Statistical Foundations of Audit Trail Analysis for the Detection of Computer Misuse

Paul Helman and Gunar Liepins

***Abstract*— We model computer transactions as generated by two stationary stochastic processes, the legitimate (normal) process N and the misuse process M. We define misuse (anomaly) detection to be the identification of transactions most likely to have been generated by M. We formally demonstrate that the accuracy of misuse detectors is bounded by a function of the difference of the densities of the processes N and M over the space of transactions. In practice, detection accuracy can be far below this bound, and generally improves with increasing sample size of historical (training) data. Careful selection of transaction attributes also can improve detection accuracy; we suggest several criteria for attribute selection, including adequate sampling rate and separation between models. We demonstrate that exactly optimizing even the simplest of these criteria is NP-hard, thus motivating a heuristic approach. We further differentiate between modeling (density estimation) and nonmodeling approaches. We introduce a frequentist method as a special case of the former, and Wisdom and Sense, developed at Los Alamos National Laboratory, as a special case of the latter. For nonmodeling approaches such as Wisdom and Sense that generate statistical rules, we show that the rules must be maximally specific to ensure consistency with Bayesian analysis. Finally, we provide suggestions for testing detection systems and present limited test results using Wisdom and Sense and the frequentist approach.**

***Index Terms*— Anomaly detection, audit trail analysis, computer security, intrusion detection, probability theory.**

I. Introduction

Recent recognition that password authorization and other administrative and physical procedures are not sufficient by themselves to prevent misuse of computer systems has led to the development of a variety of intrusion detection systems. The usefulness of such systems was argued in Denning [3] and current implementations include, for example, IDES (Lunt *et al.* [10], [15]), Discovery (Tenner [22]), MIDAS (Sebring *et al.* [20]) and Wisdom and Sense (Vaccaro and Liepins [23] and Liepins and Vaccaro [13]).

Although some detection systems have expert systems components, many flag "unusual" or "statistically suspicious" transactions. In light of the large personnel, time, and fiscal investment in the development of such systems, surprisingly little has appeared in the literature that addresses their formal properties and limitations. The primary goal of this paper is to provide formal criteria against which existing and future detection systems can be measured and optimized; the research reported here represents a first step at formally quantifying the power and limitations of statistically based detection systems.

A. Audit Trail Analysis: Concepts and Motivations

Audit trail analysis supports an approach to intrusion detection that attempts to identify suspicious computer activities. The approach is intended to augment traditional security measures (e.g., physical and password protections) by scrutinizing the activities of all users, once they have gained access to the computing system. In principle, audit trail analysis can detect, for example, an authorized user engaged in activities prohibited to him or her (e.g., a user altering a file in an illegitimate manner) and one user masquerading as another user (e.g., one user accessing the computing system via another user's account).

Since the technique of audit trail analysis is based on the careful monitoring of users' actions, serious privacy issues must be addressed. While the law currently is evolving regarding the contexts in which such monitoring is permissible, it is agreed that, at the very least, users must be notified when their actions are monitored. Further, society must determine for what purposes such monitoring is justified since, for example, a degree of monitoring that might be justified for the purpose of increasing the security of a nuclear facility might not be justified for the purpose of ascertaining the productivity of clerical workers.

An audit trail can be maintained for a variety of user activity types, logging, for example, operating system commands, database system interrogations and updates, and the details of user interactions with specialized programs (e.g., materials accountability software). Though the methodology described here can be applied to the analysis of audit trails generated by any such activity, we shall illustrate our techniques in terms of operating system-like audit trails.

The basic unit of the audit trail is called a *transaction*. A transaction provides a trace of a primitive user action by recording attribute values that characterize the action. In practice, system designers and security experts must determine what collection of attributes to record for a given application of interest. While the identification of all potentially relevant attributes is an important aspect of the audit trail analysis problem, the current paper does not directly consider this issue. In fact, one of our goals is to develop a methodology that is completely independent of the semantics of the attributes that

Manuscript received July 1991; revised June 1993. Recommended by F. Bastani. This work is supported in part by a grant from Los Alamos National Laboratory and the NSA.

P. Helman is with the Computer Science Department, University of New Mexico, Albuquerque, NM 87131.

G. Liepins was with the Oak Ridge National Laboratory, Oak Ridge, TN 37831, and the Computer Science Department of the University of Tennessee. Dr. Liepins is deceased.

