Covariance-Matrix Modeling and Detecting Various Flooding Attacks

Daniel S. Yeung, *Fellow, IEEE*, Shuyuan Jin, *Member, IEEE*, and Xizhao Wang, *Senior Member, IEEE*

***Abstract*—This paper presents a covariance-matrix modeling and detection approach to detecting various flooding attacks. Based on the investigation of correlativity changes of monitored network features during flooding attacks, this paper employs statistical covariance matrices to build a norm profile of normal activities in information systems and directly utilizes the changes of covariance matrices to detect various flooding attacks. The classification boundary is constrained by a threshold matrix, where each element evaluates the degree to which an observed covariance matrix is different from the norm profile in terms of the changes of correlation between the monitored network features represented by this element. Based on Chebyshev inequality theory, we give a practical (heuristic) approach to determining the threshold matrix. Furthermore, the result matrix obtained in the detection serves as the second-order features to characterize the detected flooding attack. The performance of the approach is examined by detecting Neptune and Smurf attacks—two common distributed Denial-of-Service flooding attacks. The evaluation results show that the detection approach can accurately differentiate the flooding attacks from the normal traffic. Moreover, we demonstrate that the system extracts a stable set of the second-order features for these two flooding attacks.**

***Index Terms*—Covariance matrix, flooding attacks, second-order feature, statistical anomaly detection, threshold matrix.**

I. Introduction

FLOODING attacks have imposed significant threat to the reliability of computer networks. These flooding attacks are defined as the attacks which exploit the huge resource asymmetry between the Internet and victim and impose abnormal exhaustion of either Internet bandwidth or the server's resource (e.g., memory and CPU resources) [1]. The consequences caused by flooding attacks are very severe. For example, a typical flooding attack can prevent network users from accessing critical services or even cause the failures of networking infrastructure. To compromise the security of an information system, various flooding attacks can take many different means. For instance, a quick probing attack [33] occupies network bandwidth and gathers host vulnerabilities by scanning a network of computers within a short period; a flooding Distributed Denial-of-Service (DDoS) attack makes a host or network service unavailable by amassing a number of compromised hosts to send useless packets to the victim at around the same time [1]; a flooding spam worm exhausts network bandwidth and server's memory by mass mailing within a short time.

To assure the reliability of computer networks, an effective detection for flooding attacks is indispensable. Flooding-attack detection belongs to the field of intrusion detection. In the past decade, a variety of studies on intrusion detection emerged in the literature. They vary in their approaches and addressed intrusions. Largely, different approaches fall into two major categories: misuse detection and anomaly detection. Misuse-detection techniques signal intrusions when the observed activities in an information system match the prebuilt rules or signatures of known intrusions. Anomaly detection techniques indicate intrusions when the subject's observed behaviors have a significant deviation from its norm profile. Compared with the techniques of anomaly detection, misuse detection techniques lack the ability to identify unknown intrusions [2]. When the attack signature changes a little, the original built-up detection rules in the misuse-detection system will have no use. For example, Xu [14] shows that a defense technique proposed by Internet Security System (ISS) is very effective in countering current DDoS software. But, it becomes powerless when such software is slightly modified. Since various unknown flooding attacks pop up at a surprising rate and become prevalent in network attack incidences [20], [34], anomaly detection tools are often employed in the detection of multiple and various flooding attacks.

In most of the anomaly detection approaches to detecting flooding attacks, large amounts of attention are mainly paid to forming the criteria among the normal traffic and known attacks, while the differences among various unknown attacks are neglected. For example, Lee and Stolfo [13] employ machine-learning algorithms to generate intrusion-detection rules based on the analysis of network-connection records, in which various types of flooding attacks in Denial-of-Service (DoS) and Probes categories are signaled when the observed activities match the rules of known attacks. However, since the rules cannot cover the high variance of the unknown attacks, their model is not effective in detecting the unknown DoS attacks [15]. Wang *et al.* [12] utilize the change-point-monitoring approach to detect flooding DoS attacks. Their detection model is effective in making a quick detection of any abrupt change in the network traffic. However, the model lacks the ability to identify the types of the detected anomalies—such as either Smurf or Neptune attack in the category of flooding DoS attacks. To

Manuscript received April 30, 2005; revised October 16, 2006. This work was supported by the Hong Kong Research Grants Council under Grant B-Q571. This paper was recommended by Guest Editor H. Pham.

D. S. Yeung and S. Jin are with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: csdaniel@comp.polyu.edu.hk; jinshuyuan@yahoo.com.cn).

X. Wang was with the School of Mathematics and Computer Science, Hebei University, Baoding, China. He is now with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2006.889480

distinguish various types of flooding attacks from each other is still a challenge to constructing an effective detection system.

Based on the investigation of correlativity changes of monitored network features during flooding attacks [9], [11], [12], this paper utilizes the changes of correlations among features in the detection. Although some approaches in the literature also take advantage of correlation changes to detect flooding attacks, the correlativity changes in these approaches is utilized either indirectly [7], [10] or partially [11], [12]. For example, the approaches in [11] and [12] only use correlativity changes to identify flooding attacks from normal traffic, but they do not further reveal the possible insights into the behavioral patterns of flooding attacks that may be exhibited by the correlativity changes. The possible insights include, e.g., whether the correlativity changes can be directly utilized in the detection or whether the correlativity changes can serve as the second-order features to distinguish various flooding attacks from each other. The answers to these questions will greatly improve our understanding of the normal traffic and flooding attacks, which will further help us take effective responses to maintain the security of information systems. From this sense, this paper proposes a general modeling and detection approach. It focuses on answering the questions of: 1) how to directly utilize the correlativity changes to construct an effective detection system so that the behavioral properties of various flooding attacks, especially the properties of various unknown flooding attacks, can be revealed and 2) in which ways to mark the detected flooding attacks so that various flooding attacks can be distinguished from each other in terms of correlativity changes.

This paper employs the covariance matrix to model and detect various flooding attacks. The covariance matrix, as one of the second-order statistics, is directly utilized to reveal the characteristics of different classes (normal traffic or various types of flooding attacks) in terms of correlation changes among monitored features. Our detection approach has three main characteristics.

1) It models the traffic samples provided by network-monitoring devices into covariance-matrix samples. This modeling process enables the approach to directly make use of the differences of correlation information among network features in the detection.
2) It not only detects various flooding attacks, but further extracts the second-order features for the detected flooding attacks—a useful tool to identify various unknown flooding attacks.
3) It is independent of prior data-distribution assumptions. Since a covariance matrix is constructed based on a sequence of samples, the statistical distribution information of the population has been embodied in the covariance matrix when a suitable sequence length is selected.

The rest of this paper is organized as follows. Section II provides the background of this paper. Section III details the covariance-matrix modeling process and gives the problem representation. Section IV describes the detection approach, where the dissimilarity function, threshold determination algorithm, and 0–1 matrix concept are introduced. Section V evaluates the performance the detection approach by detecting two types of common flooding DDoS attacks—Neptune (also known as SYN flooding) and Smurf attacks. Section VI presents the validation results and makes some discussions. Section VII discusses some practical implementation issues. In the end, Section VIII states a conclusion.

II. Related Works

Basically, the techniques of detecting various flooding attacks belong to the network anomaly detection category. In the category of network anomaly detection, many different detection techniques exist such as neural network [23], [32], clustering [31], Markov model [29], wavelet analysis [19], specification-based detection [17], [18], and statistical detection [2]–[10]. In the detection of flooding attacks, the statistical detection approaches are widely employed among others. The detailed surveys of statistical detection development in this area have appeared in the literature [3]–[5]. In this section, we review the statistical detection techniques for flooding attacks mainly from the point of view of the problem formulation.

To detect flooding DoS attacks, some detection approaches use the macroscopic formulation. For example, Xiong *et al.* [24] formulate the problem of high concentration of malicious DDoS packets to the victim as a similar hot-spot problem as observed in a multiprocessor system, where a hot spot is formed when a large number of processors simultaneously access the shared variables in the same memory module. Kong *et al.* [25] model the mitigation of flooding DoS attacks as a controllable material flow. However, since there is a lack of the underlying traffic models, the macroscopic formulation cannot ensure the applicability of their derived results [24].

In addition to the high-level solutions mentioned above, many other flooding-attack-detection approaches are based on the Bayesian formulation, in which the statistics of different network features are evaluated. For example, Jung *et al.* [27] evaluate various statistics such as the distribution of page access number or the distribution of client number to distinguish the normal hypertext transfer protocol (HTTP) requests from flooding DoS attacks. Blazek *et al.* [10] regard the protocol utilization as the evaluated statistical variable. Manikopoulos and Papavassiliou [8] regard different frequencies of selected packet attributes such as protocol or service in the protocol headers as the evaluated statistical variables. Ohsita *et al.* [28] evaluate the statistics of time variation of different flow traffic to detect DoS attacks. The approaches mentioned above mainly utilize the first-order statistics to distinguish flooding attacks from normal traffic. Few of them consider the second-order statistics of the observed subjects in the detection of flooding attacks. A drawback of the Bayesian-formulation-based detection approaches is that the detection approaches need prior distribution assumptions [1], [6]. If the evaluated statistical variables are not distributed as presumed, the detection techniques will yield a high false alarm rate [6].

