

Multivariate Statistical Analysis of Audit Trails for Host-Based Intrusion Detection

Nong Ye, *Senior Member, IEEE*, Syed Masum Emran, Qiang Chen, and Sean Vilbert

Abstract—Intrusion detection complements prevention mechanisms, such as firewalls, cryptography, and authentication, to capture intrusions into an information system while they are acting on the information system. Our study investigates a multivariate quality control technique to detect intrusions by building a long-term profile of normal activities in information systems (norm profile) and using the norm profile to detect anomalies. The multivariate quality control technique is based on Hotelling's T^2 test that detects both counterrelationship anomalies and mean-shift anomalies. The performance of the Hotelling's T^2 test is examined on two sets of computer audit data: a small data set and a large multiday data set. Both data sets contain sessions of normal and intrusive activities. For the small data set, the Hotelling's T^2 test signals all the intrusion sessions and produces no false alarms for the normal sessions. For the large data set, the Hotelling's T^2 test signals 92 percent of the intrusion sessions while producing no false alarms for the normal sessions. The performance of the Hotelling's T^2 test is also compared with the performance of a more scalable multivariate technique—a chi-squared distance test.

Index Terms—Computer security, intrusion detection, multivariate statistical analysis, chi-square test, and Hotelling's T^2 test.

1 INTRODUCTION

As we increasingly rely on information infrastructures to support critical operations in defense, banking, telecommunication, transportation, electric power, and many other systems, intrusions into information systems have become a significant threat to our society with potentially severe consequences [1], [2]. An intrusion compromises the security (e.g., availability, integrity, and confidentiality) of an information system through various means, including denial-of-service, remote-to-local, user-to-root, information probing, and so on [3]. Denial-of-service intrusions make a host or network service unavailable by overloading or disrupting the service. Remote-to-local intrusions gain unauthorized access to a host machine without a legitimate user account on the host machine. User-to-root intrusions happen when a regular user on a host machine obtains privileges normally reserved for a root or super user. Information probing intrusions use programs to scan a network of computers for gathering information or finding known vulnerabilities.

Layers of defense can be set up against intrusions through prevention, detection, etc. Firewalls, authentication, and cryptography are some examples of the online intrusion prevention mechanisms used to protect information systems from external intrusions [4]. Offline intrusion prevention efforts focus on methodologies of secure software design and engineering. The online prevention mechanisms form a fence around information systems to

raise the difficulty level of breaking into information systems. However, the fence can only be raised to a level that does not degrade services from information systems. Although secure software methodologies will continue to improve, bugs and vulnerabilities in information systems are inevitable due to difficulty in managing the complexity of large-scale information systems during their specification, design, implementation, and installation. Intruders explore the bugs and vulnerabilities in information systems to attack information systems. Hence, we expect that some intrusions will be leaked through the fence of prevention and act on information systems.

Intrusion detection techniques capture intrusions while they are acting on an information system. Existing intrusion detection techniques fall into two major categories: signature recognition and anomaly detection [5], [6]. Signature recognition techniques, also referred to as misuse detection in some literature, match activities in an information system with signatures of known intrusions and signal intrusions when there is a match. For a subject (user, file, privileged program, host, network, etc.) of interest, anomaly detection techniques establish a profile of the subject's normal behavior (norm profile), compare the observed behavior of the subject with its norm profile, and signal intrusions when the subject's observed behavior deviates significantly from its norm profile. Therefore, anomaly detection techniques rely on a norm profile and consider a deviation of the subject's behavior from its norm profile as a symptom of an intrusion. A justification of using anomaly detection for intrusion detection is provided in [7].

Signature recognition techniques utilize intrusion signatures—profiles of intrusion characteristics—and consider the presence of an intrusion signature as evidence of an intrusion. Anomaly detection techniques use only data of normal activities in information systems for training and building a norm profile. Signature recognition techniques

• N. Ye, Q. Chen, and S. Vilbert are with Arizona State University, Box 875906, Tempe, AZ 85287-5906.

E-mail: {nongye, dannyqchen, sean.vilbert}@asu.edu.

• S.M. Emran is with Motorola, 1304 E. Algonquin Road, Schaumburg, IL 60173.

Manuscript received 12 Apr. 2000; revised 13 Nov. 2000; accepted 3 Oct. 2001.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 111908.

rely on data of both normal and intrusive activities for learning intrusion signatures either manually or automatically through data mining.

Signature recognition techniques have been used in most existing intrusion detection systems, including NSM/ASIM, NetRadar, IDES/NIDES, EMERALD, NetRanger, Stalker, CMDS, NetStalker, TCP Warpper, Tripwire, SATAN, and STAT [5], [6], [8], [9], [10], [11], [12]. Intrusion signatures have been characterized as strings, event sequences, activity graphs, and intrusion scenarios (consisting of event sequences, their preconditions, and target compromised states). Finite state machines [8], colored Petri Nets [9], associate rules [10] and production rules of expert systems [11], [12] have been used to represent and recognize intrusion signatures. Intrusion signatures are either manually encoded or automatically learned through data mining. However, signature recognition techniques have a limitation in that they cannot detect novel intrusions whose signatures are unknown.

Anomaly detection techniques capture both known intrusions and unknown intrusions if intrusions demonstrate a significant deviation from a norm profile. Several types of anomaly detection techniques exist: string-based, specification-based, and statistical-based [11], [12], [13], [14], [15], [16], [17], [18]. String-based anomaly detection techniques [13], [17] collect sequences of system calls or audit events that appear in normal activities from historic data, represent those sequences as strings, store those strings as a norm profile, and employ either negative selection [17] or positive selection [13] to determine whether an observed string deviates from the string-based norm profile. Specification-based anomaly detection techniques [18] use predicates in formal logic to specify normal activities in a norm profile, and employ logical reasoning to infer the consistency of observed activities with the norm profile. Statistical-based anomaly detection techniques use statistical properties (e.g., mean and variance) of normal activities to build a statistical-based norm profile, and employ statistical tests to determine whether observed activities deviate significantly from the norm profile.

An advantage of statistical-based anomaly detection techniques is their capability of explicitly representing and handling variations and noises involved in activities of information system, whereas string-based anomaly detection techniques and specification-based anomaly detection techniques lack such a capability of noise handling and variance representation. A norm profile must consider and represent variations of normal activities for distinguishing truly anomalous activities from expected variations of normal activities.

