Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 2185–2197
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Network intrusion detection in covariance feature space

Shuyuan Jin[a,*], Daniel So Yeung[a], Xizhao Wang[b]

[a]*Department of Computing, Hong Kong Polytechnic University, P.O. Box 20, Hong Hum, Kowloon, Hong Kong*
[b]*School of Mathematics and Computer Science, Hebei University, Baoding, China*

Abstract

Detecting multiple and various network intrusions is essential to maintain the reliability of network services. The problem of network intrusion detection can be regarded as a pattern recognition problem. Traditional detection approaches neglect the correlation information contained in groups of network traffic samples which leads to their failure to improve the detection effectiveness. This paper directly utilizes the covariance matrices of sequential samples to detect multiple network attacks. It constructs a covariance feature space where the correlation differences among sequential samples are evaluated. Two statistical supervised learning approaches are compared: a proposed threshold based detection approach and a traditional decision tree approach. Experimental results show that both achieve high performance in distinguishing multiple known attacks while the threshold based detection approach offers an advantage of identifying unknown attacks. It is also pointed out that utilizing statistical information in groups of samples, especially utilizing the covariance information, will benefit the detection effectiveness.

Keywords: Covariance feature space; Threshold based detection; Decision tree; Network intrusion detection; Detection effectiveness

1. Introduction

The fact that various attacks are increasing at a surprising rate indicates the importance of detection systems. It is reported that there have been 10,000 new viruses or variants of existing viruses in the year of 2004 and there is at least one new attack spotted every hour [1]. It is highly demanded for the detection tools to effectively distinguish multiple known and unknown attacks.

A number of intrusion detection techniques have been proposed in the literature to maintain the reliability of networks. They can be largely classified into two categories: misuse detection and anomaly detection. Misuse detection techniques signal intrusions when the observed activities in an information system match the pre-built rules or signatures of known intrusions. Anomaly detection techniques determine intrusions when the subject's observed behaviors exhibit a significant variation from its norm profile. Compared with misuse detection, anomaly detection techniques offer an advantage of identifying unknown attacks [2].

As a main problem, the capability of distinguishing multiple known attacks as well as unknown attacks directly manifests the effectiveness of an intrusion detection system. This problem can be regarded as a multi-class classification problem in pattern recognition. The applications of the pattern recognition approaches in intrusion detection can be found in the examples such as artificial neural network [3–5], support vector machine [6,7], nearest neighbor rules [8], clustering [9–11], decision tree [12,13], data mining [14] and so on. The detailed surveys of pattern recognition techniques employed in intrusion detection have been published in the literature [15–18]. Besides, there are also some intrusion detection systems based on immunological approaches [4,19].

Almost all the existing pattern recognition approaches in intrusion detection utilize the differences of the first-order statistics to classify different intrusions. For example, Ye et al. utilize T^2 and χ^2 test values of the system events to evaluate the difference between the observed audit events and the mean of the normal audit events in order to detect intrusions [20]. A data mining model proposed by Lee makes use of the first-order features such as *service*, *flag*, *duration*, etc. to compute the

* Corresponding author. Tel.: +852 2766 7281; fax: +852 2774 0842.
E-mail addresses: jinshuyuan@yahoo.com.cn (S. Jin), csdaniel@comp.polyu.edu.hk (D.S. Yeung), wangxz@mail.hbu.edu.cn (X. Wang).

0031-3203/$30.00
doi:10.1016/j.patcog.2006.12.010

association rules and frequent episodes for intrusion detection [14]. The inputs of the neural network based detection approaches are the first-order features such as the distribution of 100 frequently used commands in NNID (Neural Network Intrusion Detector) [3]. In a hidden Markov model [21], the frequency of each command is utilized as the basic probability of each state.

However, "some intrusion detection techniques intrinsically ignore the coherent relations and dependencies of the features which often result in high false positive rate" [22]. On the other hand, many intrusions in information systems manifest themselves by the correlation changes [23,24]. Therefore, this paper utilizes the covariance matrices to discover the effects of relations and dependencies among features on distinguishing multiple attacks. It mainly focuses on the discussion of the utilization of covariance matrix—the second-order statistics in the intrusion detection systems in terms of effectiveness, that is, the detector should "detect a substantial percentage of known and unknown intrusions, while still keeping the false alarm rate at an acceptable level" [25]. The main contributions of this paper are:

- It utilizes statistical covariance matrix of sequential samples in the problem of multiple classification. Different from the PCA (Principal Component Analysis) or uncorrelated LDA (Linear Discriminant Analysis) approaches where the correlations among features are eliminated before classification; the classifiers in this paper directly utilize the correlations during classification. Classifying multiple classes by means of covariance matrix differences among sample groups is a contribution to pattern recognition.
- It proposes a threshold based algorithm to utilize the covariance matrices of sequential samples in the detection. By measuring the covariance differences along each dimension of the covariance feature space under the predefined threshold matrix, the approach provides a new tool to classify multiple classes in the covariance feature space.
- It investigates the effectiveness of intrusion detection in covariance feature space. It analyzes the performance improvement by utilizing the covariance information in groups of samples in the detection and compares the performance of two typical statistical supervised detection approaches.

The rest of the paper is organized as follows. Section 2 provides the background of the paper. It introduces the behavior of typical network intrusions—DoS (Denial-of-Service) attacks and explains the benefits of the covariance based detection against the traditional mean based detection approaches. Section 3 defines the problem of intrusion detection. It describes the covariance feature space and gives the problem representation in details. Section 4 presents the threshold based detection approach and decision tree approach. Section 5 presents and compares the experimental results of two detection approaches in details. It also discusses how to determine a sequence length in practice. Section 6 draws a conclusion.

2. Background

In this paper, we mainly discuss the utilization of covariance matrix in the intrusion detection systems. As the second-order statistics, a covariance matrix holds two types of information: one is the information contained in a group of samples; the other is the correlation information among the observed features. In this section, we will explain the benefits of utilizing the covariance based detection from two aspects: one is the performance improvement by utilizing groups of samples in the detection; the other is the advantage of effectively distinguishing different classes in the case where the mean based detection approaches fail.

The network traffic can be characterized in terms of sequences of discrete data with temporal dependency [8,23,24]. When we segment the observed temporal sequences into different and consecutive time fragments or intervals, we will obtain groups of samples. Each time interval corresponds to each group of samples. Our basic idea is to label such groups of samples.

In a typical DoS attack, many slaves send packets to the victim at about the same time that last for a period of time under the control of a master. It is clear that the groups of samples within that particular period should be labeled as a particular DoS attack rather than the normal class. Statistically speaking, the network exhibits the normal behavior in most of operation time. Therefore, there are more chances to label groups of samples as the normal class rather than any abnormal class.

In order to explain the basic principle clearly, we introduce the following simple mathematical examples.

Assume the network samples are i.i.d. Given two random populations $X_{(l)} \sim N(\mu_{(l)}, \sigma^2_{(l)})$, $l = 1, 2$, such as $\mu_{(1)} \leqslant \mu_{(2)}$, which represent two different classes—normal and abnormal classes, respectively. The purpose is to find which population that a group of n samples belong to, if these n samples x_t, $t = 1, 2, \ldots, n$ come from one population independently.

To solve the above problem, traditional mean based detection approaches normally label a group of n samples in a way of sample by sample; while the covariance based detection approach proposed in this paper utilizes the statistical information contained in groups of samples to classify such a group of n samples.