Recent work mainly focuses on sequential change-point detection for flooding attacks. The detection approaches are based on time-series formulation. For example, the abrupt change-detection approach in [11] determines the anomalies by analyzing the abrupt changes in Management Information Base

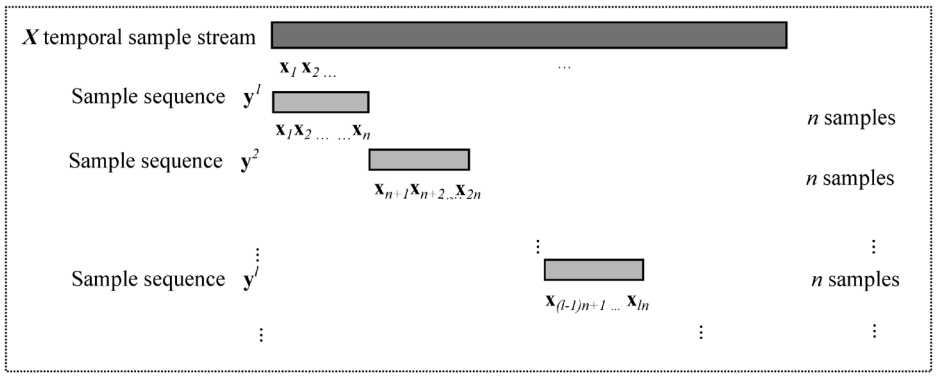

Fig. 1. Segmentation of observed temporal sample stream.

(MIB) variables. The change-point detection approach in [12] detects the SYN flooding attack based on the changed statistics between the number of SYN packets and the number of FIN or SYN/ACK packets. The sequential change-point-based detection approaches make effective detection of any abrupt change in network traffic. However, they fail to reveal the meaning of abrupt changes to flooding attacks, e.g., whether different abrupt changes correspond to different network attacks.

Different from the detection models mentioned above, where the evaluated statistical variables are the first-order statistics, our detection model utilizes the second-order statistics of monitored-network features as the evaluated variables. Specially, we employ the covariance matrices of the sequential samples and propose a covariance-matrix-based detection approach to detecting various types of flooding attacks. The covariance matrix is directly utilized in the detection in order to evaluate the changes of correlations among monitored features. The detection model presented here overcomes the limitations of prior assumptions of data distribution in the Bayesian formulation. Moreover, it further reveals the specific meanings of the second-order statistics to different types of flooding attacks.

III. Covariance-Matrix Modeling and Problem Representation

A. Covariance-Matrix Modeling

Note that the abnormal packets of flooding attacks are always amassed quickly to a single victim in order to exhaust the resources. The statistical properties within that period will mainly reflect the traffic behavioral properties of the flooding attacks, which should be different from that of the normal traffic. Therefore, we can make use of the statistical properties contained in the temporally sequential samples to detect the flooding attacks. To exhibit the correlativity of the underlying network traffic, we use statistical covariance matrices to model the sample sequences of equal and fixed length. Each element in a covariance matrix describes the correlation between any two monitored features of the corresponding sample sequence. Mathematically, the covariance-matrix modeling process can be described as follows. Assume an observed sample $\mathbf{x}$ has p features. It can be represented as a random vector $\mathbf{x} = (f_1, f_2, \ldots, f_p)^{\mathrm{T}}$. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be n observations, where $\mathbf{x}_i = (f_1^i, f_2^i, \ldots, f_p^i)^{\mathrm{T}}$ is the ith observed vector. We define a new variable $\mathbf{y}$, which represents the following statistics $\mathbf{y}^l$ related to p features of the lth sequence of length n:

$$\mathbf{y}^l = (\mathbf{x}_1^l, \ldots, \mathbf{x}_n^l)^{\mathrm{T}} \tag{1}$$

where $\mathbf{x}_k^l = (f_1^{l,k}, f_2^{l,k}, \ldots, f_p^{l,k})^{\mathrm{T}}$, $1 \le k \le n$. The definition (1) can be represented in detail as

$$\mathbf{y}^l = \begin{pmatrix} f_1^{l,1} & f_2^{l,1} & \cdots & f_p^{l,1} \\ f_1^{l,2} & f_2^{l,2} & \cdots & f_p^{l,2} \\ \vdots & \vdots & \ddots & \vdots \\ f_1^{l,n} & f_2^{l,n} & \cdots & f_p^{l,n} \end{pmatrix} \tag{2}$$

where $f_u^{l,k}$ is the value of f_u in the kth observation in the lth sequence. Parameters u, l, and k are integers and satisfy the conditions of $1 \le u \le p$, $1 \le l < \infty$, and $1 \le k \le n$.

We use the covariance matrix $\mathbf{M}^l$ to characterize the variable $\mathbf{y}^l$ as follows:

$$\mathbf{M}^l = \begin{pmatrix} \sigma_{f_1^l, f_1^l} & \sigma_{f_1^l, f_2^l} & \cdots & \sigma_{f_1^l, f_p^l} \\ \sigma_{f_2^l, f_1^l} & \sigma_{f_2^l, f_2^l} & \cdots & \sigma_{f_2^l, f_p^l} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{f_p^l, f_1^l} & \sigma_{f_p^l, f_2^l} & \cdots & \sigma_{f_p^l, f_p^l} \end{pmatrix} \tag{3}$$

where $\sigma_{f_u^l, f_\nu^l} = \mathbf{cov}(f_u^l, f_\nu^l) = 1/n \sum_{k=1}^{n} (f_u^{l,k} - \mu_{f_u^l})(f_\nu^{l,k} - \mu_{f_\nu^l})$, and $\mu_{f_u^l} = \mathbf{E}(f_u^l) = 1/n \sum_{k=1}^{n} f_u^{l,k}$.

The covariance-modeling process can be regarded as a data preprocess, where the correlations of sample sequences of equal and fixed length n are represented by covariance matrices. In practice, we obtain a temporal sample stream through the continuous sampling of the network-monitoring devices. The covariance-modeling process first segments the temporal sample stream into all nonoverlapped sequences of length n and then calculates the covariance matrices of the sequential samples. Fig. 1 illustrates the relationship between the observed temporal sample sequences and their corresponding covariance matrices.

In conceptual terms, the modeling process can be regarded as a process of a new covariance feature space construction. Each dimension of the covariance feature space gives

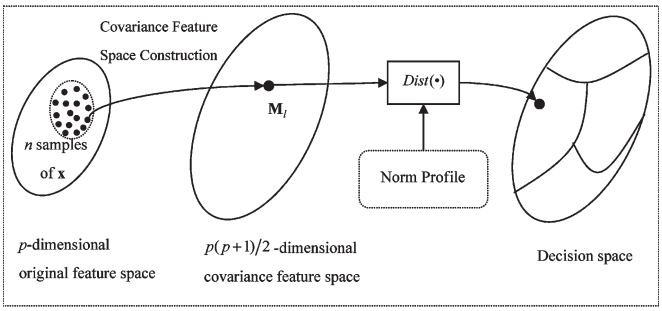

Fig. 2. Illustration of the covariance-matrix-based detection model.

a measure of correlation between any pair of monitored features. Since an original sample is described by p features, a sample sequence of length n can be viewed as n points of $\mathbf{x}$ in the original feature space with p dimensions. The covariance-matrix-modeling process can, thus, be described as a transformation that maps the temporal n samples into an intermediate point $\mathbf{y}$ in a $p(p+1)/2$-dimensional covariance feature space. The covariance-modeling process is illustrated on the left in Fig. 2.

B. Problem Representation

Based on the covariance feature space construction illustrated in Fig. 2, the problem of detecting various flooding attacks can be posed as follows.

The norm profile of normal traffic is represented by the mathematical expectation of all covariance matrices constructed from all nonoverlapped sequential samples of the normal class in the training dataset. Given a testing covariance matrix, the detection approach will determine whether the presented covariance matrix is greatly different from the norm profile by means of some Dist($\bullet$) function. The modeling and detection flow of our detection model is illustrated in Fig. 2.

Since each element in the covariance matrix exhibits the correlation between two corresponding features, the difference matrix or the result matrix of function Dist($\bullet$) will represent the correlation differences between the observed sequential samples and the normal traffic. If the correlation differences are significant, a flooding attack will be signaled.

IV. Detection Approach

A. Dissimilarity Function and 0–1 Result Matrix

We use the symbol $\mathbf{N}$ to denote the norm profile of normal traffic. The dissimilarity function Dist($\bullet$) between an observed covariance matrix $\mathbf{M}^{\text{obs}}$ and $\mathbf{N}$ is defined as follows:

$$\text{Dist}(\mathbf{M}^{\text{obs}}, \mathbf{N}; \mathbf{T}) = (d_{u\nu})_{p\times p} \quad \forall m_{u\nu}^{\text{obs}} \in \mathbf{M}^{\text{obs}} \quad \forall n_{u\nu} \in \mathbf{N} \quad \forall \delta_{u\nu} \in \mathrm{T}$$
$$d_{u\nu} = \begin{cases} 1, & \text{if} \left| m_{u\nu}^{\text{obs}} - n_{u\nu} \right| \geq \delta_{u\nu} \\ 0, & \text{if} \left| m_{u\nu}^{\text{obs}} - n_{u\nu} \right| < \delta_{u\nu} \end{cases} \tag{4}$$

where $\mathbf{T}$ is the dissimilarity threshold matrix. Each element $\delta_{u\nu}$ in $\mathbf{T}$ restricts the range within which the element $m_{u\nu}^{\text{obs}}$ in $\mathbf{M}^{\text{obs}}$ can be different from $n_{u\nu}$ in $\mathbf{N}$.