Most studies on statistical-based anomaly detection techniques [11], [12], [13], [14], [15], [16] are based on a statistical technique developed for IDES/NIDES. This technique computes test statistics of a normal distribution (called Q statistic and S statistic) using data on a single measure. This technique has several drawbacks. First of all, the technique is sensitive to the normality assumption. If data on a measure are not normally distributed, the

technique yields a high false alarm rate, especially when departures from normality are due to kurtosis (flatness). Second, the technique is univariate in that a statistical norm profile is built for only one measure of activities in information systems. However, intrusions often affect multiple measures of activities collectively. Anomalies resulting from intrusions may cause deviations on multiple measures in a collective manner rather than through separate manifestations on individual measures.

This paper presents our work on multivariate statistical analysis of audit trails for host-based intrusion detection. Specifically, Hotelling's T^2 test—a multivariate statistical process control (SPC) technique—is used to analyze audit trails of activities in an information system and detect host-based intrusions into the information system that leave trails in the audit data. Hotelling's T^2 test is also compared with a more scalable multivariate statistical analysis technique—the chi-squared test.

The rest of the paper is organized as follows: Section 2 describes the two multivariate statistical analysis techniques based on Hotelling's T^2 (T^2 test) and the chi-squared distance test (X^2 test) respectively. Section 3 defines the problem of intrusion detection. Section 4 presents and discusses the performance of the T^2 test in comparison with the performance of the X^2 test.

2 MULTIVARIATE STATISTICAL ANALYSIS

In this section, we first describe Hotelling's T^2 test and the chi-squared distance test. Then, we compare these two techniques, and discuss their differences from the statistical test in IDES/NIDES.

2.1 Hotelling's T^2 Test

Let $X_i = (X_{i1}, X_{i2}, \dots, X_{ip})'$ denote an observation of p measures on a process or system at time i . Assume that when the process is operating normally (in control), the population of X follows a multivariate normal distribution with the mean vector μ and the covariance matrix Σ . Using a data sample of size n , the sample mean vector \bar{X} and the sample covariance matrix S are usually used to estimate μ and Σ [19], [20], [21], [22], [23], [24], [25], [26], where

$$\bar{X} = (\bar{X}_1, \bar{X}_2, \dots, \bar{X}_p) \quad (1)$$

$$S = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})'. \quad (2)$$

Hotelling's T^2 statistic for an observation X_i is determined by the following [19], [20], [21], [22], [23], [24], [25], [26]:

$$T^2 = (X_i - \bar{X})' S^{-1} (X_i - \bar{X}). \quad (3)$$

A large value of T^2 indicates a large deviation of the observation X_i from the in-control population.

We can obtain a transformed value of the T^2 statistic,

$$\frac{n(n-p)}{p(n+1)(n-1)} T^2$$

which follows an F distribution with p and $n - p$ degrees of freedom, by multiplying T^2 by the constant

$$\frac{n(n-p)}{p(n+1)(n-1)}.$$

If the transformed value of the T^2 statistic is greater than the tabulated F value for a given level of significance, α , then we reject the null hypothesis that the process is in control (normal) and thus signal that the process is out of control (anomalous).

If X_i does not follow a multivariate normal distribution, the transformed value of the T^2 statistic may not follow an F distribution. As a result, we cannot use the tabulated F value as a signal threshold to determine whether a transformed value of the T^2 statistic is large enough for an out-of-control signal. For intrusion detection, multiple measures of activities in an information system are represented by multiple random variables. Typically, we do not know a priori what distribution each random variable follows, and thereby cannot assume that the variable follows a normal distribution.

However, if the p variables are independent and p is large (usually greater than 30), T^2 follows approximately a normal distribution according to the central limit theorem [20], regardless of what distribution each of the p variables follows. Using a sample of T^2 values, the mean and standard deviation of the T^2 population can be estimated from the sample mean $\overline{T^2}$ and the sample standard deviation S_{T^2} . The in-control limits to detect out-of-control anomalies are usually set to 3-sigma control limits [19], [20], [21], [22], [23], [24], [25], [26] as determined by $[\overline{T^2} - 3S_{T^2}, \overline{T^2} + 3S_{T^2}]$. Since we are interested in detecting significantly large T^2 , we need to set only the upper control limit to $\overline{T^2} + 3S_{T^2}$ as a signal threshold. That is, if the T^2 for an observation is greater than $\overline{T^2} + 3S_{T^2}$, we signal an anomaly.

An out-of-control signal from the T^2 test can be caused by a shift from the in-control mean vector (mean shift), a departure from the in-control covariance structure or variable relationships (counterrelationship), or combinations of the two situations. In a mean shift situation, one or more of the p variables is out of control. In a counterrelationship situation, a relationship between two or more of the p variables changes from the variable relationship established in the covariance matrix.

Although the T^2 test detects both mean shifts and counterrelationships, the T^2 test is more sensitive to counterrelationships than mean shifts because the T^2 test relies largely on the correlated structure of variables (covariance matrix) for signal detection. An example is illustrated in [21] with two variables and a high positive correlation between the two variables while they are in control. For this example, the T^2 test signals an observation with a counterrelationship, but does not signal an observation with an out-of-control mean shift on one variable because both variables shift in the same direction and thus still maintain their relationship of a positive correlation.

2.2 X^2 Test

We develop another multivariate statistical analysis technique based on the chi-squared distance test. In this paper, the X^2 test and its performance are mainly used for evaluating the performance of the T^2 test. Hence, the X^2 test is only briefly described in this paper. Details of the X^2 test can be found in [27].

If we have p variables to measure and X_j denotes the observation of the j th ($1 \leq j \leq p$) variable at a particular time, the X^2 test statistic is given by the equation:

$$X^2 = \sum_{j=1}^p \frac{(X_j - \overline{X_j})^2}{\overline{X_j}}. \quad (4)$$

This test statistic measures the distance of a data point from the center of a data population. Hence, we call this test the chi-square distance test. When the p variables are independent and p is large (usually greater than 30), the X^2 statistic follows approximately a normal distribution according to the central limit theorem [20], regardless of what distribution each of the p variables follows. Using a sample of X^2 values, the mean and standard deviation of the X^2 population can be estimated from the sample mean $\overline{X^2}$ and the sample standard deviation S_{X^2} . The in-control limits to detect out-of-control anomalies are usually set to 3-sigma control limits [20], [21] as determined by $[\overline{X^2} - 3S_{X^2}, \overline{X^2} + 3S_{X^2}]$. Since we are interested in detecting significantly large X^2 values, we need to set only the upper control limit to $\overline{X^2} + 3S_{X^2}$ as a signal threshold. That is, if the X^2 for an observation is greater than $\overline{X^2} + 3S_{X^2}$, we signal an anomaly.

2.3 Comparison of the T^2 Test and the X^2 Test

Both the T^2 test statistic and the X^2 test statistic measure the distance of an observation from the multivariate mean vector of a population. The T^2 test statistic uses the statistical distance that incorporates the multivariate variance-covariance matrix, whereas the X^2 test statistic uses the chi-squared distance. The chi-squared distance is similar to a Euclidean distance but using the average value on each variable to scale the Euclidean distance on that variable or dimension.