IEEE Log Number 9211723.

define transactions, and independent of how these attributes have been identified.

For simplicity, we shall assume here that all transactions in an audit trail are homogeneous, containing a common set of attributes; this set of attributes defines the *transaction template*. For example, a transaction template might include the attributes *user_id, command, port, time, elapsed_cpu*, and *status_code*. An *instance* of the transaction template (referred to as a transaction instance, or simply as a transaction) is created for each primitive user action and is appended to the audit trail. For example, an instance of a transaction template might include the following values.

user_id=Fred, command=execute program x, port=tty8,
time=1992.05.19.09.23.12.119, elapsed_cpu=1.4,
status_code=OK

While the techniques described in this paper are applied *after* the transaction templates have been defined, a major focus of our research is to investigate how to *transform* an audit trail consisting of instances of predefined transaction templates so that it becomes more amenable to statistical analysis. A typical audit trail contains transactions defined over a large number (e.g., more than 100 is not unusual) of attributes, each of whose values may be in raw form (e.g., time might be recorded in milliseconds and some attributes might assume floating point values). Consequently, any reasonably-sized training sample of transactions cannot be expected to represent accurately the entire transaction space. Data transforms such as attribute projection (i.e., the elimination of certain attributes) and value aggregation (i.e., the clustering of values) therefore are necessary to allow meaningful analysis.

B. An Overview of the Statistical Modeling Approach

We model computer transactions as generated by two stationary stochastic processes, the legitimate (*normal*) process N and the *misuse* process M. Hence, we partition the activities being monitored into two activity types: normal activities and misuse activities. This simple, binary partitioning of activities simplifies the presentation, but our results easily generalize to applications for which we wish to refine the partition. The exact specification of what constitutes normal and misuse activities is application dependent. Typically, normal activities are activities performed by an authorized user, using his or her own account, that are consistent with intent of the issuance of that account. The misuse activities typically of greatest concern include an authorized user engaged in activities prohibited to him or her (e.g., a user altering a file in an illegitimate manner), one user masquerading as another user (e.g., one user accessing the computing system via another user's account), or a user obtaining the privileges of another user, especially of a system superuser.

We define *misuse detection* to be the identification of transactions most likely to have been generated by M and demonstrate formally that the accuracy of misuse detectors is bounded by a function of the difference of the densities of the processes N and M over the space of transactions. We demonstrate further that this bound is easily achievable, provided that both processes N and M are characterized exactly, but that, in practice, detector accuracy can be far below this bound due to a lack of knowledge regarding one or both distributions.

After presenting optimality results for perfect information, we consider the opposite extreme, misuse detection in the absence of virtually all prior knowledge of the processes N and M. In practice, the designers of an audit trail system in fact utilize much prior knowledge, collecting only data which appears to be relevant and representing this data in a useful form. In addition, there may be available information such as special properties of attributes (e.g., no two transactions can have the same value for an attribute A, or attribute A is expected to assume values with a known distribution) and relationships between attributes. Further, in practice, there may be available expert rules which characterize partially normal and intrusive patterns of behavior, for example:

if *(program=P)* **and** *(elapsed_cpu > 2.3)*
then *suspect misuse*
if *(user=fred)* **and** *(port ≠ tty8* **or** *tty10)*
then *suspect misuse*

While such expert semantic information is a critical component of successful intrusion detection systems, the techniques considered in this paper are designed to utilize only a minimal amount of prior information, specifically, a sample of transactions generated by the normal process. There are several motivations for considering such techniques, including:

1) We wish to understand the limitations of, and approaches to, detection under the most severe informational restrictions. We suggest several criteria for attribute projection in such an environment, including adequate sampling rate and potential separation of models. We demonstrate that the exact optimization of even the simplest versions of such criteria are NP-hard problems.
2) In practice, we always will reach a point at which all semantic knowledge has been applied, e.g., semantic information has been applied to eliminate irrelevant attributes and to cluster data values, and expert rules have been used to classify known patterns of activity. The techniques we propose would then be applied as a "last line of defense." While we feel that the development of such heuristics is one of the most important and challenging aspects of the problem, we acknowledge that, in practice, their application should be integrated as much as possible with the application of semantic knowledge. Current research is considering how best to perform this integration.
3) We anticipate many intrusion detection applications (or applications of a similar nature) where very little expert information is available. In fact, most researchers acknowledge that it is the rule, rather than the exception, to be confronted with a paucity of misuse experiences, and hence known intrusion scenarios can provide only limited coverage. A key aspect of our approach, the employment of misuse surrogate models (see Section III-B), is meant to address this reality and expand de-

tection capabilities beyond known or conjectured misuse scenarios. Misuse surrogates are simple, generic statistical models which attempt to abstract some significant characteristics that differentiate from normal activity one or more classes of misuse activities, thereby allowing our detection algorithms to differentiate from normal behavior many classes of misuse behavior.