Note that the result of function Dist($\bullet$) is a matrix whose elements are either zeros or ones. We call it 0–1 matrix. A 0–1 matrix can represent a total of $2^{[p(p+1)/2]}$ different dissimilarity results. If two 0–1 matrices have different number of ones, they are different. If two 0–1 matrices have the same number of ones, but the ones appear at different positions (the coordinates of rows and columns), they should be considered different too. For example, if

$$\text{Dist}(\mathbf{M}^{\text{obs1}}, \mathbf{N}; \mathbf{T}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$
$$\text{Dist}(\mathbf{M}^{\text{obs2}}, \mathbf{N}; \mathbf{T}) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

we can draw two conclusions: 1) both $\mathbf{M}^{\text{obs1}}$ and $\mathbf{M}^{\text{obs2}}$ are significantly different from norm profile $\mathbf{N}$, which means that both $\mathbf{M}^{\text{obs1}}$ and $\mathbf{M}^{\text{obs2}}$ are detected as attacks and 2) $\mathbf{M}^{\text{obs1}}$ is not equal to $\mathbf{M}^{\text{obs2}}$ since there are different positions of value one in their dissimilarity results, which means $\mathbf{M}^{\text{obs1}}$ and $\mathbf{M}^{\text{obs2}}$ come from different types of attacks. That $\mathbf{M}^{\text{obs}}$ belonging to the normal class is true if and only if $\text{Dist}(\mathbf{M}^{\text{obs}}, \mathbf{N}; \mathbf{T}) = [0]_{p\times p}$.

By introducing the 0–1 matrix to evaluate the difference between the observed covariance matrix and the norm profile, our detection approach will enjoy the advantage of further extracting the patterns of the unknown attacks. Note that in a detection result represented by a 0–1 matrix, the positions with value ones exactly correspond to correlations which are significantly different from the norm profile. Therefore, the positions with value ones in the result matrix can serve as the second-order features to mark the detected flooding attack.

B. Threshold Determination

Let us assume that the variable $\mathbf{E}(\mathbf{M}^l)$ denotes the mathematical expectation of all covariance matrices of the normal traffic. The variable $\mathbf{M}^l$ (3) denotes the covariance matrix of the lth temporal sample sequence of length n of the normal traffic. Since the threshold matrix $\mathbf{T}$ (4) restricts the distance between $\mathbf{M}^l$ and $\mathbf{E}(\mathbf{M}^l)$, the threshold matrix essentially reflects the statistical variance of the random vector $(\mathbf{M}^l - \mathbf{E}(\mathbf{M}^l))$.

In order to provide a practical solution to determining a suitable threshold matrix, we employ the Chebyshev inequality theorem. The Chebyshev inequality is a result in probability theory [35]. It gives a lower bound for the probability that a random variable with finite variance lies within a certain distance from the variable's mean.

Let X be a random variable with a finite mean $\mathbf{E}(X)$ and a finite variance $\mathbf{D}(X)$. Then for any positive real number ε, we have

$$P\left(|X - \mathbf{E}(X)| \geq \varepsilon\right) \leq \frac{\mathbf{D}(X)}{\varepsilon^2}. \tag{5}$$

Equivalently

$$P\left(|X - \mathbf{E}(X)| < \varepsilon\right) \geq 1 - \frac{\mathbf{D}(X)}{\varepsilon^2}. \tag{6}$$

Equation (6) shows the probability that a random variable will assume a value lies within a certain distance from the variable's mean is greater than $(1 - \mathbf{D}(X)/\varepsilon^2)$.

In our case, we have (7), shown at the bottom of the page. In statistical theory, $\mathbf{E}(\mathbf{M}^l) = \mathbf{cov}(X)$, where $\mathbf{cov}(X)$ denotes the covariance matrix of the population of normal traffic X. Therefore, we have

$$\mathbf{M}^l - \mathbf{E}(\mathbf{M}^l) = \begin{pmatrix} \sigma_{f_1^l f_1^l} - \sigma_{f_1 f_1} & \sigma_{f_1^l f_2^l} - \sigma_{f_1 f_2} & \cdots & \sigma_{f_1^l f_p^l} - \sigma_{f_1 f_p} \\ \sigma_{f_2^l f_1^l} - \sigma_{f_2 f_1} & \sigma_{f_2^l f_2^l} - \sigma_{f_2 f_2} & \cdots & \sigma_{f_2^l f_p^l} - \sigma_{f_2 f_p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{f_p^l f_1^l} - \sigma_{f_p f_1} & \sigma_{f_p^l f_2^l} - \sigma_{f_p f_2} & \cdots & \sigma_{f_p^l f_p^l} - \sigma_{f_p f_p} \end{pmatrix}. \tag{8}$$

Since we are interested in detecting significant correlation changes which is represented by $\sigma_{f_u^l f_\nu^l} - \mathbf{E}(\sigma_{f_u^l f_\nu^l})$, we only need to set the lower bound for each element in the threshold matrix. Thus, for each element in the difference matrix $(\mathbf{M}^l - \mathbf{E}(\mathbf{M}^l))$, according to Chebyshev inequality, we obtain

$$\forall (u,\nu), P\left(\left| \sigma_{f_u^l f_\nu^l} - \mathbf{E}(\sigma_{f_u^l f_\nu^l}) \right| < \varepsilon \right) \geq 1 - \mathbf{D}(\sigma_{f_u^l f_\nu^l})/\varepsilon^2. \tag{9}$$

In (9), let $\varepsilon = 3\sqrt{D(\sigma_{f_u^l f_\nu^l})}$ and $\varepsilon = 4\sqrt{D(\sigma_{f_u^l f_\nu^l})}$, respectively, we can obtain

$$\forall (u,\nu), P\left(\left| \sigma_{f_u^l f_\nu^l} - \sigma_{f_u f_\nu} \right| < 3\sqrt{D(\sigma_{f_u^l f_\nu^l})} \right) \geq 1 - \frac{1}{9} \tag{10}$$

$$\forall (u,\nu), P\left(\left| \sigma_{f_u^l f_\nu^l} - \sigma_{f_u f_\nu} \right| < 4\sqrt{D(\sigma_{f_u^l f_\nu^l})} \right) \geq 1 - \frac{1}{16} \tag{11}$$

where $\mathbf{D}(\sigma_{f_u^l f_\nu^l}) = 1/s \sum_{l=1}^{s} (\sigma_{f_u^l f_\nu^l} - \sigma_{f_u f_\nu})^2$, s is the total number of sequences of length n in the training set.

Equation (9) provides a solution to determining the value of each element in the threshold matrix subject to the detection probability of normal traffic. For example, if the requirement of the probability of accurate detection for normal traffic is (1–1/9), the lower bound of the threshold matrix should be set to $3\sqrt{D(\sigma_{f_u^l f_\nu^l})}$ indicated in (10). Similarly, if the requirement of the probability of accurate detection for normal traffic is (1–1/16), the lower bound of the threshold matrix should be set to $4\sqrt{D(\sigma_{f_u^l f_\nu^l})}$ indicated in (11).

The Chebyshev inequality is valid for any stochastic variable regardless of what distribution of the variable follows. Therefore, this threshold-matrix determination process provides a lower bound solution constrained by the detection probability, regardless of any distribution assumption of the normal traffic. In the detection of various flooding attacks, if there exists an element in $(\mathbf{M}^l - \mathbf{cov}(X))$ which is greater than the corresponding boundary (e.g., $3\sqrt{D(\sigma_{f_u^l f_\nu^l})}$ or $4\sqrt{D(\sigma_{f_u^l f_\nu^l})}$) settled in the threshold matrix, we will signal an anomaly.

C. Detection

In practice, a network traffic record for each sampling event contains a variety of information including the number of packets to the same host, the packet rate, the number of connections to the same host, the number of connection that have "SYN" errors to the same host, and so on. In this paper, we extract and utilize nine network time-based traffic features from the sampled records. The meaning of these nine features and the reasons why we utilize them are provided in Section V.

The network traffic to and from a host machine are captured through a continuous sampling process. Each of the samples is characterized by nine features and each sequence of n samples is characterized by its corresponding covariance matrix. For intrusion detection, we will first build a norm profile to present a long term of normal traffic through training, and then, compare the traffic in the recent past to the long-term norm profile for the detection of any significant deviations. The long-term profile of normal network traffic measured by the covariance matrix is characterized by the sample mean. We can obtain the sample mean from all covariance matrix samples in the training data. During the online detection, we define the network activities in the recent past from the $(\mathrm{obs} - n)$th packet to the current obsth packet as the covariance matrix $\mathbf{M}^{\mathrm{obs}}$. Each element $\sigma_{u\nu}$ in $\mathbf{M}^{\mathrm{obs}}$ represents the linear correlation between f_u and f_ν, measuring the covariance of the monitored features f_u and f_ν in the same time period of these n samples collected. The multivariate observation $\mathbf{M}^{\mathrm{obs}}$, thus, represents the covariance among various monitored features in the recent past.

At the observed sequence, we compare the sequential network traffic represented by $\mathbf{M}^{\mathrm{obs}}$ with the norm profile represented by $\mathbf{N}$ under the threshold matrix $\mathbf{T}$ as follows: $\mathbf{M}^{\mathrm{obs}}$ is considered normal, if $\mathrm{Dist}(\mathbf{M}^{\mathrm{obs}}, \mathbf{N}; \mathbf{T}) = [0]_{p \times p}$; otherwise, an alarm will be signaled. If we signal an alarm, the positions of significant deviations of the currently observed covariance matrix from the norm profile measured by the 0–1 result matrix will be provided as the detailed information, to help the network administrator find out the second-order features of the detected attack.

