l+1)^\top, \ldots, \mathbf{x}(k)^\top\right], \tag{3}$$

where l is the time lag that is used for capturing the dynamics of time series. By doing eigenvalues analysis, dynamic PCA aim to determine a suitable time lag l for the purpose of modeling stochastic process $\mathbf{x}(k)$. Instead of focusing on the analysis of the eigenvalues of the covariance matrix of the input data matrix (3), our goal is to apply the dynamic PCA method to extract additional feature variables from univariate NPT data. For this reason, we do not need to determine the optimal value of l. What we need is the reasonable large value of l so that we can treat each window as a realization of the object of interest, that is, multivariate data. In this case, l is referred to as the length of window. Because of this, we turn analysis of one-dimensional data into a problem that focuses on analysis of multivariate data, by defining each element of the window as a time-depdendent feature variable. The extension of feature variables makes a multivariate method applicable to one-dimensional time-series data.

In multivariate analysis, ideally, observations of underlying multivariate should be collected independently. In the network-traffic-monitoring problem, NPT data is collected over time. This implies that NPT data is correlated if the a length of window is designed to be a small value. Also, in the data matrix presented in (3), the observations are highly series correlated so that further analysis is affected. In order to potentially improve the performance of using dynamic PCA, we propose a method of applying a nonoverlapping moving window technique. This method decreases correlation of each extracted time-dependent feature variable in the window when the width of the moving window is large.

When the above discussed technique is applied to the NPT data we denote each of the simulated paths of NPT training data with n observations by $N_v(k)$. Recall that for each ecf = ONE, QS, and QSPO, respectively, the $\overline{N}_v(\text{ecf}, \lambda, k)$ denotes the NPT data shifted by its time average $\overline{N}_v(\text{ecf}, \lambda)$, where λ is the source load value of NPT training data. In what follows when confusion does arise we will use for $\overline{N}_v(\text{ecf}, \lambda, k)$ a shorter notation $\overline{N}_v(k)$. Applying the nonoverlapping moving window technique to $\overline{N}_v(k)$, the input data matrix becomes

$$D^{\overline{N}_v} = \begin{pmatrix} \overline{N}_v(1) & \overline{N}_v(2) & \cdots & \overline{N}_v(l) \\ \overline{N}_v(l+1) & \overline{N}_v(l+2) & \cdots & \overline{N}_v(2l) \\ \vdots & \vdots & \vdots & \vdots \\ \overline{N}_v(ml-l+1) & \overline{N}_v(ml-l+2) & \cdots & \overline{N}_v(ml) \end{pmatrix}, \tag{4}$$

where $n = ml$, m is the total number of the moving windows of $\mathbf{x}(k)$ each with length l. The benefit of applying the nonoverlapping moving window data segmentation technique to the NPT data is that, for a large value of l, the sequence of data $[\overline{N}_v(k), \overline{N}_v(k+l), \ldots, \overline{N}_v(k+ml-l)]^\top$, for $k = 1, \ldots l$, becomes uncorrelated. Let the total number of simulated paths of NPT training data be R, for each type of ecf. The simulated paths are independent simulations, with different chosen random seeds from 1 to R. The data matrix constructed from these R simulated paths becomes $D^{\overline{N}} = [D^{\overline{N}_1{}^\top}, D^{\overline{N}_2{}^\top}, \ldots, D^{\overline{N}_R{}^\top}]^\top$, with the size $mR \times l$. PCA is then applied to map $D^{\overline{N}}$ into a new feature space and possibly to reduce the number of feature dimensions to enable a high-performance detection for the PC classifier. The PCA of the training data matrix $D^{\overline{N}}$ then yields the standardized eigenvectors V_i, for $1 \leq i \leq l$.

To perform the feature extraction of NPT test data with a length of $n = ml$ (where n, m, and l are defined as before), we organize each of NPT test data into a column vector, denoted by $\mathbf{Y}^s = [y^s(1), y^s(2), \ldots, y^s(n)]^\top$, where s represents each of the simulated paths in the test data set. For each ecf, $\mathbf{Y}^s$ refers to the NPT data shifted by its time average, that is, $\overline{N}_s(\text{ecf}, \lambda, k)$, for each ecf = ONE, QS, and QSPO, respectively, where λ is the source-load value of NPT test data. We first partition $\mathbf{Y}^s$ into m windows each of the length l, that is, $\mathbf{Y}^s = [\mathbf{y}^s(1), \mathbf{y}^s(2), \ldots, \mathbf{y}^s(m)]^\top$, where $\mathbf{y}^s(k^*) = [y_1^s(k^*), y_2^s(k^*), \ldots, y_l^s(k^*)]$ is the k^*th window of $\mathbf{Y}^s$, for $k^* = 1, 2, \ldots, m$. The objective of feature extraction by PCA is to project each nonoverlapping moving window of the network traffic test data $\mathbf{y}^s(k^*) = [y_1^s(k^*), y_2^s(k^*), \ldots, y_l^s(k^*)]$ onto the normalized eigenvectors V_i, for $1 \leq i \leq l$, of the matrix (4).