2.1. Detection by samples

Normally speaking, the mean based detection approaches will employ a distance based classifier to label a sample y according to the following rules:

$$\begin{aligned} &\text{if } d(y, X_{(1)}) \leqslant d(y, X_{(2)}), \quad y \in X_{(1)}, \\ &\text{if } d(y, X_{(2)}) < d(y, X_{(1)}), \quad y \in X_{(2)}, \end{aligned} \tag{1}$$

where

$$d(y, X_{(l)}) = |y - \mu_{(l)}|/\sigma_{(l)} \quad (l = 1, 2). \tag{2}$$

Let

$$\mu^* = \mu_{(1)} \cdot \frac{\sigma_{(2)}}{\sigma_{(1)} + \sigma_{(2)}} + \mu_{(2)} \cdot \frac{\sigma_{(1)}}{\sigma_{(1)} + \sigma_{(2)}}. \tag{3}$$

We will obtain

$$\begin{aligned} &\text{if } y \leqslant \mu^* \quad \Leftrightarrow \quad d(y, X_{(1)}) \leqslant d(y, X_{(2)}), \quad y \in X_{(1)}, \\ &\text{if } y > \mu^* \quad \Leftrightarrow \quad d(y, X_{(2)}) < d(y, X_{(1)}), \quad y \in X_{(2)}. \end{aligned} \tag{4}$$

The probabilities of correctly classifying a sample into its population will be

$$\begin{cases} p_1 = \displaystyle\int_{-\infty}^{\mu^*} (1/(\sqrt{2\pi}\sigma_{(1)})) \exp\left[-\frac{1}{2}\left(\frac{x-\mu_{(1)}}{\sigma_{(1)}}\right)^2\right] \mathrm{d}x, \\ p_2 = \displaystyle\int_{\mu^*}^{+\infty} (1/(\sqrt{2\pi}\sigma_{(2)})) \exp\left[-\frac{1}{2}\left(\frac{x-\mu_{(2)}}{\sigma_{(2)}}\right)^2\right] \mathrm{d}x, \end{cases} \tag{5}$$

where p_1 is the probability that a sample which comes from the population $X_{(1)}$ is correctly classified into $X_{(1)}$. p_2 is the probability that a sample which comes from the population $X_{(2)}$ is correctly classified into $X_{(2)}$. Clearly $p_1 = p_2$.

The probabilities of incorrectly classifying a sample into its population will be:

$$\begin{cases} f_1 = \displaystyle\int_{(\mu_{(2)}-\mu_{(1)})/(\sigma_{(1)}+\sigma_{(2)})}^{+\infty} (1/\sqrt{2\pi}) \exp\left(-\frac{1}{2}x^2\right) \mathrm{d}x, \\ f_2 = \displaystyle\int_{-\infty}^{(\mu_{(1)}-\mu_{(2)})/(\sigma_{(1)}+\sigma_{(2)})} (1/\sqrt{2\pi}) \exp\left(-\frac{1}{2}x^2\right) \mathrm{d}x, \end{cases} \tag{6}$$

where f_1 is the probability that a sample which comes from the population $X_{(1)}$ is wrongly classified into the population $X_{(2)}$. f_2 is the probability that a sample which comes from the population $X_{(2)}$ is wrongly classified into the population $X_{(1)}$. Clearly $f_1 = f_2$.

In order to obtain the probability of correctly classifying a group of n samples, we introduce a random variable B_k. Let $B_k = 1$ represent x_k is correctly classified while $B_k = 0$ represent x_k is wrongly classified ($k = 1, 2, \ldots, n$). We define a binomial random variable $W = \sum_{k=1}^{n} B_k$.

The detection precision rate that m samples are correctly classified will thus be

$$\mathbf{P}(W = m) = C_n^m p^m (1-p)^{n-m}, \tag{7}$$

where p is the probability of correctly labeling a sample and $m = 0, 1, 2, \ldots, n$.

According to Eqs. (5)–(7), we can calculate the detection precision rate of the traditional classification approaches. For example, if two populations are normally distributed as

$$\begin{cases} \mu_{(1)} = 0, \\ \sigma_{(1)} = 12, \end{cases} \quad \begin{cases} \mu_{(2)} = 10, \\ \sigma_{(2)} = 18. \end{cases} \tag{8}$$

According to Eqs. (5) and (6) (where $\mu^* = 4$), by looking into the standard normal distribution table, we can obtain that the probability of correctly classifying a sample is $p_1 = p_2 = 0.63$. The probability of wrongly classifying a sample is $f_1 = f_2 = 0.37$. Therefore, the precision rate of correctly classifying a group of n samples will be $\mathbf{P}(W = n) = p_1^n = 0.63^n$. The error rate of incorrectly classifying a group of n samples will be $\mathbf{P}(W = 0) = (1 - p_1)^n = 0.37^n$.

Note that

$$\begin{cases} \mathbf{E}(W) = np_1 = 0.63n, \\ \mathbf{D}(W) = np_1(1-p_1) = 0.37 \times 0.63n, \end{cases} \tag{9}$$

where $\mathbf{E}(W)$ and $\mathbf{D}(W)$ are the expected value and the variance of the binomial random variable W, respectively. Eq. (9) explains that approximately $0.63n$ samples will be correctly classified while others will be incorrectly classified. We can hardly improve the detection precision rate even we know that a group of n sequential samples come from the same population.

2.2. Detection by groups

The proposed covariance matrix based approach utilizes the statistical information in groups of samples. Simply speaking, for a group detection method to solving the above problem, we can firstly define a random variable

$$z = \frac{1}{n}\sum_{t=1}^{n} x_t, \quad x_t \in X_{(l)}, \quad t = 1, 2, \ldots, n.$$

Obviously,

$$Z_{(l)} = \frac{1}{n}\sum_{t=1}^{n} x_t \sim N\left(\mu_{(l)}, \frac{1}{n}\sigma_{(l)}^2\right), \quad l = 1, 2.$$

Then, we classify a group of samples represented by z according to the following rules:

$$\begin{aligned} &\text{if } z \leqslant v^*, \quad z \in X_{(1)}, \\ &\text{if } z > v^*, \quad z \in X_{(2)}, \end{aligned} \tag{10}$$

where

$$v^* = \mu_{(1)} \cdot \frac{\sigma_{(2)}}{\sigma_{(1)} + \sigma_{(2)}} + \mu_{(2)} \cdot \frac{\sigma_{(1)}}{\sigma_{(1)} + \sigma_{(2)}}. \tag{11}$$

The detection precision rates will thus be

$$\begin{cases} q_1 = \displaystyle\int_{-\infty}^{v^*} \left(1 \Big/ \left(\sqrt{2\pi}\frac{1}{\sqrt{n}}\sigma_{(1)}\right)\right) \\ \qquad \times \exp\left[-\frac{1}{2}\left(\frac{x-\mu_{(1)}}{1/\sqrt{n}\sigma_{(1)}}\right)^2\right] \mathrm{d}x, \\ q_2 = \displaystyle\int_{v^*}^{+\infty} \left(1 \Big/ \left(\sqrt{2\pi}\frac{1}{\sqrt{n}}\sigma_{(2)}\right)\right) \\ \qquad \times \exp\left[-\frac{1}{2}\left(\frac{x-\mu_{(2)}}{1/\sqrt{n}\sigma_{(2)}}\right)^2\right] \mathrm{d}x, \end{cases} \tag{12}$$

where q_1 is the probability that a group of n samples which come from the population $X_{(1)}$ are correctly classified into $X_{(1)}$. q_2 is the probability that a group of n samples which come from the population $X_{(2)}$ are correctly classified into $X_{(2)}$.

The detection error rates will be:

$$\begin{cases} r_1 = \displaystyle\int_{v^*}^{+\infty} \left(1 \Big/ \left(\sqrt{2\pi}\frac{1}{\sqrt{n}}\sigma_{(1)}\right)\right) \\ \qquad \times \exp\left[-\frac{1}{2}\left(\frac{x-\mu_{(1)}}{1/\sqrt{n}\sigma_{(1)}}\right)^2\right] \mathrm{d}x, \\ r_2 = \displaystyle\int_{-\infty}^{v^*} \left(1 \Big/ \left(\sqrt{2\pi}\frac{1}{\sqrt{n}}\sigma_{(2)}\right)\right) \\ \qquad \times \exp\left[-\frac{1}{2}\left(\frac{x-\mu_{(2)}}{1/\sqrt{n}\sigma_{(2)}}\right)^2\right] \mathrm{d}x, \end{cases} \tag{13}$$

where r_1 represents the probability that a group of n samples which come from the population $X_{(1)}$ are incorrectly classified into the population $X_{(2)}$. r_2 represents the probability that a group of n samples which come from the population $X_{(2)}$ are incorrectly classified into the population $X_{(1)}$.