In general, anomalies involving multiple variables can be caused by shifts from the means of these variables (mean shifts), departures from variable relationships (counterrelationships), or combinations of mean shifts and counterrelationships. In contrast to the T^2 statistic, the X^2 statistic does not account for the correlated structure of the p variables. Only \overline{X} is estimated to establish the norm profile according to formula (1). Hence, the T^2 test detects both mean shifts and counterrelationships, whereas the X^2 test detects only the mean shift on one or more of the p variables.

The X^2 test performs well in intrusion detection [27]. When tested on a small set of computer audit data containing sessions of both normal and intrusive activities, the X^2 test signals all the intrusion sessions and produces no false alarms on the normal sessions. A session consists of many events. For the small data set, the X^2 test detects 75 percent of the intrusive events, and produces no false alarms on the normal events. For a large multiday data set, the X^2 test signals 60 percent of

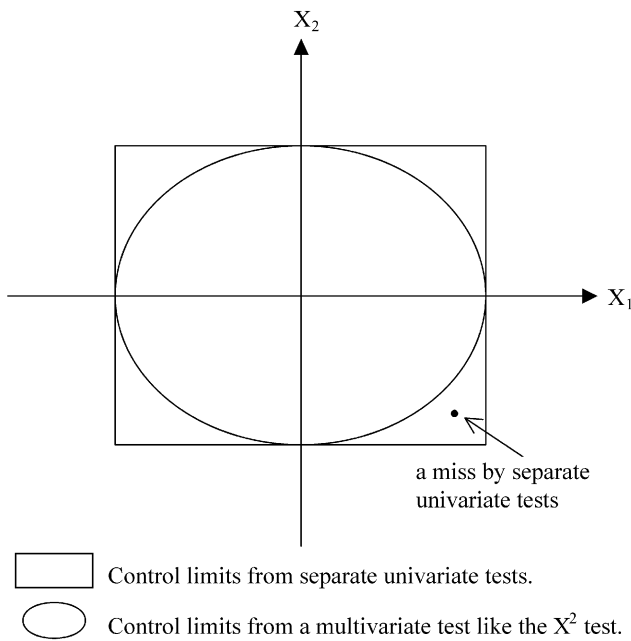


Fig. 1. Differences in control limits of separate univariate tests and the multivariate X^2 test.

intrusion sessions when the signal threshold is set so that no false alarms are produced on normal sessions.

To further investigate whether the additional capability of detecting counterrelationships in the T^2 test can produce a better performance in intrusion detection, we conduct the T^2 test on the same small data set and the same large data set in this study.

2.4 Differences from the Statistical Test in IDES/NIDES

The T^2 test and the X^2 test are multivariate statistical techniques coping with multiple measures of activities in information systems, whereas the statistical technique in IDES/NIDES is univariate for only one measure of activities. Hence, separate statistical tests must be performed for multiple measures of activities. Even without the consideration of the variance-covariance matrix, the X^2 test is still not equivalent to separate tests on individual variables measuring different characteristics of the same process such as univariate tests in IDES/NIDES. It is well understood [21], [26] that separate univariate tests on individual variables can lead to misses caused by incorrect control limits due to accumulated effects of significance probabilities. Fig. 1 shows differences between the control limits of separate univariate tests and the control limit of a multivariate test without a covariance structure such as the X^2 test.

IDES/NIDES [14], [15] proposes a multivariate test using the squared sum of the separate test statistics from multiple measures. However, this multivariate test is not robust because the individual univariate tests cannot robustly measure the distance of an observation from the mean on individual measures due to the sensitivity of the individual univariate tests to the normality assumption. In the T^2 test or the X^2 test, the distance of an observation from the mean on individual variables is measured using the statistical

distance or the chi-squared distance. The sum of the distance over multiple variables in the T^2 test or the X^2 test follows a normal distribution when multiple variables are independent and the number of multiple variable is large (e.g., greater than 30). Hence, the T^2 test and the X^2 test are robust to the normality assumption for individual variables. We illustrate later in this paper that there are more than 30 variables in an observation in our study, and our multiple variables are likely independent of each other.

3 PROBLEM DEFINITION

This section describes the intrusion detection problem, including the data source, training data, testing data, and problem representation.

3.1 Data Source

A computer and network system within an organization typically includes a number of host machines (e.g., machines running a UNIX operation system and machines running the Microsoft Windows operating system) and communication links connecting those host machines. Currently two sources of data have been widely used to capture activities in a computer and network system for intrusion detection: network traffic data and host audit trail data (audit data). Network traffic data contain data packets traveling over communication links between host machines to capture activities over communication networks. Audit data capture activities occurring on a host machine.

In this study, we use audit data from a UNIX-based host machine (specifically a Sun SPARC 10 workstation with the Solaris operating system), and focus on attacks on a host machine that leave trails in audit data. The Solaris operating system from Sun Microsystems Inc. has an auditing facility, called the Basic Security Module (BSM). BSM monitors the events related to the security of a system and records the crossing of instructions executed by the processor in the user space and instructions executed in the kernel. This is based on the assumption that the actions in the user space cannot harm the security of the system and the security-related actions that can impact the system only take place when users request services from the kernel. BSM records the execution of system calls by all processes launched by the users. A full system call trace gives us overwhelming information, whereas the audit trail provides a limited abstraction of the same information in which the context switches, memory allocation, internal semaphores and consecutive file reads do not appear. And there is always a straightforward mapping of audit events to system calls.

The BSM audit records contain detailed information about the events in the system. It includes detailed user and group identification—from the login identity to the one under which the system call is executed, the parameters of the system call execution—file names including full path, command line arguments etc., the return code from the execution and the error code. We, however, use only the event type information.

There are 284 different types of events in BSM audit data. In UNIX, there are thousands of commands available. Since the audit events are closer to the core of the operating system, the event type is more representative than the

actual command sequences used. For example, we can use any text editor, such as *vi* and *ed* to edit a file, but most of the time the audit event stream will contain the following event types: AUE_EXECVE, AUE_OPEN_R, AUE_ACCESS, AUE_STAT. In an intrusion session an intruder tries to hide the not-so-frequently-used commands essential for intrusion by adding large amounts of frequently used commands. This causes intrusion detection techniques to recognize these as noise and fail to detect the intruder's attempts. By using the event type information, we are able to extract out the redundant information about which particular commands are used, which particular files are accessed etc., and can focus on which particular kernel level event types the intruder uses. We also eliminate the problem of having different scenarios when we combine normal and intrusion sessions from different sources such as different host machines. As long as they belong to the same Solaris BSM, we get consistent event type information. Moreover, we need not look at the nature of these intrusions as we are interested only in detecting them regardless of how sophisticated the intrusions are.