3.2. Detection Schemes. If the variance-covariance structure of the extracted feature variables changes, in particular, when the variability of some of the feature variables is increased, the projections of new observations will significantly change as the dominant feature variables change. The PC classifier, which has been used successfully in anomalous event detection of network traffic (e.g., [10, 11]), can be used in order to detect such change. In the anomalous event detection, PCA was applied to multivariate network traffic data to detect the existence of the anomalous events caused by a significant change of variance-covariance structure of the network traffic data. Our work extends the PC classifier to the dynamic PCA framework and enables the application of this extended PCA to univariate time series data to detect the increase of network-source load. In our detection schemes,

the PC classifier consists of two functions of extracted PC scores of each test NPT data s as follows:

$$\hat{f}_1^s(k^*) = \sum_{i=1}^{l_1} \hat{\mathbf{y}}_i^s(k^*)^2, \qquad \hat{f}_2^s(k^*) = \sum_{i=l_2-r+1}^{l_2} \hat{\mathbf{y}}_i^s(k^*)^2, \tag{5}$$

where k^* is the index of the moving window of the NPT-test data and $k^* = 1, 2, \ldots, m$. The m is the total number of windows of each NPT-test data. The l_1 and r, respectively, are the number of major PCs and minor PCs selected, and they are referred to as feature dimension in the later discussion. l_2 is the number of total PCs retained from the feature extraction by PCA. The maximum number allowed for l_2 is equal to l, however, because data often contains noise, l_2 is usually assigned a smaller value than l. In the case, we treat the components that are corresponding to smaller eigenvalues to be noise components and ignore them in further analysis. In this paper, major PCs mean the first few PCs in the retained PCs and minor PCs correspond to the last few PCs in the retained PCs. When the increase of network-load leads to a significant increase of both variance and covariance of the selected feature variables, this increase of network load is then detectable by major PCs. Because large values for minor PCs imply a violation of the correlation structure of the feature variables, the network-load increase is then detectable when the increase of load leads to a significant change of correlation structure of the feature variables [20].

3.2.1. Detection Scheme by Single Hypothesis. The single hypothesis detection scheme has two independent hypotheses. One uses only major PCs and another one uses only minor PCs for the purpose of detection. The first detection scheme is based on the following null and alternative hypothesis:

$$H_0^{\text{major}} : f_1^s(k^*) \le f_0^{\text{major}}, \qquad H_A^{\text{major}} : f_1^s(k^*) > f_0^{\text{major}}. \tag{6}$$

The test statistics for this detection scheme are $\hat{f}_1^s(k^*)$. If the hypothesis testing rejects the null hypothesis H_0^{major}, then the test data is classified into the normal-high group; if it accepts, then it is classified into the normal group. The second detection scheme is based on the following null and alternative hypothesis:

$$H_0^{\text{minor}} : f_2^s(k^*) \le f_0^{\text{minor}}, \qquad H_A^{\text{minor}} : f_2^s(k^*) > f_0^{\text{minor}}. \tag{7}$$

The test statistics for this detection scheme are $\hat{f}_2^s(k^*)$. Similarly, if the hypothesis testing rejects H_0^{minor}, then the test data is classified into the normal-high group; otherwise it is classified into the normal group. In each of the detection schemes, the significance level of the hypothesis testing has to be specified first. This specified significance level is then used to determine the critical values: $\hat{f}_0^{\text{major}}$ and $\hat{f}_0^{\text{minor}}$ as estimates of f_0^{major} and f_0^{minor}, respectively. The normality of the extracted features is usually unlikely to be satisfied because of the high level of time variability of NPT data; therefore, we do not use the normal percentile, but calculate the percentile of the empirical cumulative distribution functions $f_1^s(k^*)$ and $f_2^s(k^*)$ as the critical values of f_0^{major} and f_0^{minor}, respectively.