It is clear that the performance is different between the sample-by-sample detection method and group detection method when we compare three Eqs. (5), (7) and (12). For the same example mentioned in Eq. (8) in Section 2.1, if $n = 100$, the precision of the group detection method will be

$$q_1 = q_2 = \int_{-\infty}^{4/(\frac{1}{10}\times 12)} \left(\frac{1}{\sqrt{2\pi}}\right) \exp\left(-\frac{1}{2}x^2\right) \mathrm{d}x = 0.9995658$$

while precision of the sample-by-sample detection method will be $\mathbf{P}(W = 100) = 0.63^{100} = 8.5912\mathrm{e}-021$; the error rate of the group detection method will be

$$r_1 = r_2 = \int_4^{+\infty} \left(\frac{1}{\frac{1}{10}\times 12\sqrt{2\pi}}\right) \exp\left[-\frac{1}{2}\left(\frac{x}{\frac{1}{10}\times 12}\right)^2\right] \mathrm{d}x$$
$$= 0.0004342$$

while the error rate of the sample-by-sample detection method will be $\mathbf{P}(W = 0) = 0.37^{100}$. If $n = 16$, the precision of the group detection method will be

$$q_1 = q_2 = \int_{-\infty}^{4/(\frac{1}{4}\times 12)} \left(\frac{1}{\sqrt{2\pi}}\right) \exp\left(-\frac{1}{2}x^2\right) \mathrm{d}x = 0.90824$$

while precision of the sample-by-sample detection method will be $\mathbf{P}(W = 16) = 0.63^{16} = 6.1581\mathrm{e}-004$.

2.3. Discussions

The comparison results show that the group detection method achieves a much higher detection rate than the traditional sample-by-sample detection method in solving the proposed problem. For the traditional sample-by-sample detection method, $\mathbf{E}(W) = 0.63n$ determines that it can only correctly classified approximately 63 samples from a total of 100 samples in the way of sample by sample. In contrast, the group detection method will correctly classify a total of 100 samples with the probability of over 99.9%.

Comparing Eqs. (5), (7) and (12), we can obtain the relation between the traditional sample-by-sample detection method and group detection method in terms of the probability of correctly classifying a group of n samples

$$q > p > \mathbf{P}(W = n), \quad \text{if } n > 1, \tag{14}$$

where q is the probability of correctly classifying a group of n samples, p is the probability of correctly classifying a sample x_t $(t = 1, 2, \ldots, n)$, and $\mathbf{P}(W = n)$ is the probability of correctly classifying a group of n samples in the way of sample by sample.

In the above comparisons, we use the mean based classification as an example to show the performance difference between the traditional sample-by-sample detection method and the group detection method. We can also use the covariance as an example to show the performance improvement by utilizing groups of samples. Its explanation will be more complex, however, the principle is the same. In fact, Eq. (15) shows that when we use the covariance matrix in the detection, the performance will also be improved when the sequence length n is greater than $2\sigma^2$.

In the following part, we will use the covariance as an example to show that the covariance matrix based detection approach provides an advantage of effectively distinguishing different classes in the case where the mean based detection approaches fail.

Let us consider such a case where the means of two populations are close to each other. For example, there are two populations $X_{(l)} \sim N(\mu_{(l)}, \sigma^2_{(l)})$, $l=1, 2$ with $\mu_{(1)}=\mu_{(2)}=0$ and $\sigma_{(1)} \neq \sigma_{(2)}$. The purpose is to determine which population a group of n samples belong to, if these n samples x_t, $t = 1, 2, \ldots, n$ are from one population independently.

To solve this problem, we cannot utilize the mean based classification approaches. Since μ^* in the determination rules (4) is equal to $\mu_{(l)}$ $(l = 1, 2)$, the probability of either correctly or incorrectly determining any sample will be 50%. The 50% precision rate has no any sense for classification.

In contrast, the covariance based detection approach will succeed in solving such a problem. In detail, we define a new variable $S_{(l)} = \sum_{t=1}^{n}(x_t/\sigma_{(l)})^2$. Since independent sample $x_t \sim N(0, \sigma^2)$, where $t = 1, 2, \ldots, n$, $S_{(l)}$ will have a $\chi^2(n)$ distribution with its mean $\mathbf{E}(S_{(l)}) = n$ and its variance $\mathbf{D}(S_{(l)}) = 2n$. Let $Y_{(l)} = (\sigma^2_{(l)}/n)S_{(l)}$, we will obtain

$$\begin{cases} \mathbf{E}(Y_{(l)}) = \mathbf{E}\left(\dfrac{\sigma^2_{(l)}}{n} S_{(l)}\right) = \sigma^2_{(l)}, \\ \mathbf{D}(Y_{(l)}) = \mathbf{D}\left(\dfrac{\sigma^2_{(l)}}{n} S_{(l)}\right) = 2n\left(\dfrac{\sigma^2_{(l)}}{n}\right)^2 = \dfrac{2\sigma^2_{(l)}}{n} \times \sigma^2_{(l)}. \end{cases} \tag{15}$$

The variance of a group of n samples is

$$y = \frac{1}{n}\sum_{t=1}^{n} x_t^2. \tag{16}$$

Clearly $s = (n/\sigma^2)y$ is a sample from the population $S \sim \chi^2(n)$. Therefore, the distance between y and the population

$X_{(l)}$, $(l = 1, 2)$ will be

$$d(y, X_{(l)}) = |y - \mathbf{E}(Y_{(l)})|/\sqrt{D(Y_{(l)})} = |y - \sigma^2_{(l)}|/(\sigma^2_{(l)}\sqrt{2/n}). \tag{17}$$

Let $\vartheta^* = 2\sigma^2_{(1)}\sigma^2_{(2)}/(\sigma^2_{(1)} + \sigma^2_{(2)})$, we obtain the following determination rules:

$$\begin{aligned} &\text{if } y \leqslant \vartheta^* \quad \Leftrightarrow \quad d(y, X_{(1)}) \leqslant d(y, X_{(2)}), \quad y \in X_{(1)}, \\ &\text{if } y > \vartheta^* \quad \Leftrightarrow \quad d(y, X_{(1)}) > d(y, X_{(2)}), \quad y \in X_{(2)}. \end{aligned} \tag{18}$$

Therefore, the precision rates of classifying a group of n samples are

$$\begin{cases} p_1 = \mathbf{P}(y \leqslant \vartheta^*) = \displaystyle\int_0^{2n\sigma^2_{(2)}/(\sigma^2_{(1)}+\sigma^2_{(2)})} \frac{1}{2^{n/2}\Gamma(n/2)} \\ \quad \times \mathrm{e}^{-s/2} s^{(n/2)-1}\,\mathrm{d}s, \\ p_2 = \mathbf{P}(y > \vartheta^*) = \displaystyle\int_{2n\sigma^2_{(1)}/(\sigma^2_{(1)}+\sigma^2_{(2)})}^{+\infty} \frac{1}{2^{n/2}\Gamma(n/2)} \\ \quad \times \mathrm{e}^{-s/2} s^{(n/2)-1}\,\mathrm{d}s, \end{cases} \tag{19}$$

where p_1 is the probability that a group of n samples from population $X_{(1)}$ are correctly classified into $X_{(1)}$, p_2 is the probability that a group of n samples from population $X_{(2)}$ are correctly classified into $X_{(2)}$.

The classification error rates are

$$\begin{cases} f_1 = 1 - p_1 = \displaystyle\int_{2n\sigma^2_{(2)}/(\sigma^2_{(1)}+\sigma^2_{(2)})}^{+\infty} \frac{1}{2^{n/2}\Gamma(n/2)} \mathrm{e}^{-s/2} s^{(n/2)-1}\,\mathrm{d}s, \\ f_2 = 1 - p_2 = \displaystyle\int_0^{2n\sigma^2_{(1)}/(\sigma^2_{(1)}+\sigma^2_{(2)})} \frac{1}{2^{n/2}\Gamma(n/2)} \mathrm{e}^{-s/2} s^{(n/2)-1}\,\mathrm{d}s, \end{cases} \tag{20}$$

where f_1 is the probability that a group of n samples from the population $X_{(1)}$ are incorrectly classified into the population $X_{(2)}$, p_2 is the probability that a group of n samples from the population $X_{(2)}$ are incorrectly classified into the population $X_{(1)}$.