Several studies [13], [17] also show that events types in information systems can be used to effectively detect intrusions for a specific application, a network service [17] or a host machine in general [13].

3.2 Training and Testing Data

In this study, we use two sets of data. A small set of audit data is sampled by recording both normal activities and intrusive activities on host machines with Solaris 2.5. Normal activities and intrusive activities are simulated to produce these audit data. Normal activities are simulated by the MIT Lincoln Laboratory according to normal activities observed in a real-world computer and network system [3]. A number of intrusions are also simulated in our laboratory on a host machine with Solaris 2.5, including password guessing, use of symbolic links to gain the root privilege, attempts to gain an unauthorized remote access, etc. The small data set is used in the early stage of our study when we do not have access to the complete data set from the MIT Lincoln Laboratory. There is only a small data set that is accessible to the general public. Since there are not many intrusions in such a small data set, we run the simulation of intrusion scenarios that we have collected over the years in our laboratory to create the audit data of intrusive activities. As discussed in the previous section, the use of the event type information minimizes the problem with two machine sources of audit data in the small data set.

In the small data set, the audit data of normal activities consist of 3,019 audit events, and the audit data of intrusive activities consist of 1,751 audit events. We use the first part of the audit data for normal activities as our training data set, and use the remaining audit data for normal activities and attack activities as our testing data set. The training data set consists of 1,613 audit events for normal activities. The testing data set consists of 1,406 audit events for normal activities and 1,225 audit events for intrusive activities.

For the large set of audit data, we obtain nine weeks of the 1998 audit data set from the MIT Lincoln Laboratory in a later stage of our study. The nine weeks of the 1998 audit

data set is divided into seven weeks of labeled training data and two weeks of unlabeled testing data. Since the testing data are not labeled, this makes it difficult to evaluate the performance of our techniques. Hence, we use the training data set for both training and testing in our study. During training, we use audit data of activities with the normal label to build the norm profile. During testing, we remove the label of audit data and use our techniques to generate a label. The labels of activities from our testing are then compared to the given labels for evaluating the performance of our techniques.

There are in total 35 days of data in the seven weeks of training data. We pick 4 days of data as a representative of the entire training data set. We pick two days with relatively few intrusions and two days with comparatively more intrusions, varied in length. We have chosen week-1, Monday data as day-1 data, week-4, Tuesday data as day-2 data, week-4, Friday data as day-3 data and week-6, Thursday data as day-4 data. Table 1 summarizes the statistics about these 4 days of data.

As we can see from Table 1, the average session length is comparatively smaller in day-2 and day-3 than that in day-1 and day-4. Around 3 percent of audit events on day-2 and day-4 are due to intrusive activities whereas less than 0.80 percent of audit events in day-1 and day-3 data are from normal activities. In terms of sessions, almost one-fourth of the sessions in day-2 and one-twelfth of the sessions in day-3 are intrusion sessions. Day-1 contains mostly normal sessions, day-4 also does not have too many intrusion sessions. A total of 176 instances of nine types of intrusions are present in these four days of data. Our objective is to detect any ongoing intrusions rather than any particular type of intrusion, so our concern is about how many of these intrusion sessions we can detect.

We use only the normal events of the first two days of data for training. Day-1 and day-2 contain 740,995 and 1,283,903 audit events arising from 296 and 372 normal sessions, respectively. Day-3 and day-4 are used for testing which contain 2,232,981 and 893,603 audit events for normal activities respectively. Numbers of audit events for intrusive activities in these two days are 16,524 and 31,476 arising from 29 and 14 intrusion sessions respectively. Day-3 contains 339 sessions and day-4 contains 447 sessions in total comprising both normal and intrusive sessions. In the large data set, an intrusion occurs in one session only.

3.3 Problem Representation

A BSM audit record for each event contains a variety of information including the event type, user ID, group ID, process ID, session ID, the system object accessed, and so on. As discussed in Section 2.1, in this study we extract and use the event type from the record of each audit event. There are 284 different types of BSM audit events from Solaris 2.5. All the 284 types of audit events are considered in this study.

Activities on a host machine are captured through a continuous stream of audit events, each of which is characterized by the event type. For intrusion detection, we want to build a long-term profile of normal activities, and to compare the activities in the recent past to the long-

TABLE 1
Statistics about the Large Data Set

| | Day-1 | Day-2 | Day-3 | Day-4 |
|---|--------|---------|---------|--------|
| <i>Event Information</i> | | | | |
| Number of Events | 744085 | 1320478 | 2249505 | 925079 |
| Number of Intrusive Events | 3090 | 36575 | 16524 | 31476 |
| Percentage of Intrusive Events | 0.42% | 2.77% | 0.73% | 3.40% |
| <i>Session Information</i> | | | | |
| Number of Sessions | 298 | 503 | 339 | 447 |
| Number of Intrusive Sessions | 2 | 131 | 29 | 14 |
| Percentage of Intrusive Sessions | 0.67% | 26.04% | 8.55% | 3.13% |
| <i>Number of Events per Normal Session</i> | | | | |
| Average | 2503 | 3451 | 7203 | 2064 |
| Minimum | 1 | 69 | 74 | 96 |
| Maximum | 253827 | 462287 | 1019443 | 214705 |
| <i>Number of Events per Intrusive Session</i> | | | | |
| Average | 1545 | 279 | 570 | 2248 |
| Minimum | 1101 | 142 | 166 | 1107 |
| Maximum | 1989 | 1737 | 4986 | 2841 |

term norm profile for detecting a significant deviation. We define activities in the recent past from time $t - k$ to time t by a vector of $(X_1, X_2, \dots, X_{284})$ for the 284 event types respectively, based on the exponentially weighted moving average technique [20]. At time t , the audit events in the recent past from time $t - k$ to time t are summarized as follows:

$$X_i(t) = \lambda * 1 + (1 - \lambda) * X_i(t - 1) \tag{5}$$

if the audit event at time t belongs to the i th event type

$$X_i(t) = \lambda * 0 + (1 - \lambda) * X_i(t - 1) \tag{6}$$

if the audit event at time t is different from the i th event type, where $X_i(t)$ is the observed value of the i th variable in the vector of an observation at time t , λ is a smoothing constant that determines k or the decay rate, and $i = 1, \dots, 284$. The most recent observation at time t receives a weight of λ , the observation at time $t - 1$ receives a weight of $\lambda(1 - \lambda)$, and the observation at time $t - k$ receives a weight of $\lambda(1 - \lambda)^k$. Hence, $X_i(t)$ represents an exponentially decaying count of event type i , measuring the

intensity of event type i in the recent past. A multivariate observation, $(X_1, X_2, \dots, X_{284})$, represents the intensity distribution of various event types.