3.2.2. Detection Scheme by Multiple Hypothesis. While the detection scheme based on major PCs detects the change of variance and covariance structure of multivariate data, the detection scheme based on minor PCs detects the change of correlation structure of multivariate data. If the increase of network source load leads to both changes, that is, of the variance-covariance structure and the correlation structure of NPT data, a combined method using both major PCs and minor PCs can be applied to increase the detection rates. This combined detection scheme is based on the following null hypothesis and alternative hypothesis:

$$\begin{aligned} &H_0^{\text{combined}} : f_1^s(k^*) \le f_0^{\text{major}}, \qquad f_2^s(k^*) \le f_0^{\text{minor}}, \\ &H_A^{\text{combined}} : f_1^s(k^*) > f_0^{\text{major}}, \quad \text{or} \quad f_2^s(k^*) > f_0^{\text{minor}}. \end{aligned} \tag{8}$$

If either $\hat{f}_1^s(k^*)$ or $\hat{f}_2^s(k^*)$ is significant, then the test data is classified into the normal-high group; otherwise it is classified into the normal group. The constructions of $\hat{f}_1^s(k^*)$ and $\hat{f}_2^s(k^*)$ are based on major PCs and minor PCs, respectively. These two test statistics are statistically independent. The performance of the detection of the load increase may depend on the choice of detection scheme as different schemes detect different types of change of variance-covariance structure and correlation structure.

3.3. Detection Performance Measures. In order to evaluate the performance of detection the rejection percentages of the test are used as a detection rate and the performance of detection is given as follows:

$$d = \frac{T_1}{T^*}, \tag{9}$$

where T_1 is the total number of rejections of hypothesis testing among a set of the NPT-test data series and T^* is the total number of the NPT-test data. In the major PCs detection scheme, given that NPT-test data series s is from a normal traffic group, the detection rate is the misclassification rate or false detection rate, denoted by d_1^{major}. When a series of test data s is from a normal-high traffic group, the detection rate is the probability of detecting normal-high traffic, denoted by d_2^{major}. Similarly, d_1^{minor} is the misclassification rate when the test traffic is normal and the probability of the presence of normal-high traffic is denoted by d_2^{minor}. The misclassification rate for the combined scheme when test traffic is normal is denoted by d_1^{combined}, and the probability of detecting normal-high traffic is denoted by d_2^{combined}. The values of d_1^{major}, d_1^{minor}, and d_1^{combined} are the estimates of the occurrence of type I errors. The values of d_2^{major}, d_2^{minor}, and d_2^{combined} are the estimates of the power of the respective tests.

The detection rate depends on the selection of l_1 and r, which are the sizes of major PCs and minor PCs used for data

classification. A satisfactory result of the calculated detection rate may be obtained by investigating the relationship between the detection rate and the feature dimension l_1 or r.

4. Results

4.1. Synthetic Data. In order to demonstrate the application of dynamic PCA as a feature extraction method, we first apply this method to a set of synthetic univariate stationary time-series data. Using the test data, we are trying to detect an increase of data variance by the scheme using major PCs, the scheme using minor PCs, or the combined scheme. The following stationary AR(1) model is used to generate data:

$$x_t = \phi_1 x_{t-1} + \omega_t, \tag{10}$$

where $|\phi_1| < 1$ and ω_t is a Gaussian white noise with mean zero and variance one. For the simulations, we choose three values of $\phi_1 = 0.6$, $\phi_1 = -0.7$, and $\phi_1 = 0.7$. Because the theoretical variance of x_t is $1/(1-\phi_1^2)$ [21], the theoretical variance of x_t in the AR(1) model with $\phi_1 = 0.6$ is equal to 1.5625, and the theoretical variance of x_t with $\phi_1 = 0.7$ or -0.7 is 1.9608. An increase of variance of the AR(1) time series when $\phi_1 = 0.6$ changes to $\phi_1 = 0.7$ or -0.7, results in the AR(1) model with $\phi_1 = 0.6$ being selected as the simulation model of the normal type, and the AR(1) models with $\phi_1 = -0.7$ and $\phi_1 = 0.7$ being treated as other two models for simulating test data. We simulate two time series of the normal type, using $\phi_1 = 0.6$. One is assigned to training data set and another one becomes the test data of the normal type. In addition, two test time series are simulated, using $\phi_1 = -0.7$ and $\phi_1 = 0.7$, respectively. The lengths of all the simulated data are equal to 10,000.