Let us give an example. Assume two normally distributed populations as follows,

$$\begin{cases} \mu_{(1)} = 0, \\ \sigma_{(1)} = 2, \end{cases} \quad \begin{cases} \mu_{(2)} = 0, \\ \sigma_{(2)} = 8. \end{cases} \tag{21}$$

When $n = 20$, we can obtain that detection precision rates are p_1=0.993 and p_2=0.999, and the error rates are f_1=0.007 and $f_2 = 0.001$, respectively. However, any mean based classifier will fail since the two populations have the same means.

In the above discussions, we use the presumed parameters such as mean and variance in the calculation. These parameters can be estimated in real applications. The probability distributions in the mentioned examples are much simpler than that of real applications, however, they help us understand the benefit of utilizing the covariance information in groups of samples to improve the effectiveness of the detection.

3. Problem definition

3.1. Covariance feature space

Assume p physical features $f_1, \ldots, f_p$ are provided to describe an observation and we can obtain $\mathbf{x}^l_1, \mathbf{x}^l_2, \ldots, \mathbf{x}^l_n$ in n observations during T_l. The covariance matrix of the n samples in T_l denoted as $\mathbf{M}^l$ is given by

$$\mathbf{M}^l = \begin{pmatrix} \sigma_{f^l_1 f^l_1} & \sigma_{f^l_1 f^l_2} & \cdots & \sigma_{f^l_1 f^l_p} \\ \sigma_{f^l_2 f^l_1} & \sigma_{f^l_2 f^l_2} & \cdots & \sigma_{f^l_2 f^l_p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{f^l_p f^l_1} & \sigma_{f^l_p f^l_2} & \cdots & \sigma_{f^l_p f^l_p} \end{pmatrix}, \tag{22}$$

where

$$\sigma_{f^l_u f^l_v} = \mathbf{cov}(f^l_u, f^l_v) = \frac{1}{n}\sum_{k=1}^{n}(f^{l,k}_u - \mu_{f^l_u})(f^{l,k}_v - \mu_{f^l_v}),$$

$$\mu_{f^l_u} = \mathbf{E}(f^l_u) = \frac{1}{n}\sum_{k=1}^{n} f^{l,k}_u$$

and l is the number of time intervals, such as $T_1, T_2, \ldots, T_l, \ldots$ $1 \leqslant l \leqslant \infty$.

The covariance matrix $\mathbf{M}^l$ describes the network status during T_l by means of measuring the correlativity among the network features $f_1, \ldots, f_p$. Basically, the features $f_1, \ldots, f_p$ can be directly obtained from the network monitoring devices [26–28]. For example, the features can be *the packet number every second* or *the frequencies of the different source IP addresses usage* and so on, which can be directly recorded by the statistical model provided in current routers or switches. There are also some special features proposed by the experienced network experts. These special features can be obtained through a simple pre-processing on the statistical information provided by the monitoring devices. The examples of these features are *the number of connections to the same service* or *the percentage of connections that have "SYN" errors to the same host* [29].

In conceptual terms, a sample can be regarded as a point in the feature space. As a sample, a covariance matrix can be regarded as a point in the covariance feature space. Each dimension of the covariance feature space gives the coordinate of the point along each axis of the space. If p features $f_1, \ldots, f_p$ are utilized to describe an observation, then each covariance matrix will provide the correlation information in a total of $p^*(p + 1)/2$ measurements (because a covariance matrix is symmetric). Each measurement or each dimension of the covariance feature space gives the coordinate of the point by means of the correlation between each pair of features.

Fig. 1 illustrates that a covariance matrix M^l is viewed as a point in a $p^*(p + 1)/2$-dimensional covariance feature space. The detection or the classification itself is described as a transformation that maps the point M^l into one of the classes in a c-dimensional decision space, where c is the number of classes to be distinguished.

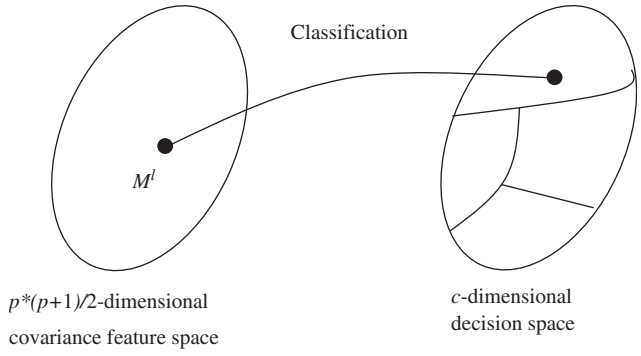

Fig. 1. Illustration of the classification in covariance feature space.

Y	$\sigma_{f_1 f_1}$	$\sigma_{f_1 f_2}$	…	…	…	$\sigma_{f_p f_p}$	Desired output
$\mathbf{y}_1$	$\sigma_{f_1^1 f_1^1}$	$\sigma_{f_1^1 f_2^1}$	…	…	…	$\sigma_{f_p^1 f_p^1}$	d_1
$\mathbf{y}_2$	$\sigma_{f_1^2 f_1^2}$	$\sigma_{f_1^2 f_2^2}$	…	…	…	$\sigma_{f_p^2 f_p^2}$	d_2
⋮						⋮	⋮
⋮						⋮	⋮
$\mathbf{y}_T$	$\sigma_{f_1^T f_1^T}$	$\sigma_{f_1^T f_2^T}$	…	…	…	$\sigma_{f_p^T f_p^T}$	d_T
$\mathbf{y}_{T+1}$	$\sigma_{f_1^{T+1} f_1^{T+1}}$	$\sigma_{f_1^{T+1} f_2^{T+1}}$				$\sigma_{f_p^{T+1} f_p^{T+1}}$	?

Fig. 2. Samples and features in $p*(p+1)/2$ dimensional covariance feature space.

3.2. Problem representation

We take a view of intrusion detection problem as a statistical multi-classification problem in pattern recognition.

Assume we are given a set of training samples $\{\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \ldots, \mathbf{x}_{1,t_1}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \ldots, \mathbf{x}_{2,t_2}, \ldots, \ldots, \mathbf{x}_{R,1}, \mathbf{x}_{R,2}, \ldots, \mathbf{x}_{R,t_R}\}$ and its corresponding set of classes $\{\omega_1, \omega_2, \ldots, \omega_R\}$, where $\{\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \ldots, \mathbf{x}_{1,t_1}\}$ belongs to class ω_1, $\{\mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \ldots, \mathbf{x}_{2,t_2}\}$ belongs to class ω_2 and $\{\mathbf{x}_{r,1}, \mathbf{x}_{r,2}, \ldots, \mathbf{x}_{r,t_r}\}$ belongs to class ω_r, t_r is the number of training samples in ω_r and $1 \leqslant r \leqslant R$. Let us use the symbol $d_r \in \{\omega_1, \omega_2, \ldots, \omega_R\}$ to represent the label of a sample $\mathbf{x}_{r,i}$ $(1 \leqslant i \leqslant t_r)$, where $\mathbf{x}_{r,i}$ having p features $f_1, \ldots, f_p$ is the ith sample in class ω_r, and d_r is the target output. In order to evaluate the correlation differences among features, we construct the covariance matrix training set $\{(\mathbf{y}_{r,i}, d_r)\}_{i=1}^{\mathrm{T}}$, where $\mathbf{y}_{r,i}$ is the covariance matrix of n samples $\mathbf{x}_{r,1}^i, \mathbf{x}_{r,2}^i, \ldots, \mathbf{x}_{r,n}^i$, r $(1 \leqslant r \leqslant R)$ is the number of training classes, i is the sequence number and d_r is the class label of n samples $\mathbf{x}_{r,1}^i, \mathbf{x}_{r,2}^i, \ldots, \mathbf{x}_{r,n}^i$. Therefore, the whole training set in the covariance feature space will be $\{\mathbf{y}_{1,1}, \mathbf{y}_{1,2}, \ldots, \mathbf{y}_{1,\lfloor t_1/n \rfloor}, \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \ldots, \mathbf{y}_{2,\lfloor t_2/n \rfloor}, \ldots, \ldots, \mathbf{y}_{R,1}, \mathbf{y}_{R,2}, \ldots, \mathbf{y}_{R,\lfloor t_R/n \rfloor}\}$, where the covariance matrices in $\{\mathbf{y}_{1,1}, \mathbf{y}_{1,2}, \ldots, \mathbf{y}_{1,\lfloor t_1/n \rfloor}\}$ have the label d_1, the covariance matrices in $\{\mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \ldots, \mathbf{y}_{2,\lfloor t_2/n \rfloor}\}$ have the label d_2 and the covariance matrices in $\{\mathbf{y}_{r,1}, \mathbf{y}_{r,2}, \ldots, \mathbf{y}_{r,\lfloor t_r/n \rfloor}\}$ have the label d_r, $1 \leqslant r \leqslant R$ and $\lfloor t_1/n \rfloor + \lfloor t_2/n \rfloor + \cdots + \lfloor t_R/n \rfloor = T$. The aim of the classification is to compute a classifier, such as $f(d|\sigma_{f_1 f_1} \sigma_{f_1 f_2} \ldots \sigma_{f_{n-1} f_n} \sigma_{f_n f_n})$, that can correctly label as many samples as possible. When we present an unknown sample e.g. $\mathbf{y}_{T+1} = (\sigma_{f_1 f_1}^{T+1}, \sigma_{f_1 f_2}^{T+1}, \ldots, \sigma_{f_{n-1} f_n}^{T+1}, \sigma_{f_n f_n}^{T+1})'$ as the input, the output of pattern recognition system is a d_{T+1}, which represents the class that $\mathbf{y}_{T+1}$ belongs to, e.g., either one of already known classes provided in the training stage (e.g. normal class or known attacks) or an unknown attack. Fig. 2 demonstrates the samples and the covariance features used in the covariance feature space.