In this study, we initialize $X_i(0)$ to 0 for $i = 1, \dots, 284$. We set λ to 0.3, a common value for the smoothing constant [21]. Fig. 2 shows the decay effect of the smoothing constant 0.3.

Hence, for each audit event in the training and testing data, we obtain a vector of (X_1, \dots, X_{284}) . For example, given the following stream of audit events:

$t = 0,$ 1, 2, 3, ...
 EventType3, EventType8, EventType1, ...

At $t = 0$, all variables in the vector of (X_1, \dots, X_{284}) have a value of 0. At time $t = 1$, X_3 has a value of

$$0.3 \quad (= 0.3 * 1 + 0.7 * 0),$$

and all other variables have a value of 0. At time $t = 2$, X_3 has a value of $0.21 (= 0.3 * 0 + 0.7 * 0.3)$, X_8 has a value of $0.3 (= 0.3 * 1 + 0.7 * 0)$, and all other variables have a value of 0. At $t = 3$, X_3 has a value of $0.147 (= 0.3 * 0 + 0.7 * 0.21)$,

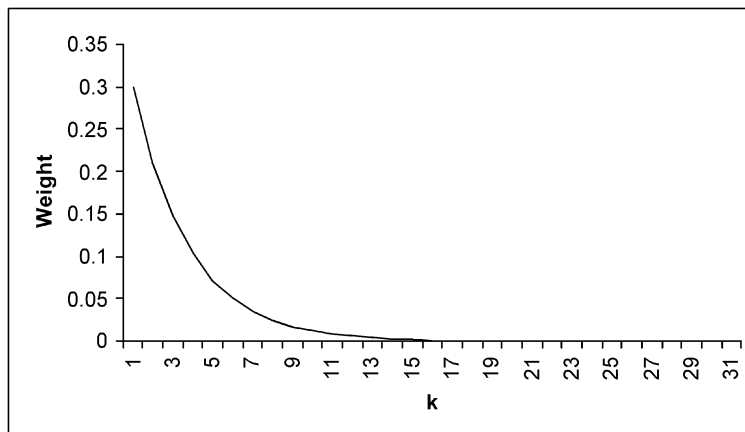


Fig. 2. The decay effect of the smoothing constant 0.3.

TABLE 2
The Statistics of the T^2 Values for Normal and Intrusive Events in the Testing Part of the Small Data Set

| Testing Data | Minimum | Maximum | Average | Standard Deviation |
|----------------|----------|----------|----------|--------------------|
| Normal data | 1.71E+00 | 1.26E+02 | 9.82E+00 | 1.41E+01 |
| Intrusive data | 2.69E+00 | 5.25E+02 | 7.39E+01 | 8.95E+01 |

X_8 has a value of 0.21 ($= 0.3 * 0 + 0.7 * 0.3$), X_1 has a value of 0.3 ($= 0.3 * 1 + 0.7 * 0$), and all other variables have a value of 0.

The long-term profile of normal activities measured by the 284 variables is captured by the sample mean vector \bar{X} and the sample covariance matrix S . The audit events for normal activities in the training data give us a sample of (X_1, \dots, X_{284}) 's to obtain the sample mean vector \bar{X} and the sample covariance matrix S .

For each of the audit events in the testing data and the corresponding observation of (X_1, \dots, X_{284}) , we compute the T^2 statistic according to formula (3). The T^2 value is small if the observation is close to the norm profile.

To determine the upper limit of T^2 in terms of $\bar{T}^2 + 3S_{T^2}$ as the signal threshold, we need to compute \bar{T}^2 and S_{T^2} . For the small data set, we use the T^2 values for the first half (first 703 audit events) of the 1,406 audit events for normal activities in the testing data to obtain \bar{T}^2 and S_{T^2} . For the large data set, we use T^2 values for the 740,995 and 1,283,903 audit events for normal activities in day-1 and day-2 data, respectively, to obtain \bar{T}^2 and S_{T^2} . The upper limit, $\bar{T}^2 + 3S_{T^2}$, is then used to determine for which events in testing data we should generate signal. For small data, we compute T^2 values for each event in the second half (last 703 audit events) of the 1,406 audit events for normal activities and 1,225 audit events for intrusive activities in the testing data and determine which events we should signal. For the large data set the testing days are day-3 and day-4 which contain 2,232,981 and 893,603 audit events for normal activities and 6,524 and 31,476 audit events for intrusive activities, respectively. The T^2 value for each of these events is compared with the signal threshold, and if it exceeds the threshold, we signal the audit event as anomalous.

There is only a small amount of audit data in the small data set. This might cause an overfitting problem. However, in the small data set only 49 event types appear. Hence, to prevent the overfitting problem, we use only 49 variables in the T^2 test and the X^2 test for the small data set.

4 RESULTS AND DISCUSSIONS

This section describes the results obtained by applying the T^2 test and the X^2 test to the small and the large sets of audit data, as described in the previous section, to test the performance of these two techniques.

4.1 Results for the Small Data Set

Table 2 summarizes the statistics (minimum, maximum, average and standard deviation) of the T^2 values for 1,406 audit events of normal activities and 1,225 audit events of intrusive activities. As shown in Table 2, the T^2 values of the audit events for normal activities are, on average,

smaller than the T^2 values of the audit events for intrusive activities. Note that the smaller the T^2 is, the closer the audit event is to the norm profile.

Using the T^2 values for the first 703 audit events for normal activities in the testing data, we obtain 10.4, 16.0 and 58.4 for \bar{T}^2 , S_{T^2} , and $\bar{T}^2 + 3S_{T^2}$, respectively. That is, the upper limit of the T^2 values for the audit events of normal activities is 58.4 as a signal threshold. If a T^2 value for an audit event is greater than 58.4, we signal this audit event as anomalous.

When we use the upper limit of 58.4 as a signal threshold to examine the T^2 values for the remaining 703 audit events of normal activities, we have signals for 15 audit events of normal activities. This indicates a 2 percent ($= 15/703$) false alarm rate by audit event. A false alarm is a signal when the audit event comes from normal activities. When we use the upper limit of 58.4 as a signal threshold to examine the T^2 values for the 1,225 audit events of intrusive activities, we have 465 signals. The detection rate by audit event is 38 percent ($= 465/1225$). To reduce the false alarm rate of the T^2 test to 0 percent, we can use 126—the maximum T^2 value of the audit events for normal activities in the testing data as a signal threshold. Using this signal threshold, we obtain the detection rate of 16 percent.