V. Validations

As case studies, we evaluate the performance of the proposed detection approach by detecting two common flooding DDoS attacks—Neptune and Smurf attacks. The aim of the case

studies in this section is to demonstrate the effectiveness of the detection approach in two respects of: 1) detecting the unknown flooding attacks and 2) extracting the second-order features for the detected flooding attacks.

Similar to most anomaly detection approaches, the basic principles of our detection approach are to build a norm profile first, and then, to determine the attack when any significant deviation from the norm profile happens. In the first stage, we will obtain a detector through a supervised training process. The detector will be only provided with the samples of the normal class in the training stage. We use the sample center of the normal traffic data and a determined boundary that constrains the range of all covariance matrix samples deviate from their center to build the norm profile for the normal traffic. In the second stage, we utilize the detector obtained from training to label the unknown attack samples—the testing samples that do not appear in the training stage. Since no samples from the attack class are provided in the training dataset, the effectiveness of our approach will be validated if the detector can correctly detect the attack samples in the testing dataset as attack.

We will also illustrate how to utilize the 0–1 result matrix to extract the second-order statistics for the detected attacks. As discussed earlier, we will assign the values of either ones or zeros to different positions in the detection-result matrix, in order to magnify the significant deviations of the observed covariance matrix from the norm profile. The positions assigned with value ones in the result matrix represent the significant deviations. The positions assigned with the value zeros represent no significant deviations. In the case studies, we find that the ones' positions in the result matrix are stable in detecting the testing samples from the same class. Therefore, we extract the covariance feature set which stably exhibits the significant deviations of the detected attack from the normal traffic as the second-order features to mark the detected flooding attack. We will use the average result matrix as one of performance indexes to present the results of such feature extraction.

A. Detection-Result Specification

In order to manifest the characteristics of our detection approach to various flooding attacks, we use two indexes to indicate the performance of our detection approach. One is the detection rate, which will present the detection accuracy. The other is the average detection-result matrix, which will present the second-order features for the detected flooding attacks.

1) Detection Rate: The detection rate of a class is defined as the probability of correctly detecting the testing samples in the class. For example, if the detection approach can accurately detect m covariance matrices as the normal class from a total of n covariance matrices in the normal class, the detection rate will be $(m/n) \times 100\%$. This result indicates that we will have $(m/n) \times 100\%$ faith to believe that the approach can detect the normal class accurately. Similarly, the detection rate of a flooding attack will indicate how much faith can be placed on the conclusion that the detection approach can accurately detect the attack.

2) Average Detection-Result Matrix: For each $\mathbf{M}^{\text{obs}}$ in the testing dataset, we will obtain a 0–1 dissimilarity result matrix under the function of $\text{Dist}(\mathbf{M}^{\text{obs}}, \mathbf{cov}(X); \mathbf{T})$ (4). In order to reflect detection result statistically, we sum up all 0–1 result matrices in detecting all samples in a particular class and use the average sum to represent the detection result. The average detection-result matrix is defined as

$$(1/s) \sum_{l=1}^{s} \text{Dist}\left(\mathbf{M}^{l}, \mathbf{cov}(X); \mathbf{T}\right) \tag{12}$$

where s is the total number of sample sequences in a particular class.

Clearly, the value of each element in the average detection-result matrix is between zero and one. A nonzero element in the average detection-result matrix reports the statistical probability of its corresponding correlation of the detected attack changed significantly in comparison with the norm profile. The larger a nonzero element is, the more faithful we are willing to believe that this position will mark the second-order statistical features for the detected flooding attack. The positions with value ones report that we have 100% confidence to believe that the correlations at these positions will always significantly deviate from the norm profile during detection.

B. Training and Testing Data

The dataset we employ is a subset of KDD CUP 99 dataset at http://kdd.ics.uci.edu/databases/kdd cup99/kddcup99.html. Although the KDD CUP dataset has many flaws [16], it is the only public dataset with labeled attack samples we can find. The attack samples in the dataset are obtained by passive monitoring, rather than by inserting the attack packets into the normal traces. Therefore, we select this public dataset as the base dataset in our case studies.

As a public dataset, KDD CUP 99 contains many different types of attacks for the purpose of network intrusion detector competition, such as Probe, User-to-Root (U2R), Remote-to-Local (R2L), and DoS attacks. A detailed description of different types of attacks contained in KDD CUP 99 dataset is provided in.[1] In spite of other attack types, Neptune and Smurf attacks are the only flooding DDoS attacks labeled in the whole training dataset. Therefore, we extract all records with the labels of Normal, Neptune, and Smurf from the whole training dataset to form the dataset used in our case studies. Our training dataset only includes the Normal records, while our testing dataset includes all records of the Normal, Neptune, and Smurf classes. The description of the datasets used in our case studies is presented in Table I.

C. Features Used

The KDD CUP dataset provides a total of 41 features to describe various types of attacks. These features are grouped

[1] KDD CUP 1999 DATA, Dataset used for the Third International Knowledge Discovery and Data Mining Tools Competition, in conjunction with KDD-99 the Fifth International Conference on Knowledge Discovery and Data Mining, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

TABLE I
DATASET DESCRIPTION

Dataset	Training Set	Testing Set		
	Normal	Normal	Neptune	Smurf
Number of Records	972780	972780	1072017	2807886
Number of Corresponding Covariance Matrix*	6485	6485	7146	18719

*sequence length is selected as 150 as discussed in Part D, Section V.

TABLE II
DESCRIPTION OF TIME-BASED TRAFFIC FEATURES DERIVED FROM [13]

Label	Features	Description
1	count	number of connections to the same host as the current connection in the past two seconds
2	srv_count	number of connections to the same service as the current connection
3	serror_rate	% of connections that have ``SYN" errors to the same host
4	srv_serror_rate	% of connections that have ``SYN" errors to the same service
5	rerror_rate	% of connections that have ``REJ" errors to the same host
6	srv_serror_rate	% of connections that have ``SYN" errors to the same service
7	same_srv_rate	% of connections to the same service on same host
8	diff_srv_rate	% of connections to different services on same host
9	srv_diff_host_rate	% of connections to different hosts

into three sets: basic, content, and time-based traffic features. The basic and content feature groups describe the host audit and log information. The time-based traffic features describe the network connections and traffic information. For example, one of the basic features called *src_bytes* describes the number of data bytes from source to destination; the feature *num_failed_logins* in the content feature group describes the number of failed login attempts. A detailed description of these 41 features is available (see footnote 1).

As a conclusion in [15], detecting different categories of intrusions require different feature groups. The basic and content features are suitable in the detection of host-based intrusions such as U2R or R2L intrusions, while the time-based traffic features are more suitable in the detection of DoS and probing attacks, which are typical flooding attacks [15]. Therefore, we employ all nine time-based traffic features in this paper. All of them are continuous type, shown in Table II. These nine time-based traffic features reflect 2-s Transmission Control Protocol (TCP) connection statistics, which can be online obtained using a packet monitoring and capturing program [30].

D. Parameter Settlement

There are several parameters to be chosen, namely the fixed sequence length n and the threshold matrix $\mathbf{T}$. The parameter n controls the number of the samples observed in each covariance matrix calculation, while the threshold $\mathbf{T}$ restricts the degree to which the evaluated covariance matrix $\mathbf{M}^l$ is different from the norm profile $\mathbf{E}(\mathbf{M}^l)$.

To select a suitable sequence length n, we utilize the statistics of maximum value in the difference matrix. The maximum value in the difference matrix is denoted as $\max(|d^l_{uv}|), \forall d^l_{uv} \in (\mathbf{M}^l - \mathbf{E}(\mathbf{M}^l))$, where $\mathbf{M}^l$ is the covariance matrix of the lth sample sequence and $\mathbf{E}(\mathbf{M}^l)$ is equal to the covariance matrix of the normal population represented by $\mathbf{cov}(X)$. For a certain length n, we can obtain a sequence of maximum value $H: \max(|d^1_{uv}|), \ldots, \max(|d^s_{uv}|)$, where each $\max(|d^l_{uv}|)$ represents the maximum element in the difference matrix $(\mathbf{M}^l - \mathbf{cov}(X))$. The relationship between $(\mathbf{M}^l - \mathbf{cov}(X))$ and n can be reflected by the mean and the standard deviation of sequence H. The mean of H versus the sequence length n is shown in Fig. 3, where x axis represents the sequence length n and y axis represents the value of $(1/s)\sum_{i=1}^{s}[\max(|d^l_{uv}|)]$, $\forall d^l_{uv} \in (\mathbf{M}^l - \mathbf{cov}(X))$. The standard deviation of H versus the sequence length n is shown in Fig. 4, where x axis represents the sequence length n and y axis represents the value of $\mathbf{std}[\max(|d^l_{uv}|)], \forall d^l_{uv} \in (\mathbf{M}^l - \mathbf{cov}(X))$.

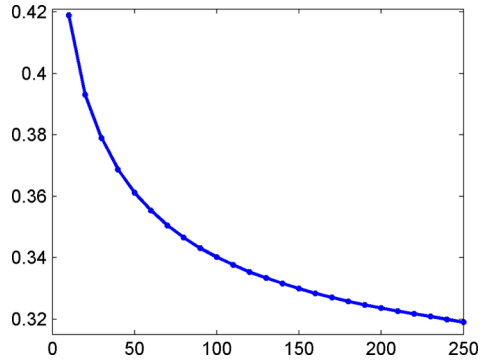

Fig. 3. Mean of H versus sequence length n.