3.3. Training and testing data

The data set we use is KDDCUP 99 data set which can be found at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. It is constructed on the basis of the raw data of TCP dump from 1998 DARPA evaluations [29] for the purpose of network intrusion detector competition. The testing data in KDDCUP 99 data set are not labeled, which makes it difficult to evaluate the performance of different detection techniques. Hence, we use the training data set for both training and testing in our study. A covariance matrix mainly reflects the correlation information of a sequence of data which satisfies the DoS attacks initiation where tons of packets from one type of DoS attacks comes within a short time. Therefore, we select all of the DoS attack samples from the whole training set of KDDCUP 99 as the data used in our experiments. We use $\frac{3}{5}$ randomly selected data as the training set and use the rest $\frac{2}{5}$ data as the testing set. The training set and testing set are disjoint. Table 1 gives a description of the data used in our experiments. In order to investigate the effect of different sequence lengths on the classification performance, we also test different sequence lengths in the experiments. As mentioned in Section 3.1, the samples in the covariance feature space are covariance matrices, therefore, we also list different covariance matrix samples under different sequence lengths of $n = 10$ (in the column *Cov_Len1_10*), $n=50$ (in the column *Cov_Len2_50*) and $n=150$ (in the column *Cov_Len3_150*) in Table 1, respectively. The samples in the column *Cov_Len1_10* and *Cov_Len2_50* use the non-overlapped sequences while the samples in the column *Cov_Len1_150* use the overlapped sequences by sliding 50 samples once a time. Let us take the normal class as an example. The original sample number of the normal class is 972,780. The $\frac{3}{5}$ training set will contain a total of 583,668 original samples as shown in the column of *Original*. The number of corresponding covariance matrix samples will be 58,366 in the training set with sequence length $n = 10$ as described in the column of *Cov_Len2_10*. Similarly, the training set with sequence length $n=50$ will contain a total of 11,673 covariance matrices as described in the column of *Cov_Len3_50*. If we use sequence length 150 and slide 50 original samples once a time, we will obtain a total of 11,671 covariance matrices as described in the column of *Cov_Len3_150*. Totally, the KDDCUP 99 data set contains six different types of DoS attacks as listed in Table 1. The last DoS attack type "land" only has 21 samples. Since it is not always possible to formulate a classification model to learn the detector with insufficient training data [14], we neglect the "land" attack in the experiments. Similarly, we also neglect the "pod" attack in the detection when the sequence length is set to 150.

Table 1
Multiple DoS attack samples used in the experiments

Type	Total	Original		*Cov_Len1_10*		*Cov_len2_50*		*Cov_len3_150*	
		Train	Test	Train	Test	Train	Test	Train	Test
Normal	972,780	583,668	389,112	58,366	38,911	11,673	7782	11,671	7780
Neptune	1,072,017	643,211	428,806	64,321	42,880	12,864	8576	12,862	8574
Smurf	2,807,886	1,684,732	1,123,154	168,473	112,315	33,694	22,463	33,692	22,461
Back	2203	1322	881	132	88	26	17	24	15
Pod	264	159	105	15	10	3	2	Null	Null
Teardrop	979	588	391	58	39	11	7	9	5
Land	21	Null	Null	Null	Null	Null	Null	Null	Null

3.4. Features used

As described in Section 3.1, the features used in the detection are the covariances among p physical features. There are a total of 41 features provided in the KDDCUP data set. Among them, some features describe the audit information in host audit logs, which is mainly used in detecting host based intrusions such as U2R or U2L intrusions [14]. The others describe the network connection and traffic information which are called time based features [14]. Therefore, we employ a total of 9 time based features as the physical features to detect the DoS attacks in our experiments. These 9 time based traffic features measure the TCP connection statistics in 2 s, which can be statistically calculated on the monitored network traffic using a packet capturing program such as *TCPDUMP*. The features include *count*, *srv_count*, *serror_rate*, *srv_serror_rate*, *rerror_rate*, *srv_serror_rate*, *same_srv_rate*, *diff_srv_rate* and *srv_diff_host_rate*. A detailed description of the meaning of each feature can be found in Ref. [14] or the webpage of http://kdd.ics.uci.edu/databases/kddcup99/task.html. Consequently, the dimension of the covariance feature space in our experiments will be $(9^*(9+1)/2) = 45$.

4. Detection approaches

In order to investigate the performance of different detectors in detecting different types of attacks in the covariance feature space, we employ two statistical pattern recognition approaches. One is the threshold based detection approach where the classification boundaries are determined by the threshold matrices, the other is the traditional decision tree approach. The reasons why we compare the performance of these two approaches are that: (i) both are typical supervised statistical pattern recognition approaches and (ii) the classification boundaries in both approaches have specific meanings. They can measure the covariance differences on each dimension of the covariance feature space by either a threshold or a rule.

In this section, we describe the basic idea of threshold based approach and the decision tree approach. We will compare and discuss the experimental results of these two techniques for intrusion detection in the covariance feature space in the next section.

4.1. Threshold based approach

Suppose that we have samples from R already known classes: $\omega_1, \omega_2, \ldots, \omega_R$. For each class ω_r $(1 \leqslant r \leqslant R)$, its training set consists of all the covariance matrices calculated on the sample sequences of equal, fixed length n. For instance, we obtain a total of l covariance matrices in the training set of class ω_r as $\{\mathbf{M}_r^1, \mathbf{M}_r^2, \ldots, \mathbf{M}_r^l\}$, where each covariance matrix $\mathbf{M}_r^i$ $(1 \leqslant i \leqslant l)$ describes the correlation among samples in the sequence T_i. In order to evaluate the differences between two covariance matrices, we define a dissimilarity function $Dist(\mathbf{A}, \mathbf{B}) = (d_{uv})_{p\times p}$ between two covariance matrices $\mathbf{A}$ and $\mathbf{B}$ as follows:

$$\forall a_{uv} \in \mathbf{A} \quad \forall b_{uv} \in \mathbf{B}, \quad d_{uv} = |a_{uv} - b_{uv}|. \tag{23}$$

The center of training covariance matrices in each class is utilized to construct the class profile. Given a new $\mathbf{M}^j$, the threshold based classifier will assign it the class label according to the following classification algorithm:

$$\begin{cases} \text{if } \exists r, \ni Dist(\mathbf{M}^j, \mathbf{E}(\omega_r)) \leqslant \boldsymbol{\delta}_r, & \text{then } \mathbf{M}^j \in \omega_r, \\ \text{else } \forall r, \ni Dist(\mathbf{M}^j, \mathbf{E}(\omega_r)) > \boldsymbol{\delta}_r, & \text{then } \mathbf{M}^j \in \text{unknown attack}, \end{cases} \tag{24}$$

where r is the training class label, $1 \leqslant r \leqslant R$, $\mathbf{E}(\omega_r)$ is the expectation of training class ω_r, $\boldsymbol{\delta}_r$ is the settled threshold matrix for class ω_r.