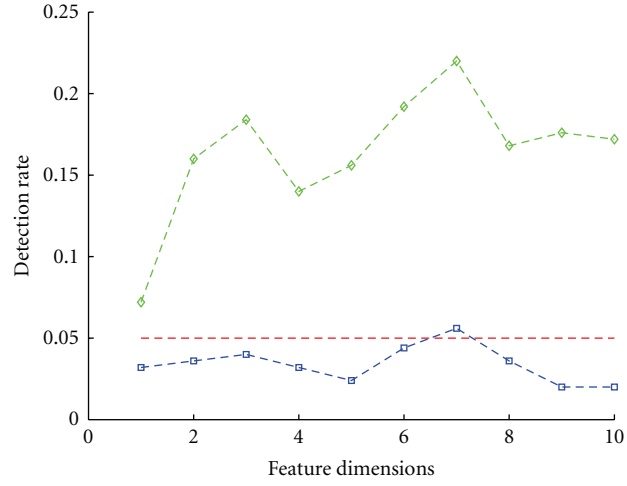
In this experiment, the width of the nonoverlapping moving window is set to be $l = 40$ (i.e., it is determined by the significant time lag of autocorrelation function plots of the data). The detection results using the discussed simulated data are reported in Figure 1. In Figure 1, the increase of variance of test data with $\phi_1 = 0.7$ shows that it is detectable by major PCs (Figure 1(a)), but it is not detectable by minor PCs (Figure 1(b)). The marked change in the correlation matrix causes an increase of variance to be detectable by minor PCs (Figure 1(d)), but it is not detectable by major PCs (as shown in Figure 1(c)). The performance of using minor PCs shown in Figure 1(d) is not extremely satisfactory for most of the retained feature dimensions, but the result is acceptable when r is 3. In this case, the detection rate d_1^{minor} is slightly higher than the predefined 5% type I error rate, and the values of d_2^{minor} are much larger than the predefined type I error rate.

Figures 1(e) and 1(f) show that the increase of data variance is detectable for feature dimension $l_1 \leq 5$ and $r \leq 5$, but the performance of the detection is dropped when $l \geq 6$ and $r \geq 6$. The l_2 in (5) is set to be 20 for the results shown in Figure 1, as the first 20 PCs explain about 86% of the total variation of the training data. In this simulation experiment, we have demonstrated that the dynamic PCA and its detection schemes can successfully capture the increase of data variance in data simulated from an AR(1) model with various model parameters. In particular, the combined detection scheme promises increased precision of detection.

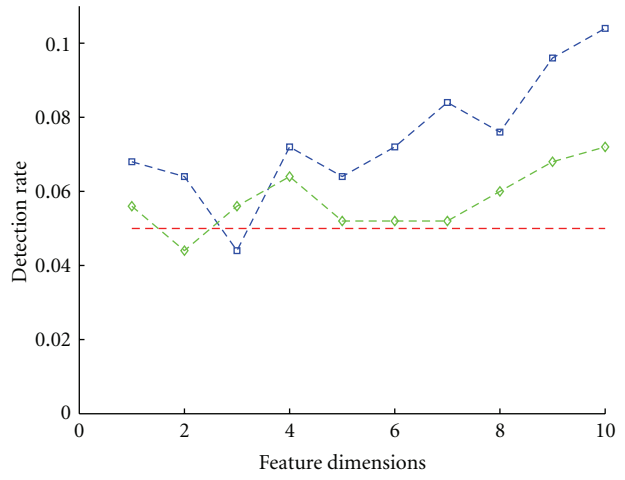
4.2. Network Traffic Data. The dynamic PCA method and its detection schemes are applied to NPT data associated with different source loads and routing algorithms to detect the load increase for each ecf. Because of the dimension-reduction property of PCA, l_2 corresponding to the dimension of subfeature space is often far smaller than the total number of originally selected feature variables. The nonoverlapping moving window size l from the modified dynamic PCA is set as $l = 100$ and $l_2 = 20$, for all types of the ecfs. The first 20 PCs explain about 95% of total variations of the training data of each type. The threshold values of f_0^{major} and f_0^{minor} used for load increase detections are determined by the 95th percentiles of the empirical cumulative distribution functions of $\widehat{f}_1^s(k)$ and $\widehat{f}_2^s(k)$, respectively, where $1 \leq k \leq 24 \times 60$. Figure 2 shows the results of detection rate d for a single-hypothesis-detection scheme, using either major PCs or minor PCs of different sizes of selected major PCs or selected minor PCs.

Figures 2(a) and 2(b) display the results based on the single-hypothesis-detection schemes using major PCs and minor PCs, for $\lambda = 0.095, 0.100$, respectively. The detection rate d is calculated using a 5% type I error (i.e., 5% significant level of hypothesis testing) for each hypothesis testing of the PC scores. The feature dimension parameter is l_1 or r, depending on the choice of detection method and varies from 1 to 10. In the case of source load $\lambda = 0.095$, the detection rate d is smaller than 5% for the detection schemes using a smaller number of major PCs and for the detection schemes using marginal minor PCs for all types of the ecfs, suggesting that the proposed methods successfully prevent a high false alarm when the network traffic source load is lower than the source load of the training data. For the test data with source load $\lambda = 0.100$, and for some predefined type I error rates, the single-hypothesis-detection schemes fail to accept the null hypothesis. However, the calculated type I error rate d_1^{major} or d_1^{minor} is only slightly larger than the type I error. The NPT training data and the test data were generated using the same network setup, and these NPT data have high local-time variability.