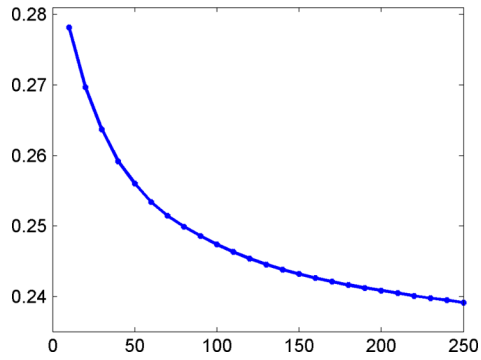

Fig. 4. Standard deviation of H versus sequence length n.

From Figs. 3 and 4, we can find that the larger the sequence length n is, the less of the difference between the covariance matrix of a sample sequence and the covariance matrix of the population is. The sequence length n is a feature-dependent parameter based on the training dataset. A suitable n can be selected as a relatively stable value. In the case studies, we select n as 150, where the corresponding mean and the corresponding standard deviation of H is 0.3299 and 0.2432, respectively. At this point, $\Delta\mathbf{E}(H)$ levels out at 0.0016 and $\Delta\mathbf{std}(H)$ levels out at 5.9511e-004.

The threshold matrix $\mathbf{T}$ is determined with constraint of the detection probability of normal traffic. According to the threshold determination algorithm discussed in Section IV-B, we use two different threshold matrices to evaluate the performance of our detection approach in the validations: one is set to $3\mathbf{D}(\mathbf{M})$, and the other is set to $4\mathbf{D}(\mathbf{M})$. The definition of $\mathbf{D}(\mathbf{M})$ is

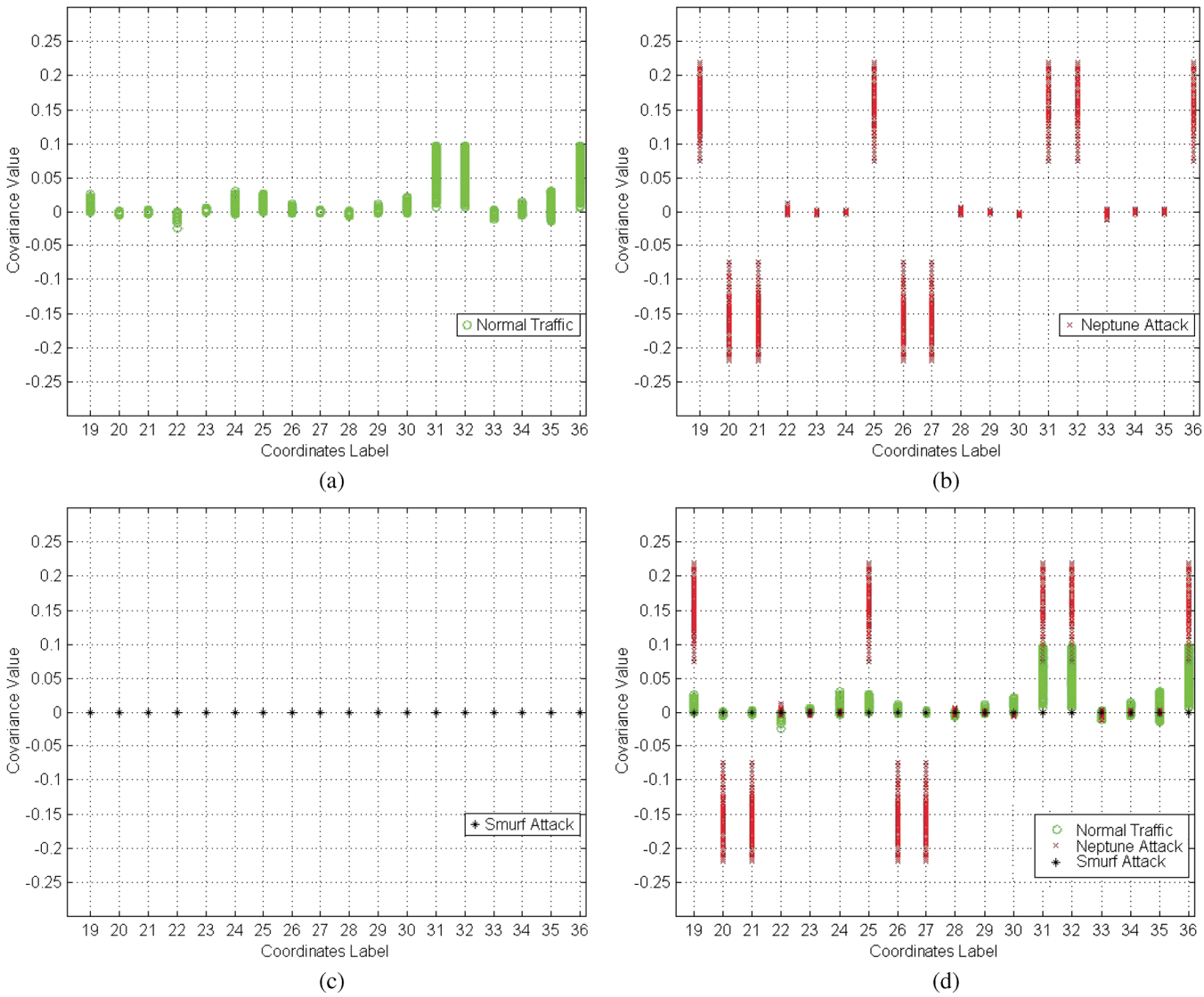

Fig. 5. Covariance distributions of the Normal, Neptune, and Smurf traffic. (a) Covariance distributions of the normal traffic. (b) Covariance distributions of the Neptune attack. (c) Covariance distributions of Smurf attack. (d) Covariance distributions of three classes.

given in (13), where each element can be obtained in the training stage

$$\mathbf{D}(\mathbf{M}) = \begin{pmatrix} \mathbf{std}(\sigma_{f_1^l f_1^l}) & \mathbf{std}(\sigma_{f_1^l f_2^l}) & \cdots & \mathbf{std}(\sigma_{f_1^l f_p^l}) \\ \mathbf{std}(\sigma_{f_2^l f_1^l}) & \mathbf{std}(\sigma_{f_2^l f_2^l}) & \cdots & \mathbf{std}(\sigma_{f_2^l f_p^l}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{std}(\sigma_{f_p^l f_1^l}) & \mathbf{std}(\sigma_{f_p^l f_2^l}) & \cdots & \mathbf{std}(\sigma_{f_p^l f_p^l}) \end{pmatrix}. \quad (13)$$

Theoretically, we can at least obtain the following detection probability in detecting the normal traffic:

$$\begin{cases} P\left(\mathbf{M}^l - \mathbf{cov}(X)| < 3\mathbf{D}(\mathbf{M})\right) \geq 1 - \frac{1}{9} = 0.8889 \\ P\left(\mathbf{M}^l - \mathbf{cov}(X)| < 4\mathbf{D}(\mathbf{M})\right) \geq 1 - \frac{1}{16} = 0.9375 \end{cases} \quad (14)$$

where $\mathbf{M}^l$ is the covariance matrix of the lth sample sequence of length 150 and $\mathbf{cov}(X)$ is the covariance matrix of the normal population.

VI. Results and Discussions

This section describes the results obtained by applying the $3\mathbf{D}(\mathbf{M})$ and $4\mathbf{D}(\mathbf{M})$ as the threshold matrices, respectively, as described in the previous section, in order to validate the performance of the detection approach.

A. Covariance Distributions of Experimental Data

To validate the covariance differences among the normal traffic and different flooding attacks, we plot the covariance values of all samples of different classes in the testing dataset. As we know, the covariance feature space in the validations has a dimension of 45, where the feature number p equals nine. In order to present the changes of covariance on different dimensions clearly, we convert the form of a matrix into a multidimensional vector with the rules as $(1,1) \Rightarrow 1$, $(1,2) \Rightarrow 2$, $(1,9) \Rightarrow 9$, and $(2,2) \Rightarrow 10 \ldots$. Since a covariance matrix is symmetric, in the transformation, we only consider the elements in the upper triangle matrix. The transformation rule is that any coordinates

$\forall(u,\nu)\ 1 \le u \le \nu \le p$ in a covariance matrix will correspond to the entry $u^*(2p-u+1)/2-(p-\nu)$ in its corresponding vector. For example, the coordinates (3, 4) in a covariance matrix will correspond to the entry $3^*(2^*9-3+1)/2-(9-4)=19$ in the corresponding vector. The coordinates (3, 5) in a covariance matrix will correspond to the entry 20 in the corresponding vector. The coordinates (9, 9) in a covariance matrix will correspond to the last entry 45 in the corresponding vector. Fig. 5 demonstrates the changes of covariance values of all samples of different classes in the testing dataset, from the dimension 19 to the dimension 36. The x axis represents the entry label in the corresponding vector. The y axis represents the covariance value. Fig. 5(a) shows the covariance distributions of the normal traffic. Fig. 5(b) shows the covariance distributions of the Neptune attack traffic. Fig. 5(c) shows the covariance distributions of the Smurf attack traffic. Fig. 5(d) shows the differences of covariance distributions of these three different classes.