As we know, each element in a covariance matrix reflects the correlation between two features. The threshold matrix $\boldsymbol{\delta}_r$ will measure correlation differences among all the observed features between an observed covariance matrix and the profile of each class. If all difference matrices among the observed covariance matrix and already known class' profiles exceed the ranges within which the corresponding threshold matrices restricts, an unknown attack will be signaled.

The aim of the threshold based approach is to find a suitable threshold matrix for each training class where each element in the threshold matrix can provide a reasonable range to cover the variance of the covariance changes in the same class on the one hand and to keep the samples from other classes outside on the other hand. Therefore, a total of R threshold matrices $\boldsymbol{\delta}_r$ $1 \leqslant r \leqslant R$ are required to obtain for R already known classes $\omega_1, \omega_2, \ldots, \omega_R$ in the training stage. In order to settle a practical threshold $\boldsymbol{\delta}_r$ for each class ω_r $1 \leqslant r \leqslant R$, we employ the

Chebyshev's Inequality $P(|X - \mathbf{E}(X)| < \varepsilon) \geqslant 1 - \mathbf{D}(X)/\varepsilon^2$ as follows:

For each element $\sigma_{f_u^l f_v^l}$ in a training covariance matrix $\mathbf{M}^l$, we will obtain

$$\forall(u, v), \quad P(|\sigma_{f_u^l f_v^l} - \mathbf{E}(\sigma_{f_u^l f_v^l})| < \varepsilon) \geqslant 1 - \mathbf{D}(\sigma_{f_u^l f_v^l})/\varepsilon^2, \quad (25)$$

$$\mathbf{D}(\sigma_{f_u^l f_v^l}) = \frac{1}{s}\sum_{l=1}^{s}(\sigma_{f_u^l f_v^l} - \sigma_{f_u f_v})^2, \quad (26)$$

where $\sigma_{f_u^l f_v^l}$ is the covariance between feature f_u and f_v, $\mathbf{E}(\sigma_{f_u^l f_v^l})$ is the expected value of $\sigma_{f_u^l f_v^l}$ in class ω_r, s is the total number of sequences of length n in class ω_r.

Let $\varepsilon = 3\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ and $\varepsilon = 4\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ respectively, we will obtain

$$\forall(u, v), \quad P(|\sigma_{f_u^l f_v^l} - \sigma_{f_u f_v}| < 3\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}) \geqslant 1 - \tfrac{1}{9}, \quad (27)$$

$$\forall(u, v), \quad P(|\sigma_{f_u^l f_v^l} - \sigma_{f_u f_v}| < 4\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}) \geqslant 1 - \tfrac{1}{16}. \quad (28)$$

Eqs. (26)–(28) provide a solution to determine the value of each element in the threshold matrix $\boldsymbol{\delta}_r$, subject to the detection probability of each class. For example, if the requirement of the probability of correctly detecting ω_r is 88.89%, the lower bound of the threshold matrix should be set to $3\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ as indicated in Eq. (27). Similarly, if the requirement of the probability of correctly detecting ω_r is 93.75%, the lower bound of the threshold matrix should be set to $4\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ as indicated in Eq. (28).

4.2. Decision tree approach

The decision tree approach presented in this paper is mainly used for performance evaluation in the covariance feature space, hence we only briefly introduce its principle and theory in this section. The details of the decision tree theory can be found in Ref. [30].

Decision tree is a statistical classification approach which encodes a classifier in a form of a tree. The aim of the decision tree techniques is to find a tree that can correctly assign labels to the samples in the training set. The knowledge represented by the tree can be expressed into rules which make the decision tree easy to understand and presentable to non-specialists. A typical greedy algorithm to construct a decision tree is in a top-down recursive divide-and-conquer manner. At start, all the training samples are at the root and attributes (features) are categorical. Then the examples are partitioned recursively based on selected attributes. Test attributes are selected on the basis of a heuristic or statistical measure. Many different approaches can be used to construct a decision tree. In this paper, we utilize commercial software *See5* to construct a decision tree to detect intrusions in covariance feature space. The decision tree's optimizations, such as obtaining a small size of the tree and compressing classification rules, have already been provided by *See5*.

```
:...  σ_{f7 f8} ≤ -0.0071 :
:       :...σ_{f1 f7} > -5.054 : Normal Class (844)
:       :   σ_{f1 f7} ≤ -5.054 :
:       :   :...σ_{f3 f7} ≤ -0.060028 : Teardrop Attack (4)
:       :       σ_{f3 f7} > -0.060028 : Neptune Attack (7)
:...  σ_{f7 f8} > -0.0071 :
:       :...σ_{f2 f2} ≤ 100.99 : Neptune Attack (64177)
:           σ_{f2 f2} ≤ 100.99 :
:           :...  σ_{f4 f4} > 0.03223 : Neptune Attack (7)
:                 σ_{f4 f4} ≤ 0.03223 :
                        ... ... ...
```

Fig. 3. Examples of sub-tree in the covariance feature space.

```
For each covariance matrix sample M^j
    if   Dist(M^j, E(ω1)) ≤ δ1,    then M^j ∈ ω1
    elseif Dist(M^j, E(ω2)) ≤ δ2,  then M^j ∈ ω2
    elseif Dist(M^j, E(ω3)) ≤ δ3,  then M^j ∈ ω3
        :                :              :
        :                :              :
    elseif Dist(M^j, E(ωR)) ≤ δR,  then M^j ∈ ωR
    else   M^j ∈ Unknown Attack
end
```

Fig. 4. A rule-like implementation of the classification algorithm.

A decision sub-tree we obtained in the intrusion detection in the covariance feature space is exemplified in Fig. 3, where the sequence length is set to 10.

Since different sequence lengths correspond to different training sets, the constructed decision trees and the classification rules will be different with different sequence lengths. However all the constructed trees have the same forms as shown in Fig. 3. The number in the bracket is the number of samples which have been assigned the class label. For example, the second line in Fig. 3 shows that 844 covariance matrices have been assigned as the normal class under the rule of "if $\sigma_{f_7 f_8} \leqslant -0.0071$ and $\sigma_{f_1 f_7} > -5.054$, then Normal Class".

5. Result comparisons and discussions

This section describes the results obtained by applying the threshold based and decision tree approaches to detect the normal class and five different types of DoS attacks as described in the previous section, in order to evaluate the performance of intrusion detection in covariance feature space.

5.1. Threshold based approach results

In the experiments, we apply the following sequential rule-like technique to implement the classification algorithm as described in Eq. (24). Firstly, we order all the known classes in the training set based on the number of samples. For instance, we obtain R ordered classes as $\omega_1, \omega_2, \ldots, \omega_R$ such that class ω_1 contains the largest number of samples while ω_R contains the smallest number of samples. Then the classifier will determine the label of a test sample according to the procedure in Fig. 4.

Table 2
Detection results of threshold based approach with sequence length $n = 10$ and threshold based on $3\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ principle

	Normal	Neptune	Smurf	Back	Pod	Teardrop	Unknown attack
Normal	32,109	87	210	0	0	0	6505
Neptune	8	39,717	0	0	0	7	3148
Smurf	0	0	111,022	0	0	0	1293
Back	80	0	1	1	0	0	6
Pod	10	0	0	0	0	0	0
Teardrop	9	0	3	0	0	23	4

Table 3
Detection results of threshold based approach with sequence length $n = 50$ and threshold based on $3\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ principle

	Normal	Neptune	Smurf	Back	Pod	Teardrop	Unknown attack
Normal	6235	0	0	0	0	0	1547
Neptune	0	7611	0	0	0	0	965
Smurf	0	0	21,772	0	0	0	691
Back	14	0	0	1	0	0	2
Pod	2	0	0	0	0	0	0
Teardrop	0	0	0	0	0	5	2

Table 4
Detection results of threshold based approach with sequence length $n = 150$ and threshold based on $3\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ principle

	Normal	Neptune	Smurf	Back	Pod	Unknown attack
Normal	6135	0	0	0	0	1645
Neptune	0	7702	0	0	0	872
Smurf	0	0	21,773	0	0	688
Back	4	0	0	6	0	5
Teardrop	0	0	0	0	1	4

Tables 2–4 summarize the detection results of the threshold based detection approach, which employs $3\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ principle of Chebyshev's Inequality as described in Eq. (27) on different covariance matrix data sets of *Cov_Len2_10*, *Cov_Len2_50* and *Cov_Len3_150*, respectively.