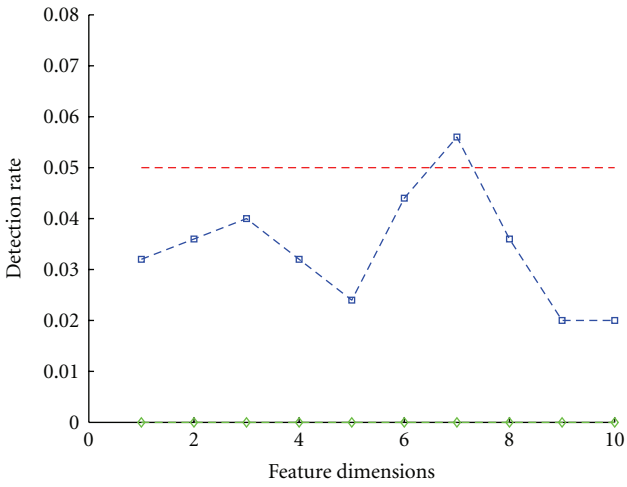
For the detection of the load increase, the modified dynamic PCA method is highly successful with a large value of power, even for a small increase of source load, that is, for a normal-high traffic with source load $\lambda = 0.105$. Figure 2(c) shows the results of using major PCs and Figure 2(d) displays the results of using minor PCs. The feature dimensions that correspond to l_1 and r in (5) vary from 1 to 10 in both cases. The detection rates of d_2^{major} and d_2^{minor} are all larger than the predefined type I error rate. The only exception is the case of the test data coming from the PSN model with setup $\mathcal{L}_{\square}^{p}(16, \text{ONE}, 0.105)$, where a detection scheme using a value smaller than $r = 3$ of feature dimension r is applied. The obtained results suggest that the proposed methods are very promising



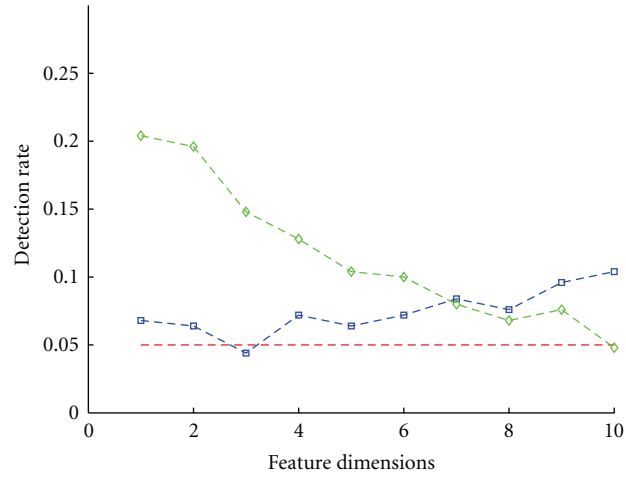
(a) d_1^{major} and d_2^{major} for the test data with $\phi_1 = 0.6$ and $\phi_1 = 0.7$, respectively



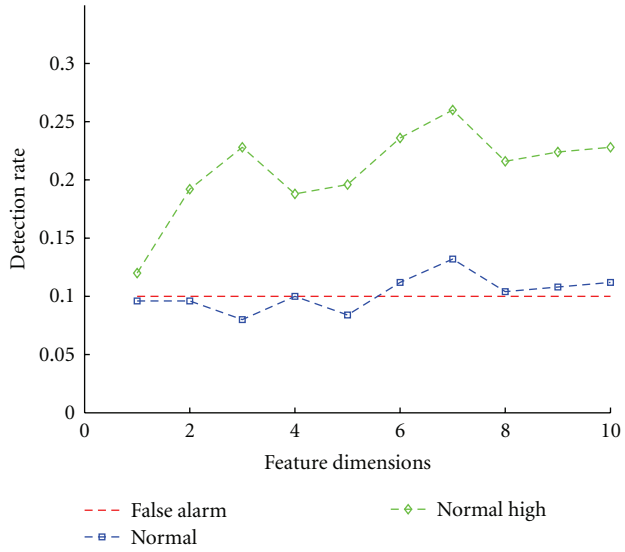
(b) d_1^{minor} and d_2^{minor} for the test data with $\phi_1 = 0.6$ and $\phi_1 = 0.7$, respectively