The performance of the T^2 test is not as good as the performance of the X^2 test on the same set of testing data as presented in Section 2. Using the X^2 values for the first 703 audit events for normal activities in the testing data, we obtain 6.85 for $\bar{X}^2 + 3S_{X^2}$ as the upper limit of X^2 values for normal activities. Using the upper limit of 6.85 as a signal threshold to examine the X^2 values for the remaining 703 audit events of normal activities and the 1,225 audit events of intrusive activities in the testing data, we obtain 0 percent false alarm rate by audit event and 75 percent detection rate by audit event. The pair of 0 percent false alarm rate and 75 percent detection rate from the X^2 testing results are better than the pair of 2 percent false alarm rate and 38 percent detection rate and the pair of 0 percent false alarm rate and 16 percent detection rate from the T^2 testing results.

Since an intrusion session corresponds to one intrusion, we can signal a session as intrusive when there is a signal on at least one audit event in that session. Using the upper limit of 6.85 as a signal threshold, the X^2 test yields 0 percent false alarm rate by session and 100 percent detection rate by session. Using the maximum T^2 value of audit events for normal activities in the testing data as a signal threshold, the T^2 test also achieves 0 percent false alarm rate by session and 100 percent detection rate by session. Both X^2 test and T^2 test yield ideal performance when we analyze performance by session.

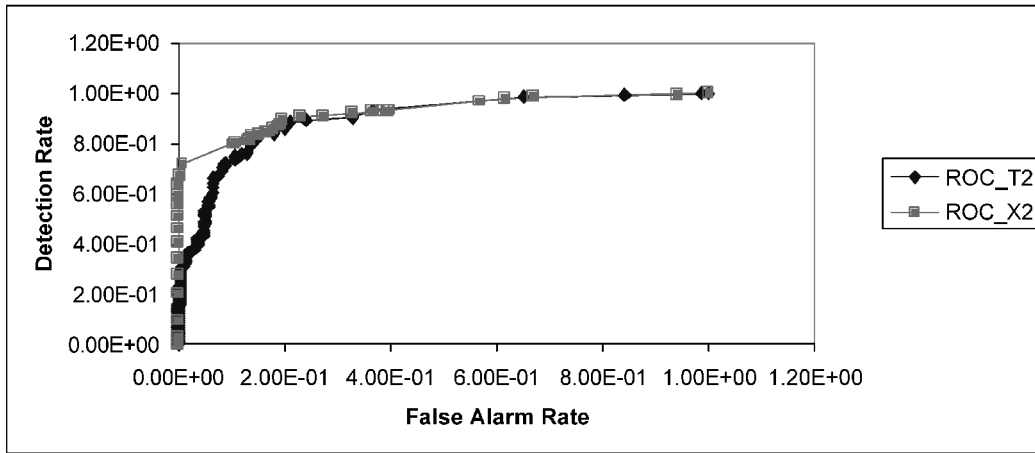


Fig. 3. The ROC curves of the T^2 test and the X^2 test for the event-wise analysis sor the small data set.

For a given test, different signal thresholds lead to different pairs of false alarm and detection rates that describe the performance of the test according to signal detection theory [28]. A Receiver Operating Characteristic (ROC) curve plots pairs of false alarm rate and detection rate as points when various signal thresholds are used. Fig. 3 shows the ROC curve of the X^2 test and the ROC curve of the T^2 test by plotting pairs of false alarm rate by audit event and detection rate by audit event. The nearer the ROC curve of a test is to the upper-left corner (representing 100 percent detection rate and 0 percent false alarm rate), the better the performance of the test is. Since the ROC curve of the X^2 test is nearer to the upper-left corner than the ROC curve of the T^2 test (see Fig. 3), the X^2 test performs better than the T^2 test in intrusion detection with respect to both false alarm rate and detection rate.

4.2 Results for the Large Data Set

Table 3 summarizes the statistics (minimum, maximum, average, and standard deviation) of the T^2 values for the testing data that include 3,174,584 audit events of normal activities and 48,000 audit events of intrusive activities in total. As shown by the statistics in Table 3, the T^2 values of the audit events for normal activities are in average smaller than the T^2 values of the audit events for intrusive activities. Note that the smaller the T^2 is, the closer the audit event is to the norm profile.

Using the T^2 values for the normal activities in the training data, we obtain $2.664802E + 04$, $4.059345E + 04$, and $1.484284E + 05$ for $\overline{T^2}$, S_{T^2} , and $\overline{T^2} + 3S_{T^2}$, respectively. That is, the upper limit of the T^2 values for the audit events of normal activities is 148,428.4 as a signal threshold. If a T^2 value for an audit event is greater than 148,428.4, we signal this audit event as anomalous.

When we use the upper limit of 148,428.4 as a signal threshold to examine the T^2 values for the audit events of normal activities during testing data, we have signals for 27 audit events of normal activities. This indicates 0.0008 percent ($= 27/3174584$) false alarm rate by audit event. When we use the upper limit of 148,428.4 as a signal threshold to examine the T^2 values for the audit events of intrusive activities during testing data, we have 142 signals. The detection rate by audit event is 0.3 percent ($= 142/48000$) only.

Using the X^2 values for the normal activities in the training data, we obtain $4.975038E + 02$ for $\overline{X^2} + 3S_{X^2}$ as the upper limit of X^2 values for normal activities. Using the upper limit of 497.5038 as a signal threshold to examine the X^2 values for the audit events of normal activities and intrusive activities in the testing data, we obtain 0.1 percent false alarm rate by audit event and 1.36 percent detection rate by audit event.

Fig. 4 shows the ROC curves from the X^2 test and the T^2 test by plotting pairs of false alarm rate by audit event and detection rate by audit event. We do not get good performance from any of these techniques. X^2 achieves 90 percent intrusion detection rate only after 40 percent false alarm rate, whereas the performance of T^2 is not good at all. Many normal events are signaled and many intrusion events are missed. Therefore event-wise analysis does not yield good performance.