In order to show different performance of the threshold based approach as an intrusion detector, we employ different performance indices such as detection rate, false positive rate, and false negative rate detector. We also employ the performance indices of classification precision rate and classification error rate to show the performance of the threshold based approach as a multiple classifier. The results are given in Table 5, where the column 3D means the threshold matrices are settled based on $3\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ principle and the column 4D means the threshold matrices are settled based on $4\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ principle.

The high detection rates and high classification precision rates in Table 5 show that the threshold based detection approach is effective in distinguishing multiple attacks in the covariance feature space. Table 5 also shows that classification precision rate increases with the increase of the threshold value. Since each element in the threshold matrix restricts the range of the covariance changes in the same class, these results satisfy the mechanism of the threshold based approach. We also find out that with the increase of the sequence length, the detection rate and classification precision rate also increase whereas the false positive rate, false negative rate and classification error rate decrease, which indicate that the more samples are used to calculate a covariance matrix, the more accurate the multiple classification is in the covariance feature space.

5.2. Decision tree results

Similar experiments have been made by employing the *See5* software on the same data sets as described in Table 1. The experimental results are presented in Tables 6–8 , respectively.

In order to reflect the performance differences of the decision tree approach with different sequence lengths, we compare the performance of the decision tree approach with different sequence lengths as described in Table 9.

Table 9 shows that the decision tree approach achieves very high detection rates and very low false positive and false negative rates in detecting multiple intrusions in covariance feature space. It is also shown that the detection rate and classification precision rate increase with the increase of the sequence length whereas the false positive rate, false

Table 5
Performance comparisons under different thresholds and sequence lengths

	Cov_Len1_10		*Cov_Len2_50*		*Cov_Len3_150*	
	3D (%)	4D (%)	3D (%)	4D (%)	3D (%)	4D (%)
Detection rate	98.91	97.88	99.95	99.94	99.99	99.95
Classification precision rate	94.91	95.29	94.19	96.87	91.71	96.60
False positive rate	17.27	11.05	19.88	9.23	21.14	10.33
False negative rate	1.09	2.12	0.05	0.06	0.01	0.05
Classification error rate	5.09	4.71	5.81	3.13	8.29	3.40

Table 6
Detection results of decision tree with sequence length $n = 10$

	Normal	Neptune	Smurf	Back	Pod	Teardrop
Normal	38,655	13	231	12	0	0
Neptune	20	42,860	0	0	0	0
Smurf	0	0	112,315	0	0	0
Back	15	0	1	72	0	0
Pod	8	0	0	0	2	0
Teardrop	0	0	6	0	0	33

Table 7
Detection results of decision tree with sequence length $n = 50$

	Normal	Neptune	Smurf	Back	Pod	Teardrop
Normal	7782	0	0	0	0	0
Neptune	0	8576	0	0	0	0
Smurf	0	0	22,463	0	0	0
Back	0	0	0	17	0	0
Pod	2	0	0	0	0	0
Teardrop	0	0	7	0	0	0

Table 8
Detection results of decision tree with sequence length $n = 150$

	Normal	Neptune	Smurf	Back	Pod
Normal	7780	0	0	0	0
Neptune	0	8574	0	0	0
Smurf	0	0	22,461	0	0
Back	0	0	0	15	0
Teardrop	0	0	5	0	0

Table 9
Performance comparisons under different sequence lengths

	Cov_Len1_10 (%)	*Cov_Len1_50* (%)	*Cov_Len1_150* (%)
Detection rate	99.97	99.97	99.98
Classification precision rate	99.84	99.98	99.99
False positive rate	0.66	0.00	0.00
False negative rate	0.03	0.01	0.00
Classification error rate	0.16	0.02	0.01

negative rate and classification error rate decrease with the increase of the sequence length. The above results also indicate that the more samples are used to calculate a covariance matrix, the more accurate the multiple classification is for the decision tree classifier in the covariance feature space.

Table 10
Performance comparisons of different detection approaches using the same sequence length

	Threshold (%)	Decision tree (%)
Detection rate	99.95	99.98
Classification precision rate	96.60	99.99
False positive rate	10.33	0.00
False negative rate	0.05	0.00
Classification error rate	3.40	0.01

Table 11
Detection results of threshold based approach in identifying unknown attack

New attack (number of samples)	Detection rate of detecting an unknown attack as an unknown attack	
	$3\sqrt{\mathbf{D}(\sigma_{f_2^l f_1^l})}$ principle (%)	$4\sqrt{\mathbf{D}(\sigma_{f_2^l f_1^l})}$ principle (%)
Neptune (8574)	100	100
Smurf (22,461)	100	100
Back (15)	73.33	6.67
Teardrop (5)	100	100

5.3. Result comparisons and discussions

We compare the experimental results of the threshold based approach and the decision tree approach using the same sequence length $n = 150$ in Table 10 , where the threshold matrices are settled according to $4\sqrt{\mathbf{D}(\sigma_{f_u^l f_v^l})}$ principle.

Table 10 shows that the experimental results of decision tree approach are either better than or comparable to that of the threshold based detection approach in detecting multiple intrusions in the covariance feature space, since both approaches achieve a high detection rate and a high classification precision rate. However, the decision tree approach achieves a lower false positive rate and a lower false negative rate. The reasons that the decision tree is better than the threshold based approach are due to: (i) the commercial software *See5* provides an optimal process in constructing the decision tree, whereas the threshold based approach only provides a heuristic threshold determination algorithm and (ii) the KDDCUP data set is constructed by means of data mining approaches [14], which is more suitable for a decision tree approach. As we know, many flaws exist in the KDD data set as discussed in Ref. [31], but to our best of knowledge, the KDDCUP 1999 contains many labeled attacks which can serve as a benchmark to evaluate different intrusion detection methodologies.

As an intrusion detector, the capability of identifying the unknown attacks is another important performance indicator. Although decision tree can achieve a high detection rate for the known classes, it cannot identify any unknown classes. In order to demonstrate the performance of the threshold based approach in identifying the unknown attacks, we simulate the unknown attacks through a leave-one-out approach. Let us take the Neptune attack as an example. In the leave-one-out detection experiments, we delete all the samples of Neptune attack in the training set but still keep the samples of Neptune attack in the testing set. That is, the $\frac{3}{5}$ samples which are originally used in the training set as described in Table 1 are deleted while the rest $\frac{2}{5}$ samples in the testing set are still kept. Therefore, Neptune attack will be served as an unknown attack in the testing stage because the detector will not get any information of Neptune attack in the training stage. Table 11 summarizes the detection results of threshold based approach in identifying unknown attacks where the sequence length is set to 150. The first column indicates the unknown class which is not included in the training set but appears in the testing set.

Table 11 shows that the threshold based approach achieves a very high detection rate in detecting the unknown attacks, especially in detecting the unknown attacks which contains large number of samples such as *Neptune* and *Smurf*.

In the experiments, we use two detection approaches and employ different sequence lengths to evaluate the detection performance in the covariance feature space. The experimental results show that the covariance can be a good feature to be used to classify multiple and various DoS attacks. In practice, how to determine a suitable sequence length will be a problem. In fact, the sequence length determines how many the monitored packets will be enough to embody the covariance characteristics of the network traffic. It can be determined through training. In details, we can extract the data with different sequence lengths from the collected normal traffic trace and compare the average covariance matrix difference between the sequential data and the whole data set. Among all sequential data with different lengths, we can determine such a sequence length where the average difference between the sequential data and the whole trace levels off relatively within a small range. Since the sequential data with such a length can embody the statistical covariance characteristics of whole trace with a little gap, it can be settled as a practical sequence length.