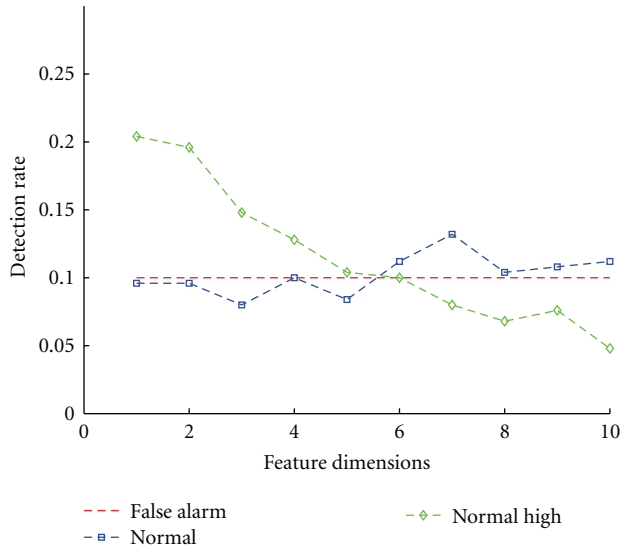
(c) d_1^{major} and d_2^{major} for the test data with $\phi_1 = 0.6$ and $\phi_1 = -0.7$, respectively



(d) d_1^{minor} and d_2^{minor} for the test data with $\phi_1 = 0.6$ and $\phi_1 = -0.7$, respectively

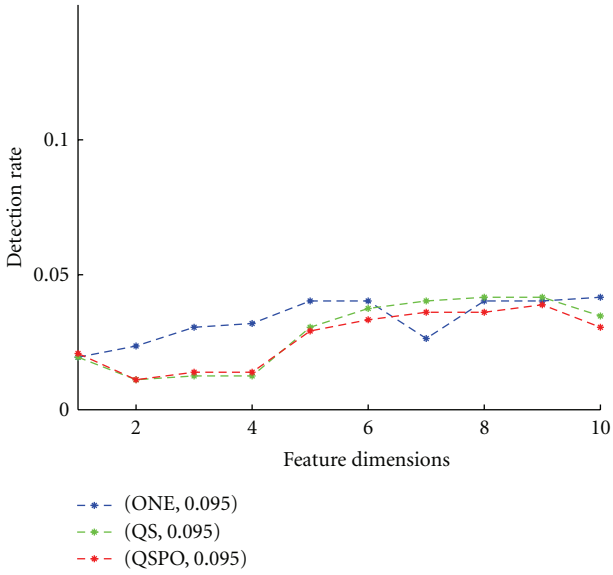


(e) d_1^{combined} and d_2^{combined} for the test data with $\phi_1 = 0.6$ and $\phi_1 = 0.7$, respectively

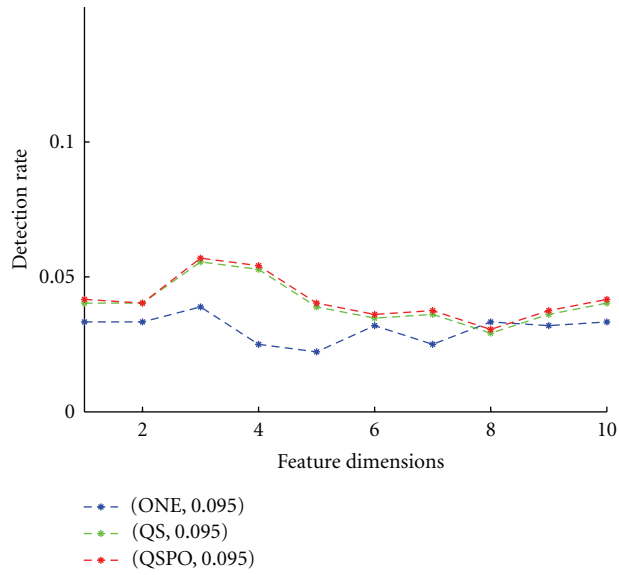


(f) d_1^{combined} and d_2^{combined} for the test data with $\phi_1 = 0.6$ and $\phi_1 = -0.7$, respectively