Fig. 5(a)–(c) shows that the covariance distribution on each dimension of a particular class concentrates stably within a certain range. Fig. 5(d) further shows that different classes have different covariance distributions obviously. For instance, in Fig. 5(d), the distribution of the $\sigma_{f_3f_5}$ (corresponding to the 20th label on x axis) of the Neptune attack traffic denoted by plus symbols is clearly different from that of the normal traffic denoted by circles. Thus, in this paper, we utilize the covariance matrix differences to distinguish different flooding attacks from the normal traffic. We use a threshold matrix to constrain the range, within which an observed covariance matrix can be different from the norm profile, and introduce the 0–1 detection-result matrix to characterize the significant covariance differences in the detection.

B. Detection Rate

Table III summarizes the detection rates under the threshold matrices of $3\mathbf{D}(\mathbf{M})$ and $4\mathbf{D}(\mathbf{M})$. Table III shows that the experimental detection rate for the normal class is lower than the theoretical value of 88.89% under the threshold matrix of $3\mathbf{D}(\mathbf{M})$, while it is nearly equal to the theoretical value of 93.75% under the threshold matrix of $4\mathbf{D}(\mathbf{M})$. The 100% detection rates for the Neptune and Smurf attacks in Table III also validate that the flooding attacks are significantly different from the normal class in terms of the correlations among monitored features.

For a given test, different threshold matrices will lead to different pairs of false alarm rate and detection rate. Table IV shows some pairs of false alarm rate and detection rates under different threshold matrices of different multipliers by $\mathbf{D}(\mathbf{M})$.

Table IV shows that the covariance-matrix-based detection approach achieves very high detection rates and low false alarm rates on the experimental data. The 100% detection rates also show the high sensitivity of our detection model in the detection of flooding attacks, which will be attributed to the utilization of a matrix rather than a scalar as the threshold to evaluate the covariance changes. Since each entry in the threshold matrix evaluates the changes of the covariance of two corresponding features, it is easy to happen that the changes of some elements in the observed covariance matrix exceed their corresponding elements in the threshold matrix.

TABLE III
DETECTION RATE ON THE TESTING DATASET

Threshold Matrix	Detection Rate		
	Normal	Neptune	Smurf
3**D**(**M**)	75.77%	100%	100%
4**D**(**M**)	93.63%	100%	100%

TABLE IV
FALSE ALARM AND DETECTION RATES ON THE TESTING DATASET

Threshold Matrix	False Alarm Rate	Detection Rate	
		Neptune	Smurf
3.0**D**(**M**)	24.23%	100.00%	100.00%
3.5**D**(**M**)	15.64%	100.00%	100.00%
4.0**D**(**M**)	6.37%	100.00%	100.00%
4.5**D**(**M**)	4.18%	100.00%	100.00%
5.0**D**(**M**)	3.15%	100.00%	100.00%

$$\begin{bmatrix}
1.0000 & 0 & 0.9023 & 0.9780 & 0.8342 & 0.7999 & 0.9999 & 0.8753 & 0.0146 \\
0 & 0 & 0.7294 & 0.7785 & 0.2308 & 0.2362 & 0 & 0 & 0 \\
0 & 0 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0.2757 & 0.2653 & 0.0003 \\
0 & 0 & 0 & 1.0000 & 1.0000 & 1.0000 & 0.7067 & 0.0662 & 0.0014 \\
0 & 0 & 0 & 0 & 0.9948 & 0.9948 & 0.5847 & 0.0098 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.9948 & 0.4741 & 0.0017 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0011 & 0.0041 & 0.0010 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0004 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000
\end{bmatrix}$$

Fig. 6. Average detection-result matrix in detecting a Neptune attack.

C. Second-Order Features of Detected Flooding Attacks

We use a binary image to virtually represent the average detection-result matrix for each attack. In order to demonstrate how the average detection-result matrix is represented by a binary image, the average detection-result matrix for a Neptune attack is selected as an example. In detecting a Neptune attack, we obtain the following average detection-result matrix according to (12), where the threshold matrix is set to $3\mathbf{D}(\mathbf{M})$. Note the symmetry of a covariance matrix, we only present the upper triangle matrix in Fig. 6.

As we know, each nonzero element in the average detection-result matrix presents the confidence with that we can draw a conclusion that this position marks the second-order statistical feature for the detected flooding attack. Therefore, the value 0.9023 at position (1, 3) in Fig. 6 indicates that we have 90.23% confidence to believe that the covariance between feature 1 and feature 3 changes significantly in the Neptune attack in comparison with the norm profile in the detection. The maximum value 1 at position (3, 5) in Fig. 6 indicates that we have 100% confidence to believe that the covariance between feature 3 and feature 5 is significantly different from the norm profile in the detection, more definitely that the element at (3, 5) in

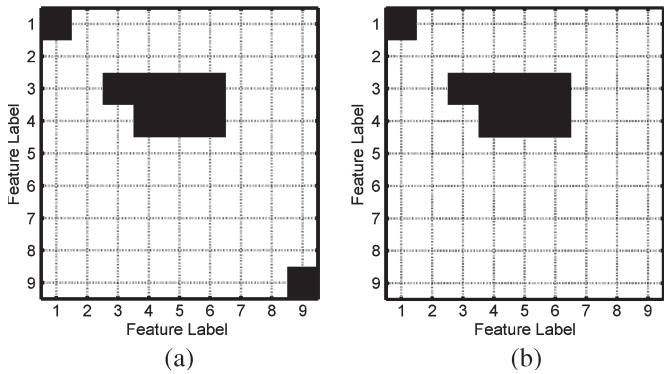

Fig. 7. Second-order features extracted for Neptune attack. (a) $3\mathbf{D}(\mathbf{M})$ as threshold matrix. (b) $4\mathbf{D}(\mathbf{M})$ as threshold matrix.

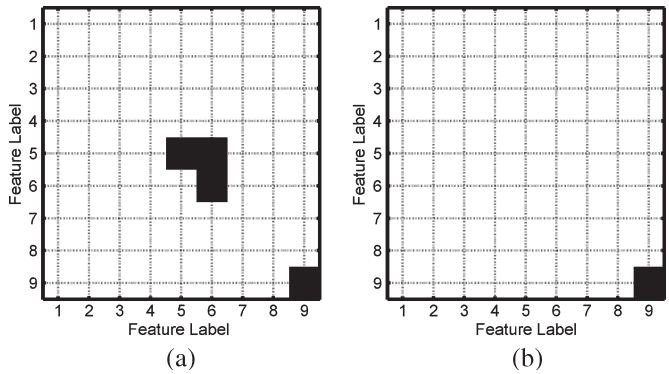

Fig. 8. Second-order features extracted for Smurf attack. (a) $3\mathbf{D}(\mathbf{M})$ as threshold matrix. (b) $4\mathbf{D}(\mathbf{M})$ as threshold matrix.

the average result matrix equal to one enables us to draw a conclusion that the covariance between feature 3 and feature 5 will always significantly deviate from the norm profile in the detection of the Neptune attack. The appearance of ones in the average result matrix also indicates the stabilities of these positions in the detection. That is, for each testing covariance matrix from a particular class, the value of elements at those positions in the 0–1 result matrix will always equal one in the detection. Therefore, we can utilize the nondiagonal positions of (3, 4), (3, 5), (3, 6), (4, 5), and (4, 6) and diagonal positions of (1, 1), (3, 3), (4, 4), and (9, 9) to indicate the second-order features for the Neptune attack. A nondiagonal element with value one indicates that the covariance between two first-order features has significantly changed, while a diagonal element with value one indicates that the variance of the first-order feature itself has changed greatly. For example, that the element at (1, 1) equal to one in Fig. 6 means that the variance of feature *count* (the number of connections to the same host as the current connection in the past 2 s) changes greatly in the Neptune attack in comparison with the norm profile. Similarly, that the element at (3, 5) in Fig. 6 equal to one means the covariance between feature *serror_rate* (the percentage of connections that have "SYN" errors to the same host) and feature *rerror_rate* (the percentage of connections that have "REJ" errors to the same host) changes greatly in the Neptune attack.

To virtually present the second-order features, we use a simple binary image to represent an average detection-result matrix. The elements with value ones in the average detection-result matrix are drawn in gray color, while other elements are in white color. Therefore, the average detection-result matrix of the Neptune attack in Fig. 6 will be represented as the image in Fig. 7(a), where the gray colored squares in the image exhibit the extracted second-order features for a Neptune attack with 100% confidence during detection. Figs. 7 and 8 present the results of extracted second-order features for the Neptune and Smurf attacks under different threshold matrices in our case studies, respectively.

A summary of the second-order features extracted for Neptune and Smurf attacks under different threshold matrices is given in Table V.

Table V shows the set of second-order features for different attacks under different threshold matrices. For example, when the threshold matrix is set to $4\mathbf{D}(\mathbf{M})$, the second-order features

TABLE V
LISTS OF THE SECOND-ORDER FEATURES OF NEPTUNE AND SMURF ATTACKS

Threshold Matrix	Positions with value 1 in the detection result matrix	
	Detecting Neptune Attack	Detecting Smurf Attack
$3\mathbf{D}(\mathbf{M})$	(1,1),(3,3),(4,4),(9,9) (3,4),(3,5),(3,6),(4,5),(4,6)	(5,5),(6,6),(9,9); (5,6)
$4\mathbf{D}(\mathbf{M})$	(1,1),(3,3),(4,4) (3,4),(3,5),(3,6),(4,5),(4,6)	(9,9)

of the Neptune attack are the variances of feature one, feature 3 and feature 4, and the covariance between feature pairs of (3,4), (3,5), (3,6), (4,5), and (4,6). Similarly, the second-order features of the Smurf attack are the variance of feature 9, which means that the variance of feature 9 has changed greatly during the Smurf attack in comparison with the norm profile. The corresponding detection rate of the normal class is 93.63% in experiments (Table III) and 93.75% in statistical theory (14), while the detection rates of both attacks are 100% with $4\mathbf{D}(\mathbf{M})$ as the threshold matrix.