6. Conclusions

This paper directly utilizes the covariance matrices of sample sequences to detect multiple and various attacks. With respect to the behavior of typical DoS network intrusions, it firstly analyzes the difficulties that traditional mean based detection approaches fail to improve the effectiveness and explains that a better detection performance can be achieved by utilizing the statistical information contained in groups of network samples. By constructing a covariance feature space, a detection approach can thus utilize the correlation differences of sequential samples to identify multiple network attacks. Two typical statistical detection approaches are evaluated to detect multiple attacks in the covariance feature space. One is the proposed threshold based detection approach where the classification boundaries are determined by the corresponding threshold matrices while the other is the traditional decision tree approach. A public data set which contains all different types of DoS attacks is used for the evaluations. The experimental results show that both approaches are effective in distinguishing multiple known attacks in the covariance feature space. Compared with the decision tree, the threshold based approach offers an advantage of identifying the unknown attacks with a high detection rate. The high performance of both approaches in the covariance feature space verifies that different network intrusions have different correlation statistics which can be directly utilized in the covariance feature space to distinguish multiple and various network intrusions effectively.

It is also pointed out that the covariance based detection will succeed in distinguishing multiple classes with near or equal means while any traditional mean based classification approach will fail. In summary, despite some open problems such as what the favorite feature set is and how to determine a suitable sequence length and so on, utilizing the covariance information directly in the detection will improve the detection effectiveness. It will be worth a wider study to help the understanding of the characteristics of different network intrusions and to be applied into other applications in pattern recognition.

Acknowledgment

This research work is supported by the Hong Kong RGC Project Research Grant B-Q571.

References

[1] J. Kay, Low volume viruses: new tools for criminals, Network Secur. 6 (2005) 16–18.
[2] D.E. Denning, An intrusion-detection model, IEEE Trans. Software Eng. 13 (2) (1987) 222–232.
[3] J. Ryan, M.J. Lin, R. Miikkulainen, Intrusion detection with neural networks, Advances in Neural Information Processing, MIT Press, Cambridge, MA, 1998.
[4] S. Forrest, S.A. Hofmeyr, A. Somayaji, T.A. Longstaff, A sense of self for unix processes, in: IEEE Symposium on Security and Privacy, 1996, pp. 120–128.
[5] W.S. Sarle, Neural networks and statistical models, in: Proceedings of 19th Annual SAS User Group Internet Conference, April 1994.
[6] S. Mukkamala, A.H. Sung, Feature ranking and selection for intrusion detection using support vector machines, Presentations in Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, June 2002.
[7] M. Fugate, J.R. Gattiker, Computer intrusion detection with classification and anomaly detection using SVMs, Int. J. Pattern Recognition Artif. Intell. 17 (3) (2003) 441–458.
[8] T. Lane, C.E. Brodley, Temporal sequence learning and data reduction for anomaly detection, ACM Trans. Inform. Syst. Secur. 2 (3) (1999).
[9] M.O. Depren, M. Topallar, E. Anarim, K. Ciliz, Network-based anomaly intrusion detection system using SOMs, in: IEEE 12th Signal Processing and Communications Applications Conference, April 2004, pp. 76–79.
[10] E. Leon, O. Nasraoui, J. Gomez, Anomaly detection based on unsupervised niche clustering with application to network intrusion detection, Evol. Comput. 2004(CEC2004) 1 (2004) 502–508.
[11] J. Jung, B. Krishnamurthy, M. Rabinovich, Flash crowds and denial of service attacks: characterization and implications for CDNs and web sites, in: The Eleventh International World Wide Web Conference, May 2002.
[12] T. Abbes, A. Bouhoula, M. Rusinowitch, Protocol analysis in intrusion detection using decision tree, Inform. Technol. Coding Comput. 2004(ITCC 2004) 1 (2004) 404–408.
[13] L. Peck, G. Trachier, Security technology decision tree tool, in: The 38th Annual 2004 International Carnahan Conference on Security Technology, October 1999, pp. 91–98.
[14] W. Lee, A data mining framework for constructing features and models for intrusion detection systems, Ph.D. Dissertation, Columbia University, 1999.
[15] H. Debar, M. Dacier, A. Wespi, Towards a taxonomy of intrusion-detection systems, Comput. Networks 31 (1999) 805–822.
[16] A.K. Jones, R.S. Sielken, Computer system intrusion detection: a survey, Technical Report, Samuel Mc Innes Bechard University of Virginia Computer Science Department, September 2000.
[17] A. Chakrabarti, G. Manimaran, Internet infrastructure security: a taxonomy, IEEE Network (11–12) (2002).
[18] Z. Zhang, J. Li, C.N. Manikopoulos, J. Jorgenson, J. Ucles, HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification, Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, June 2001.
[19] S.A. Hofmeyr, The implications of immunology for secure systems design, Computers Secur. 23 (6) (2004) 453–455.
[20] N. Ye, S.M. Emran, Q. Chen, S. Vilbert, Multivariate statistical analysis of audit trails for host-based intrusion detection, IEEE Trans. Comput. 51 (7) (2002) 810–820.
[21] D.Y. Yeung, Y. Ding, Host-based intrusion detection using dynamic and static behavioral models, Pattern Recognition 36 (2003) 229–243.
[22] S.T. Sarasa, Q.A. Zhu, J. Huff, Hierarchical kohonenen net for anomaly detection in network security, IEEE Trans. Syst. Man, Cybern.—part B: Cybernetics 35 (2) (2005) 302–312.
[23] M. Thottan, C. Ji, Anomaly detection in IP networks, IEEE Trans. Signal Processing 51 (8) (2003) 2191–2204.
[24] S. Jin, D. Yeung, A covariance analysis model for DDoS attack detection, IEEE International Communication Conference (ICC04), vol. 4, June 2004, pp. 20–24.
[25] S. Axelsson, The base-rate fallacy and the difficulty of intrusion detection, ACM Trans. Inform. Syst. Security 3 (3) (2000) 186–205.
[26] L. Feinstein, D. Schnackenberg, Statistical approaches to DDoS attack detection and response, in: Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03), April 2003.
[27] C. Manikopoulos, S. Papavassiliou, Network intrusion and fault detection: a statistical anomaly approach, IEEE Commun. Mag. (2002).
[28] R.B. Blazek, H. Kim, B. Rozovskii, A. Tartakovsky, A novel approach to detection of Denial-of-Service attacks via adaptive sequential and batch-sequential change-point detection methods, Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, June 2002.

[29] Lincoln Laboratories, 1999 DARPA Intrusion Detection Evaluation, ⟨http://www.ll.mit.edu/⟩ IST/ideval/index.html.
[30] J.R. Quinlan, Induction of decision trees, Mach. Learn. (1986) 81–106.
[31] M.V. Mahoney, P.K. Chan, An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection, in: Proceedings of Recent Advances in Intrusion Detection (RAID03), 2003, pp. 220–237.

About the Author—SHUYUAN JIN received her B.Sc. and M.Sc. degrees in Computer Science from Harbin Institute of Technology, China, in 1996 and 1998, respectively. She obtained her Ph.D. degree from the Hong Kong Polytechnic University in 2006. Her main research interests include network security, especially intrusion detection and responses, Internet technologies, machine learning, pattern recognition techniques and applications.

About the Author—DANIEL S. YEUNG received the Ph.D. degree in applied mathematics from Case Western Reserve University in 1974. He is a Chair Professor in the department of Computing, Hong Kong Polytechnic University, Hong Kong. His current research interests include neural-network sensitivity analysis, data mining, Chinese computing, and fuzzy systems. He was the President of IEEE Hong Kong Computer Chapter, an associate editor for both IEEE Transactions on Neural Networks and IEEE Transactions on SMC (Part B). He is the Vice President for Technical Activities for the IEEE SMC Society. He leads a group of researchers in Hong Kong and China who are actively engaging in research works on computational intelligence and data mining.

About the Author—XIZHAO WANG received the B.Sc. and M.Sc. degrees in mathematics from Heibei University, Baoding, China, in 1983 and 1992, respectively, and the Ph.D. degree in computer science from Harbin Institute of Technology, China, in 1998. He is a Full Professor in the Department of Mathematics, Hebei University. His main research interests include inductive learning and fuzzy representation, fuzzy measures and integrals, neuro-fuzzy systems and genetic algorithms, and feature extraction.