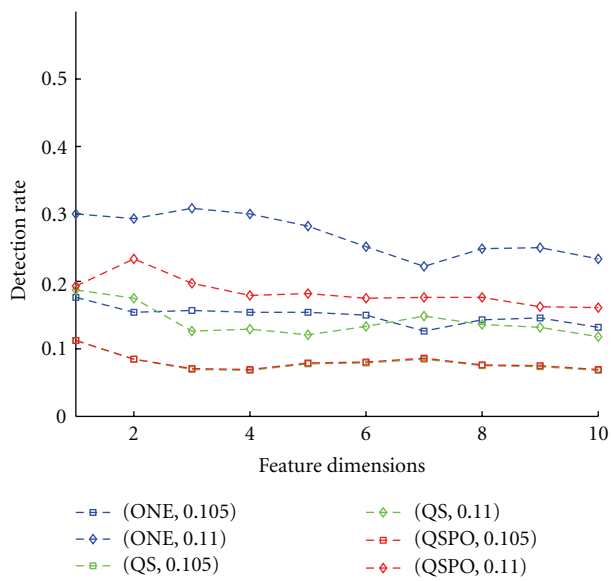
FIGURE 1: Detection performance for the scheme using major PCs, minor PCs, and the combined scheme. The line in red corresponds to the significant level.



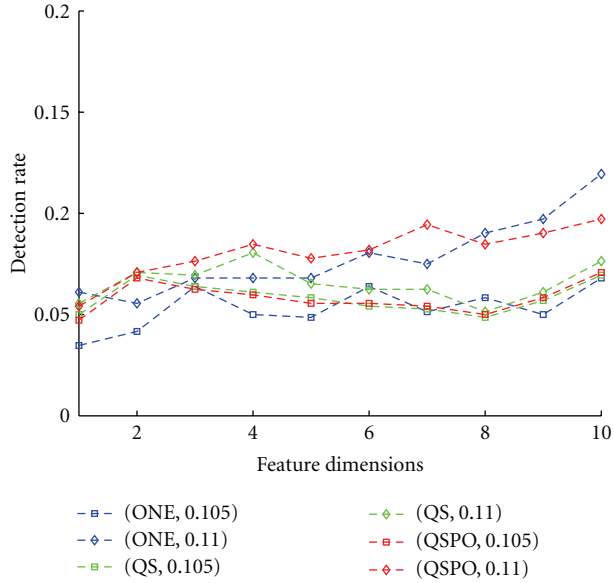
(a) Detection rate (the estimate of d_1) for normal traffic using major PCs



(b) Detection rate (the estimate of d_1) for normal traffic using minor PCs



(c) Detection rate (the estimate of d_2) for the load increase of traffic using major PCs



(d) Detection rate (the estimate of d_2) for the load increase of traffic using minor PCs

FIGURE 2: Detection rate with different numbers of features used for detection under the 95% confidence level. The threshold is calculated using the 95th percentile of the empirical cumulative distribution of the extracted features.

in the detection of network-load increase for all types of the ecfs when major PCs are used for detection. This successful detection indicates a major change in variance-covariance structure when the network-traffic load goes from a normal level to a normal-high level.

The detection scheme using major PCs performs best in detecting a load increase of a network traffic. The detection scheme based on minor PCs performs well for the test data with an increase of the load, but it gives a larger type I error rate than the specified ones when the test data are part of normal traffic. The combined detection scheme performs better than the detection scheme with minor PCs, not only successfully detecting an increase of network load, but also performing well in preventing false alarms for the test data from normal traffic.

5. Conclusions and Future Work

In this paper, we examined new network load increase detection schemes based on a modified dynamic PCA approach and on parts of extracted features acting as

a classifier to detect the load increase of a set of univariate NPT data. The initial testing used a set of simulation data from stationary AR(1) models. The 95th percentile of the empirical cumulative distribution function of the extracted features was calculated as the threshold value for classification and the feature variables of the test data were extended according to the number of feature variables of the training data. After being projected onto the feature space obtained from the training data of the test data, the test statistics of hypothesis testing were calculated and then compared to the threshold value to enable a decision of load increase at each time k. The final decision of detection of load increase is based on the relative ratio of the number of successful detections to the total number of detections. This rate specifies the probability of PC scores of the NPT-test data over the threshold value. The proposed detection schemes show enhanced performance for the detection of load increase; in particular, the detection scheme that uses only the first PC. These detection schemes prevent false alarms when the test data show normal traffic because the method differentiates normal network traffic from normal-high network traffic.

However, the difficulty of applying this linear method when dealing with high local-time variability needs a solution. Extending this method to a kernel-based method for NPT data may be promising. Improvement of analysis and detection performance using kernel-based detection methods could explain potential nonlinearity within the extended feature variables. The proposed detection methods, tested on the offline simulation data, can also be applied to an online detection problem. Extending our current work to an online-load-increase detection problem would facilitate detecting normal-high network traffic instantaneously.