For the same column in Table V, the number of positions with value ones decreases as the value of threshold matrix increases, which is consistent with the principle of threshold-based detection: if the classification boundary becomes wider, more points will be classified as normal.

D. Discussions

In our case studies, the high detection rates for the two common DDoS flooding attacks validate the effectiveness of the covariance-matrix-based approach in detecting the unknown flooding attacks, since no prior knowledge of these attacks is provided in the training stage. Moreover, the results also show that the detection approach can extract a new important feature set—the second-order feature set to mark each of the detected flooding attacks.

Two major reasons contribute to the high performance achieved by the detection approach we developed in the validations. One is the dataset itself. We should notice that the detection results would vary with different datasets and different feature sets. The features we used in the validations are

effective enough in detecting the flooding attacks in the KDD CUP dataset, but will not be sufficient enough to detect all the flooding attacks in the Internet. The other is the threshold determination algorithm. As we know, any threshold determination will face a tradeoff between the beta errors. In this paper, the principal consideration of settling the threshold matrix is the requirement of decreasing the false alarm rate as much as possible. According to (9), a lower false alarm rate will correspond to a higher detection rate of normal traffic. Therefore, the Chebyshev-inequality-based threshold determination algorithm provides a suitable threshold matrix solution constrained by the false alarm rates. Our final objective is to obtain an adaptive threshold, which can be learned from examples. The Chebyshev inequality provides a loose-bound solution, but so far, we have not yet found a better inequality to set the threshold. It may remain to be studied further.

Since the detection approach we developed mainly utilizes the covariance changes in the detection, one challenge would be that the detection approach will not work under the situation where an attack linearly changes all monitored features. We can analyze this challenge as follows. Let $\mathbf{M}^x$ denote the covariance matrix constructed from the normal samples $\mathbf{x} = (f_1, f_2, \ldots, f_p)^{\mathrm{T}}$. Let $\mathbf{M}^z$ denote the covariance matrix constructed from the linearly changed attack samples $\mathbf{z} = (a_1 + b_1 f_1, a_2 + b_2 f_2, \ldots, a_p + b_p f_p)^{\mathrm{T}}$. Essentially, the challenge can be restated as: the detection approach which mainly utilizes the differences among covariance matrices will not work, since the statement $\mathbf{M}^x = \mathbf{M}^z$ is true. In fact, the statement $\mathbf{M}^x = \mathbf{M}^z$ will be true, if and only if $\mathbf{cov}(f_i, f_j) = b_i b_j \mathbf{cov}(f_i, f_j)$, where i and j are integers which satisfy $\forall i,\ \forall j,\ 1 \le i,\ j \le p$. Obviously, we can obtain $b_i b_i = 1$, $i = 1, 2, \ldots, p$. Thus, $b_i = \pm 1$, $i = 1, 2, \ldots, p$. Since $\exists i,\ \exists j$, $\mathbf{cov}(f_i, f_j) \neq 0$, we will obtain $b_i b_j = 1$. Therefore, in order to satisfy $\mathbf{M}^x = \mathbf{M}^z$, we must have either $b_i = 1$ or $b_i = -1$, $\forall i$. In practice, it hardly happens that all p features have and only have the shift transformation. Thus, it is impossible to satisfy $\forall i,\ b_i = -1$, neither does $\forall i,\ b_i = 1$. Therefore, the covariance matrix will also change in the situation where a flooding attack linearly changes all monitored features. The covariance-matrix-based anomaly detection model can still work to detect such kind of attacks.

In this paper, we utilize the second-order statistics—one of numerical characteristics of a random variable to describe the normal traffic. It would be more accurate if we could use the probability density function (pdf) or joint pdfs to completely describe the normal traffic. However, in practice, it is very difficult to obtain the real pdf of the normal network traffic. For example, some work in the last few years claims that the network traffic can be modeled as a long-range-dependent or Self-Similar (SS) stationary process. However, recent evidence shows that the Internet traffic cannot be characterized by only a single Hurst parameter in the SS because it is extremely nonstationary [21], [22]. Therefore, there is a need for more approaches and models to discover the real nature of the normal network traffic. Utilizing the numerical characteristics, such as utilizing the second-order statistics proposed in this paper, will be an alternative approach to possibly reflect the statistical characteristics of the normal traffic.

VII. Practical Implementation Issues

A. Role and Placement of the Detector

The features we used in our validation are all network-based, rather than the host-based features. Therefore, as has been done in most network intrusion detection systems, it is possible that our network anomaly detector works in a bypass monitoring mode. The covariance-matrix-based detector proposed in this paper can be used to protect any server in a local network. To capture the traffic of the victim inside a local network and detect flooding attacks, the detector should be installed in the same local network where the victim is located. In a 100/10-M local/stub network, the network connection statistics of the victim can be captured and calculated by utilizing some online-traffic-capturing software such as Sniffer or TCPDUMP, with the presettled sequential sampling rules such as making a statistical calculation after collecting a fixed number of packets. However, in an over 1000-M/G core network, when the bandwidth is near saturation, the software such as TCPDUMP has to drop the packets at a very high rate. In this circumstance, we should make use of high-performance capturing hardware such as NP for packet capturing and state analysis.

In order to decrease the traffic that the detector machine needs to analyze, we suggest linking the detector machine with a mirror interface of the router that classifies and forwards the traffic to the victim. By setting up the mirror rules, the detector machine can only receive the network traffic of the monitored host rather than the whole traffic of the local network. We can even utilize the existing network statistical monitoring system running on the routers to forward the statistics of the sequential samples to the detection system by building up special capturing rules. For example, the detector can directly receive the needed statistical samples from the commercial product Netflow running on Cisco routers.

With the development of large-scale packet classifying techniques [26], the network-based detector's capability to withstand any flooding attacks will mainly depend on the ability of a local router to classify and forward packets [12]. During a flooding attack, the arrival time between the adjacent anonymous packets is very short. However, at one particular time, there will be at most one packet transported on the communication channel, since almost all local networks are broadcast network based on multiaccess channel with Carrier Sense Multiple Access/Collision Detect protocol on the medium access-control sublayer. Under the condition that the router can forward the packet properly, the passive sequential sampling of the victim traffic will work, even as the requests cannot be accepted by the victim machine.

B. Performance Improvements

The detection model we developed has a number of weaknesses. We are investigating some possible modifications that would likely to rectify the weaknesses and improve the performance of the detection model.

Online Training: In this paper, we utilize the offline normal traffic data provided by a public dataset to obtain the norm profile. The obtained profile is somewhat simple and will not

be responsive enough to adapt to the rapidly evolving network traffic. Thus, an online training extension to our present model can greatly reduce the impact of traffic behavioral changes.

Feature Set Engineering: The features we used in this paper are the network-connection-based features. The detection model needs to keep the connection status, which limits itself to the stub networks. There are many other network feature sets available and could be utilized to improve the performance. We suggest utilizing the stateless network features, such as the TCP flags [9] and MIB variables [36], which will enable the detection model to be settled into any core network. Furthermore, the application of different feature sets to our detection model will help to understand the nature of network traffic from a viewpoint of its second-order statistics.

Sequence Length Determination: The complexity of presented detection model is $O(np^2)$, where n is the sequence length and p is the number of features. The size of n will affect the time spent on the covariance-matrix modeling. In this paper, we use an experimental way to determine the parameter n in the stage of training. Although the time spent on training will not affect the detection speed, it is still a limitation of not providing a more effective way to determine the sequence length.

Alternative Dissimilarity Measures: The dissimilarity measure we proposed in this paper is relatively coarse. It equally evaluates all of the dimensions of two compared covariance matrices. Using a different measure, such as assigning different weights to different dimensions and including *a priori* domain knowledge, will improve the performance at the cost of decreasing generality of the approach.

Detection Delay Decrease: The detection delay is another important factor to exhibit the effectiveness of the detection model in addition to the detection accuracy. The detection delay in our detection model relates with the time of the sampling interval and the time spent on covariance-matrix modeling. The covariance-matrix modeling needs a total of np^2 calculations. The time spent on the calculations will be fixed after determining the parameters n and p. However, the time spent on sampling can be reduced. In this paper, we utilize the nonoverlapped sequences of length n in the covariance-matrix modeling. The corresponding detection delay is nt_s, if the sampling interval is t_s. An improvement to shorten the detection delay is to employ the sliding sequences instead of nonoverlapped sequences in the covariance-matrix modeling process. If each sequence of length n is obtained by sliding $m(m \ll n)$ samples once a time, the detection delay of will be shortened within the range of $[mt_s, nt_s]$.