We also conduct a session-wise analysis of the results. In order to do the session-wise analysis, we group the T^2 and X^2 values for the audit events according to each session and count how many of the audit events inside that session are signaled. We divide the signal count by the number of audit events in that session and call it “session signal ratio.” If the

TABLE 3
The Statistics of the T^2 Values for Normal and Intrusive Events in the Testing Part of the Large Data Set

| Testing Data | Minimum | Maximum | Average | Standard Deviation |
|----------------|----------|----------|----------|--------------------|
| Normal data | 1.30E+00 | 1.02E+06 | 4.38E+01 | 3.79E+06 |
| Intrusive data | 1.43E+00 | 5.18E+05 | 1.41E+03 | 3.83E+08 |

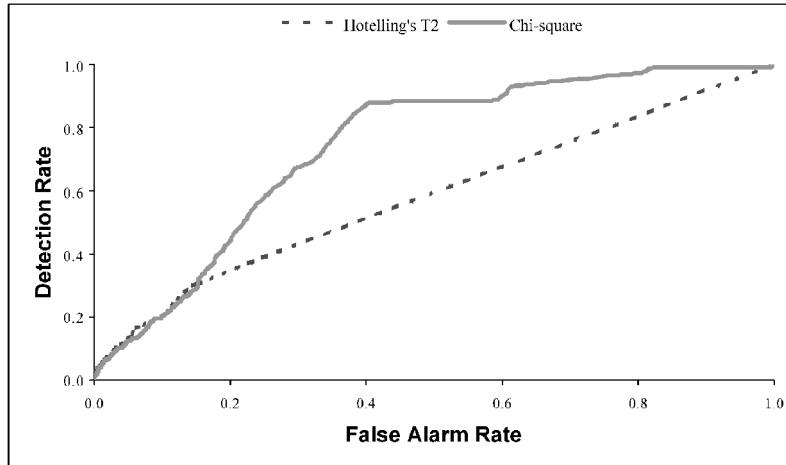


Fig. 4. The ROC curves of the T^2 test and the X^2 test for the event-wise analysis for the large data set.

TABLE 4

The Statistics of the Session Signal Ratio (SSR) for Normal and Intrusive Sessions in the Testing Part of the Large Data Set

| Testing Data | Minimum SSR | Maximum SSR | Average SSR |
|-----------------|-------------|-------------|-------------|
| Normal sessions | 0.00E+00 | 1.45E-02 | 3.23E-05 |
| Attack Sessions | 0.00E+00 | 2.40E+00 | 9.39E-01 |

session is an intrusion session, we expect a high session signal ratio. If it is a normal session, we expect the session signal ratio to be low. Table 4 summarizes the statistics about session signal ratio for the large data set for T^2 . Though the average session signal ratio for intrusion sessions is low (0.94 percent), it is significantly higher than that of the normal sessions (0.00003 percent). Therefore, if we plot the ROC curve using the session signal ratio, it will tell us how much separation we get among the session signal ratio for normal sessions and attack sessions.

Fig. 5 shows that both X^2 and T^2 tests perform very well in intrusion sessions from normal sessions through session signal ratios. The T^2 test achieves 95 percent intrusion

detection rate, whereas the X^2 test achieves 60 percent intrusion detection rate at 0 percent false alarm rate by session. But after 5 percent false alarm rate X^2 performs better than T^2 technique though the margin is not large.

4.3 Discussions

Figs. 3, 4, and 5 show that the X^2 testing results are either better than or comparable to the T^2 testing results for both the small and the large data sets. Both the T^2 test and the X^2 test detect mean shifts. The T^2 test differs from the X^2 test only in the T^2 test's additional capability in detecting counterrelationships. Considering the similarity and difference between the T^2 test and the X^2 test, the better

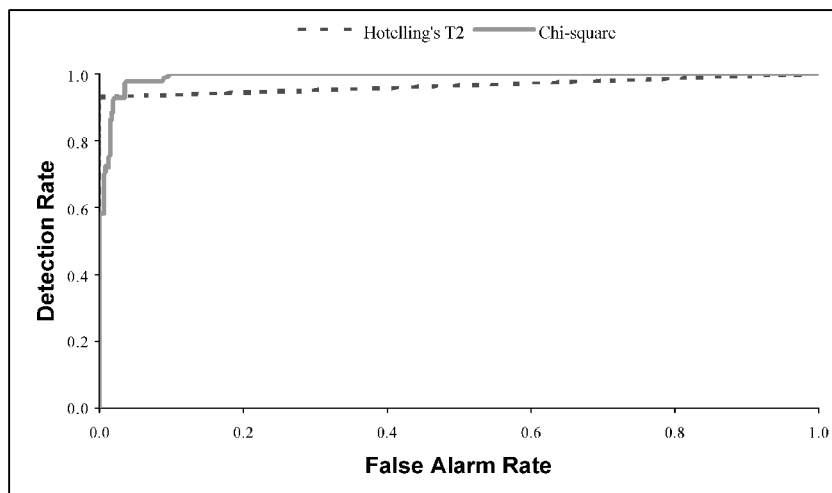


Fig. 5. The ROC curves of the T^2 test and the X^2 test for the session-wise analysis for the large data set.

or comparable performance of the X^2 test indicates two possibilities. First, intrusions manifest themselves mainly through mean shifts in the intensity distribution of various event types. Hence, the capability of the X^2 test in detecting mean shifts allows it to detect intrusions effectively. Second, the additional capability of the T^2 test in detecting countershifts might pick up some normal noises and variations leading to counterrelationships. Those counterrelationships increase the variance of the T^2 test values for audit events of normal activities, and thus make the boundary between normal activities and intrusive activities less distinctive. This is confirmed by our observation from the testing results that the variance of the X^2 test values for the audit events of normal activities is less than the variance of the T^2 test values for those audit events of normal activities.

In addition to special means of attacking information systems, intrusions usually need to use some means (e.g., commands such as *ls*) that also appear in normal activities. Hence, we should not expect an intrusion detection technique to signal every audit event in an intrusive session. However, we should expect that the number of signals during an intrusive session and the number of signals during a normal session would be largely different in general. Therefore, the session-wise analysis is expected to be more reliable than the event-wise analysis as seen in the testing results of this study.

In summary, despite of its ability to capture the correlated structure of multiple variables and detecting counterrelationships as well as mean shifts, the performance of Hotelling's T^2 test for intrusion detection is not as good as the performance of the X^2 test that detects only mean shifts. Because intrusions may manifest more through mean shifts than through counterrelationships, we can suppress noises and variations in normal activities causing counterrelationships to improve the accuracy of intrusion detection. Note that without the computation of the covariance matrix, the computational complexity of the X^2 test is much less than that of the T^2 test. Hence, it appears that a more scalable multivariate analysis technique detecting mean shifts only is sufficient for intrusion detection, possibly due to the nature of the intrusion detection problem.

ACKNOWLEDGMENTS

This work is sponsored in part by the Air Force Office of Scientific Research (AFOSR) under grant number F49620-99-1-001, and the Defense Advanced Research Project Agency (DARPA) /Air Force Research Laboratory—Rome (AFRL-Rome) under grant number F30602-99-1-0506. The U.S. government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of, AFOSR, DARPA/AFRL-Rome, or the US government. We also thank Xiangyang Li, Qiu Zhong, and Mingming Xu for their assistance in data collection and data preprocessing.