Acknowledgments

This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET: http://www.sharcnet.ca/). The authors acknowledge the prior work of A. T. Lawniczak with A. Gerisch, B. N. Di Stefano, X. Tang, and J. Xu. A. T. Lawniczak acknowledges partial financial support from SHARCNET and NSERC of Canada. S. Xie acknowledges the financial support from MITACS and Ryerson University, under MITACS Elevate Strategic Post doctoral Award. The authors acknowledge use of simulation data produced by J. Xu as part of the fulfilment of a SHARCNET grant of A. T. Lawniczak. The authors thank The Fields Institute for Research in Mathematical Sciences for the hospitality while conducting this research and B. Allen and Y. Sun for helpful comments.

References

[1] T. Sheldon, *Encyclopedia of Networking & Telecommunications*, Osborne/McGraw-Hill, Berkeley, Calif, USA, 2001.

[2] A. Y. Tretyakov, H. Takayasu, and M. Takayasu, "Phase transition in a computer network model," *Physica A*, vol. 253, no. 1–4, pp. 315–322, 1998.

[3] L. Kocarev and G. Vattay, Eds., *Complex Dynamics in Communication Networks*, Springer, New York, NY, USA, 2005.

[4] J. Wang, Z. Ma, and L. Li, "Detection, mining and forecasting of impact load in power load forecasting," *Applied Mathematics and Computation*, vol. 168, no. 1, pp. 29–39, 2005.

[5] F. Mata, J. Aracil, and J. L. Garcia-Dorado, "Automated detection of load changes in large-scale networks," in *TMA 2009*, M. Papadopouli, P. Owezarski, and A. Pras, Eds., vol. 5537 of *Lecture Notes in Computer Science*, p. 3441, Springer, Berlin, Germany.

[6] B.-Y. Choi, J. Park, and Z.-L. Zhang, "Adaptive random sampling for load change detection," *Performance Evaluation Review*, vol. 30, no. 1, pp. 272–273, 2002.

[7] S. Storie and M. Sosonkina, "Packet probing as network load detection for scientific applications at run-time," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, pp. 871–880, April 2004.

[8] G. E. P. Box, "Non-normality and tests on variances," *Biometrika*, vol. 40, no. 3/4, Article ID 318335.

[9] C. A. Markowski and E. P. Markowski, "Conditions for the effectiveness of a preliminary test of variance," *The American Statistician*, vol. 44, no. 4, Article ID 322326, 1990.

[10] R. Kwitt and U. Hofmann, "Unsupervised anomaly detection in network traffic by means of robust PCA," in *Proceedings of the International Multi-Conference on Computing in the Global Information Technology (ICCGI '07)*, 2007.

[11] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM '03)*, pp. 172–179, Melbourne, Fla, USA, November 2003.

[12] A. T. Lawniczak, A. Gerisch, and B. Di Stefano, "OSI network-layer abstraction: analysis of simulation dynamics and performance indicators, science of complex networks," in *Science of Complex Networks: From Biology to the Internet and WWW (CNET '04)*, J. F. Mendes, Ed., vol. 776 of *AIP Conference Proceedings*, pp. 166–200, 2005.

[13] A. Gerisch, A. T. Lawniczak, and B. Di Stefano, "Building blocks of a simulation environment of the OSI network layer of packet switching networks," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering: Toward a Caring and Humane Technology (CCECE '03)*, pp. 1067–1070, Montreal, Canada, May 2003.

[14] A. Gerisch, A. T. Lawniczak, and B. Di Stefano, "Building blocks of a simulation environment of the OSI network layer of packet-switching networks," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering: Toward a Caring and Humane Technology (CCECE '03)*, pp. 1067–1070, Montreal, Canada, May 2003.

[15] A. T. Lawniczak and S. Xie, "Impact of source load and routing on QoS of packets delivery," *Journal of Computational Science*, vol. 1, no. 2, pp. 121–129, 2010.

[16] A. T. Lawniczak and S. Xie, "Number of packets in transit as a function of source load and routing," in *Proceedings of the 10th International Conference on Computational Science (ICCS '10)*, pp. 2363–2370, June 2010.

[17] S. Xie and A. T. Lawniczak, "Detection of stationary network load increase using univariate network aggregate traffic data by dynamic PCA," in *IEEE Symposium Series on Computational Intelligence*, Paris, France, April 2011.

[18] W. Ku, R. H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 30, no. 1, pp. 179–196, 1995.

[19] F. Tsung, "Statistical monitoring and diagnosis of automatic controlled processes using dynamic PCA," *International Journal of Production Research*, vol. 38, no. 3, pp. 625–637, 2000.
[20] I. T. Jolliffe, *Principal Component Analysis*, Springer Science, New York,, 2004.
[21] R. H. Shumway and D. S. Stoer, *Time Series Analysis and Its Applications With R Examples*, Springer Science, New York, NY, USA, 2005.