VIII. Conclusion

This paper presents a general and effective modeling and anomaly detection approach to detecting various flooding attacks. The modeling process utilizes the correlations among network features provided by the passive-network-monitoring devices. In conceptual terms, this modeling process formulates a sequence of original samples provided by the passive-monitoring devices or software into a point in the covariance feature space. The detection approach itself directly works with the covariance matrices and maps the point represented by each covariance matrix in the covariance feature space into the decision space as normal or a specific flooding attack. The classification boundaries for the norm profile are defined by the elements in a threshold matrix. Each element in the threshold matrix evaluates the degree to which an observed covariance matrix is different from the norm profile in terms of the correlations of the monitored first-order features represented by this element. Significant difference is determined if an element in the difference matrix between the observed covariance matrix and the norm profile is bigger than the corresponding element in the threshold matrix. One appears at the position in the result matrix, where significant difference of the corresponding correlation is determined; otherwise, zero appears to represent no significant difference. Since the appearance of one in the result matrix indicates that the observed covariance matrix is significantly different from the norm profile and different number and positions of ones in the result matrix indicates the different types of flooding attacks, the result matrix serves as the second-order features to mark the detected flooding attack.

As case studies, we validate the effectiveness of the covariance-matrix modeling and detection approach to detect Neptune and Smurf attacks—two common DDoS flooding attacks. The approach can accurately differentiate these two unknown attacks. We also demonstrate that the system extracts these two attacks' own second-order features based on the detection-result matrix, respectively. By employing the Chebyshev inequality, we provide a practical method to determine the threshold matrix.

In summary, the accurate detection of two typical DDoS flooding attacks and the second-order features extracted by the detection approach will further improve our understanding of various flooding attacks. As an anomaly detection approach, which directly utilizes correlation information among the first-order features, the covariance-matrix modeling and detection approach will have wide applications in the fields such as signal detection or high-order feature extractions.

Acknowledgment

The authors thank H. Duan and T. Q. Anh (CERNET Computer Emergency Response Team in Network Research Center, TsingHua University) and our reviewers for their helpful comments.

References

[1] R. K. C. Chang, "Defending against flooding based distributed denial-of-service attacks: A tutorial," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 42–51, Oct. 2002.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[3] D. Anderson, T. Frivold, and A. Valdes, "Next-generation intrusion detection expert system (NIDES): A summary," SRI Int., Menlo Park, CA, Tech. Rep. SRI-CSL-97-07, 1995.

[4] H. S. Javitz and A. Valdes, "The NIDES statistical component description of justification," SRI Int., Menlo Park, CA, Tech. Rep. A010, 1994.

[5] P. Neumann and P. Porras, "Experience with EMERALD to date," in *Proc. 1st USENIX Workshop Intrusion Detect. and Netw. Monit.*, 1999, pp. 73–80.

[6] N. Ye, S. M. Emran, Q. Chen, and S. Vilbert, "Multivariate statistical analysis of audit trails for host-based intrusion detection," *IEEE Trans. Comput.*, vol. 51, no. 7, pp. 810–820, Jul. 2002.

[7] L. Feinstein and D. Schnackenberg, "Statistical approaches to DDoS attack detection and response," in *Proc. DISCEX*, Apr. 2003, vol. 1, pp. 303–314.
[8] C. Manikopoulos and S. Papavassiliou, "Network intrusion and fault detection: A statistical anomaly approach," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 76–82, Oct. 2002.
[9] S. Jin and D. Yeung, "A covariance analysis model for DDoS attack detection," in *Proc. IEEE ICC*, Jun. 2004, vol. 4, pp. 20–24.
[10] R. B. Blazek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods," in *Proc. Workshop Stat. and Mach. Learn. Tech. Comput. Intrusion Detect.*, Jun. 2001, pp. 220–226.
[11] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.
[12] H. Wang, D. Zhang, and K. G. Shin, "Change-point monitoring for the detection of DoS attacks," *IEEE Trans. Dependable Secur. Comput.*, vol. 1, no. 4, pp. 193–208, Oct.–Dec. 2004.
[13] W. Lee and S. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 227–261, Nov. 2000.
[14] J. Xu, "Sustaining availability of Web services under severe denial of service attacks," Georgia Inst. Technol., Atlanta, GA, Tech. Rep. GIT-CC-01-10, May 2001.
[15] W. Lee, "A data mining framework for constructing features and models for intrusion detection systems," Ph.D. dissertation, Columbia Univ., New York, 1999.
[16] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection," in *Proc. RAID*, Oct. 2003, pp. 220–237.
[17] P. Uppuluri and R. Sekar, "Experiences with specification-based intrusion detection," in *Proc. RAID*, Oct. 2001, pp. 172–189.
[18] C. Tseng, P. Balasubramanyam, and C. Ko, "A specification-based intrusion detection system for AODV," in *Proc. 1st ACM Workshop Secur. Ad Hoc and Sensor Netw. Fairfax*, Oct. 2003, pp. 125–134.
[19] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas.*, Nov. 2002, pp. 71–82.
[20] D. Moore, G. Voelker, and S. Savage, "Inferring internet denial of service activity," in *Proc. USENIX Secur. Symp.*, Aug. 2001, pp. 9–22.
[21] J. A. Gubner, "Theorems and fallacies in the theory of long-range-dependent processes," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 1234–1239, Mar. 2005.
[22] T. Karagiannis, M. Molle, and M. Faloutsos, "Long-range dependence ten years of Internet traffic modelling," *IEEE Internet Comput.*, vol. 8, no. 5, pp. 57–64, Sep./Oct. 2004.
[23] J. Ryan, M. J. Lin, and R. Miikkulainen, "Intrusion detection with neural networks," in *Advances in Neural Information Processing*. Cambridge, MA: MIT Press, 1998.
[24] Y. Xiong, S. Liu, and P. Sun, "On the defense of the distributed denial of service attacks: An on-off feedback control approach," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 31, no. 4, pp. 282–293, Jul. 2001.
[25] J. Kong, M. Mirza, J. Shu, C. Yoedhana, M. Gerla, and S. Lu, "Random flow network modeling and simulations for DdoS attack mitigation," in *Proc. IEEE ICC*, May 2003, vol. 1, pp. 487–491.
[26] T. V. Lakshman and D. Stiliadis, "High speed policy-based packet forwarding using efficient multi-dimensional range matching," in *Proc. ACM SIGCOMM*, Sep. 1998, pp. 203–214.
[27] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites," in *Proc. 11th Int. World Wide Web Conf.*, Honolulu, HI, May 2002, pp. 252–262.
[28] Y. Ohsita, S. Ata, and M. Murata, "Detecting distributed denial-of-service attacks by analyzing TCP SYN packets statistically," in *Proc. IEEE GLOBECOM*, Nov./Dec. 2004, vol. 4, pp. 2043–2049.
[29] D. Y. Yeung and Y. X. Ding, "Host-based intrusion detection using dynamic and static behavioral models," *Pattern Recognit.*, vol. 36, no. 1, pp. 229–243, Jan. 2003.
[30] W. Lee, S. Stolfo, P. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, and J. Zhang, "Real time data mining-based intrusion detection," in *Proc. DISCEX II*, Jun. 2001, pp. 85–100.
[31] J. Toelle and O. Niggenmann, "Supporting intrusion detection by graph clustering and graph drawing," in *Proc. 3rd Int. Workshop RAID*, Oct. 2000.
[32] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," in *Proc. IEEE Comput. Soc. Symp. Res. Comput. Secur. and Privacy*, 1992, pp. 240–250.
[33] *DARPA Intrusion Detection Evaluation Documentation*. (1999). [Online]. Available: http://www.ll.mit.edu/IST/ideval/docs/docs_index.html
[34] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, *2004 CSI/FBI Computer Crime and Security Survey*, 2004, San Francisco, CA: Comput. Secur. Inst. (CSI). [Online]. Available: http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2004.pdf
[35] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Chebychef's Inequality*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1988, ch. 2.17 and 5.8, pp. 43–45. 123.
[36] X. Qin, W. Lee, L. Lewis, and J. B. D. Cabrera, "Using MIB II variables for network anomaly detection—A feasibility study," in *Proc. ACM Workshop Data Mining Secur. Appl.*, Philadelphia, PA, Nov. 2001, pp. 609–622.

Daniel S. Yeung (M'89–SM'99–F'04) received the Ph.D. degree in applied mathematics from Case Western Reserve University, Cleveland, OH, in 1974.

He is currently a Chair Professor at the Department of Computing, Hong Kong Polytechnic University, Kowloon. His current research interests include neural-network sensitivity analysis, data mining, Chinese computing, and fuzzy systems.

Dr. Yeung was the President of the IEEE Hong Kong Computer Chapter. He is a member of the Board of Governors for the IEEE SMC Society, and has been elected the Vice President for Technical Activities for the same society. He has served as the General Co-Chair of the 2002–2004 International Conference on Machine Learning and Cybernetics held annually in China, and a keynote speaker for the same conference. He is currently an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS and IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B.

Shuyuan Jin (M'04) received the B.Sc. and M.Sc. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1996 and 1998, respectively, and the Ph.D. degree from the Hong Kong Polytechnic University, Kowloon, in 2006.

Her main research interests include network security, especially intrusion detection and responses, Internet technologies, machine learning, and pattern recognition techniques and applications.

Xizhao Wang (M'03–A'04–SM'04) received the B.Sc. and M.Sc. degrees in mathematics from Hebei University, Baoding, China, in 1983 and 1992, respectively, and the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1998.

From 1983 to 1998, he worked as a Lecturer, an Associate Professor, and a Full Professor in the Department of Mathematics, Hebei University. Since 1998, he has been working as a Research Fellow with the Department of Computing, Hong Kong Polytechnic University, Kowloon. His main research interests include inductive learning and fuzzy representation, fuzzy measures and integrals, neuro-fuzzy systems and genetic algorithms, and feature extraction.