REFERENCES

- [1] DARPA Proc. *DARPA Information Survivability Conf. and Expo.* Los Alamitos, Calif.: IEEE CS, Jan. 2000.
- [2] W. Stallings, *Network and Inter-Network Security Principles and Practice*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- [3] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation," *Proc. DARPA Information Survivability Conf. and Exposition*, pp. 12-26, Jan. 2000.
- [4] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- [5] H. Debar, M. Dacier, and A. Wespi, "Towards a Taxonomy of Intrusion-Detection Systems," *Computer Networks*, vol. 31, pp. 805-822, 1999.
- [6] T. Escamilla, *Intrusion Detection: Network Security beyond the Firewall*. New York: John Wiley & Sons, 1998.
- [7] D.E. Denning, "An intrusion-detection model," *IEEE Trans. Software Eng.*, vol. 13, no. 2, pp. 222-232, Feb. 1987.
- [8] G. Vigna, S. Eckmann, and R. Kemmerer, "The STAT Tool Suite," *Proc. DARPA Information Survivability Conf. and Exposition*, pp. 46-55, Jan. 2000.
- [9] S. Kumar, "Classification and Detection of Computer Intrusions," PhD dissertation, Dept. of Computer Science, Purdue Univ., West Lafayette, Indiana, 1995.
- [10] W. Lee, S.J. Stolfo, K. Mok, "Mining in a Data-Flow Environment: Experience in Network Intrusion Detection," *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '99)*, Aug. 1999.
- [11] D. Anderson, T. Frivold, and A. Valdes, "Next-Generation Intrusion Detection Expert System (NIDES): A Summary," Technical Report SRI-CSL-97-07, Menlo Park, Calif.: SRI Int'l, May 1995.
- [12] P. Neumann and P. Porras, "Experience with EMERALD to Date," *Proc. First USENIX Workshop Intrusion Detection and Network Monitoring*, pp. 73-80, Apr. 1999. <http://www.csl.sri.com/neumann/det99.html/>.
- [13] A.K. Ghosh, A. Schwatzbard, and M. Shatz, "Learning Program Behavior Profiles for Intrusion Detection," *Proc. First USENIX Workshop Intrusion Detection and Network Monitoring*, Apr. 1999. <http://www.rstcorp.com/~anup/>.
- [14] H.S. Javitz and A. Valdes, "The SRI Statistical Anomaly Detector," *Proc. 1991 IEEE Symp. Research in Security and Privacy*, May 1991.
- [15] H.S. Javitz and A. Valdes, "The NIDES Statistical Component Description of Justification," Technical Report A010, Menlo Park, Calif.: SRI Int'l, Mar. 1994.
- [16] Y. Jou, F. Gong, C. Sargor, X. Wu, S. Wu, H. Chang, and F. Wang, "Design and Implementation of a Scalable Intrusion Detection System for the Protection of Network Infrastructure," *Proc. DARPA Information Survivability Conf. and Expo.*, pp. 69-83, 2000.
- [17] S. Forrest, S.A. Hofmeyr, and A. Somayaji, "Computer Immunology," *Comm. ACM*, vol. 40, no. 10, pp. 88-96, Oct. 1997.
- [18] C. Ko, G. Fink, and K. Levitt, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-Based Approach," *Proc. 1997 IEEE Symp. Security and Privacy*, pp. 134-144, 1997.
- [19] Y.-M. Chou, R.L. Mason, and J.C. Young, "Power Comparisons for a Hotelling's T^2 Statistic," *Comm. Statistical Simulation*, vol. 28, no. 4, pp. 1031-1050, 1999.
- [20] R.A. Johnson and D.W. Wichern, *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Prentice Hall, 1998.
- [21] T.P. Ryan, *Statistical Methods for Quality Improvement*. New York: John Wiley & Sons, 1989.
- [22] R.L. Mason, N.D. Tracy, and J.C. Young, "Decomposition of T^2 for Multivariate Control Chart Interpretation," *J. Quality Technology*, vol. 27, no. 2, pp. 99-108, Apr. 1995.
- [23] R.L. Mason, N.D. Tracy, and J.C. Young, "A Practical Approach for Interpreting Multivariate T^2 Control Chart Signals," *J. Quality Technology*, 29, no. 4, pp. 396-406, vol. Oct. 1997.
- [24] R.L. Mason and J.C. Young, "Improving the Sensitivity of the T^2 Statistic in Multivariate Process Control," *J. Quality Technology*, vol. 31, no. 2, pp. 155-164, Apr. 1999.
- [25] B.S. Everitt, "A Monte Carlo Investigation of the Robustness of Hotelling's One- and Two-Sample T^2 Tests," *J. Am. Statistical Assoc.*, vol. 74, no. 365, pp. 48-51, Mar. 1979.

- [26] R.L. Mason, C.W. Champ, N.D. Tracy, S.J. Wierda, and J.C. Young, "Assessment of Multivariate Process Control Techniques," *J. Quality Technology*, vol. 29, no. 2, pp. 140-143, Apr. 1997.
- [27] N. Ye and Q. Chen, "An Anomaly Detection Technique Based on a Chi-Square Statistic for Detecting Intrusions into Information Systems," *Quality and Reliability Eng. Int'l*, vol. 17, no. 2, pp. 105-112, 2001.
- [28] B.H. Kantowitz and R.D. Sorkin, *Human Factors: Understanding People-System Relationships*. New York: John Wiley & Sons, 1983.



Nong Ye received the PhD degree from Purdue University, West Lafayette, Indiana. She is an associate professor of industrial engineering at Arizona State University (ASU). She is the director of the Information and Systems Assurance Laboratory at ASU. Her research work focuses on assuring process quality and reliability of information systems, manufacturing and enterprise systems, and human-machine systems. She has published more than 80 journal

and conference articles. She is a senior member of the Institute of Industrial Engineers, and a senior member of the IEEE. She is a member of the editorial boards of the *International Journal of Human-Computer Interaction* and the *International Journal of Cognitive Ergonomics*.



Syed Masum Emran received the MS degree in computer science from the Department of Computer Science & Engineering, Arizona State University, in 2000. His research area was computer intrusion detection. He is currently working as a software engineer at Motorola.



Qiang Chen received the BS and MS degrees from the Manufacturing Engineering Department at Beijing University of Aeronautics and Astronautics (BUAA), Beijing, PR China, in 1993 and 1999, respectively. Since 1999, he has been a PhD student in the Department of Industrial Engineering, Arizona State University.

Sean Vilbert received the MS degree in industrial engineering at Arizona State University.

► For more information on this or any computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.