Hence, while expert semantic information and our approaches ideally are integrated as described in the previous point, the current state-of-the-science is such that statistical techniques appear currently to be more widely applicable. The experiences of systems such as IDES [10], [15] and Wisdom and Sense [23], which contain both statistical and expert components, support this contention. Therefore, statistical techniques such as ours which, theoretically, might appear most useful as a last line of defense must assume an even more prominent role until the semantics of intrusion scenarios are better understood.

The remainder of this paper is organized as follows. Section II details our formalism and derives our results for optimal detection. Section III considers the problem of detection under limited information and outlines a heuristic approach. Section IV summarizes another class of detection systems—a class exemplified by Wisdom and Sense developed at Los Alamos National Laboratory—that utilize a nonmodeling approach to generate statistical rules. This section demonstrates that the rules generated by any such system must be maximally specific to ensure consistency with Bayesian analysis. Finally, Section V provides suggestions for testing detection systems and presents limited test results comparing Wisdom and Sense with a simple modeling approach.

II. Detection Objectives

A. Statistical Modeling

We model the generation of computer transactions as a stationary, stochastic process

$$\mathrm{H} : \{1, 2, \cdots, \} \rightarrow S.$$

H is interpreted as mapping discrete "time units" t into a finite transaction space S, which consists of the set of all possible instances of the (fixed) transaction template of interest. The value of $\mathrm{H}(t)$ may be interpreted as the tth transaction generated by H.

Process H is specified as a mixture of two auxiliary stationary stochastic processes N (normal or legitimate transactions) and M (misuse transactions), each having the same domain and range as H, the selection of which is based on the output of still another stationary stochastic process D having the same domain and range as the other processes:

$$\mathrm{H}(t) = \begin{cases} N(t) & \text{if } D(t) = 0 \\ M(t) & \text{if } D(t) = 1 \end{cases}.$$

The three stochastic processes should be interpreted as follows. The tth transaction placed in the audit trail is generated by either a normal or a misuse activity, as is determined by stochastic process D. If $D(t) = 0$, then normal process N is invoked to generate the tth transaction and hence, in this case, $\mathrm{H}(t) = N(t)$; otherwise, $D(t) = 1$, misuse process M is invoked to generate the tth transaction, and, hence, $\mathrm{H}(t) = M(t)$.

We assume that each of the four processes is stationary and that N, M, and D are pairwise independent. That these three processes are pairwise independent is a consequence of our definitions. Processes N and M operate in isolation (think of them, for example, as generating the activities of potential computer users) and, hence, the tth transaction generated by one is independent of that of the other. Process D is independent of N and M, since it determines, without knowledge of the identities of the tth transaction generated by each of N and M, whether the tth transaction is to result from a normal or misuse activity. The assumption that the four processes are stationary simplifies the analysis, but prevents consideration of temporal patterns that the audit trail may contain; the assumption, in effect, forces us to view the audit trail as an unordered collection of transactions. Current research attempts to extend our framework so that temporal patterns can be analyzed.

Given the preceding assumptions, the following notation is well defined.

$\lambda = \Pr\{D(t) = 0\}$, for $t \in \{1, 2, \cdots, \}$. That is, λ is the *a priori* probability that the tth transaction to be generated by H is generated by the normal process. By our assumptions, this probability is independent of t, that is, independent of the order in which transactions occur.

$h(x) = \Pr\{\mathrm{H}(t) = x\}$, for $x \in S$ and $t \in \{1, 2, \cdots, \}$. That is, $h(x)$ is the probability that the tth transaction to be generated by H is x. By our assumptions, this probability is independent of t.

$n(x) = \Pr\{N(t) = x\}$, for $x \in S$ and $t \in \{1, 2, \cdots, \}$. That is, $n(x)$ is the probability that the tth transaction to be generated by N is x. By our assumptions, this probability is independent of t and can be interpreted as $\Pr\{t\text{th transaction in the audit trail is } x \mid t\text{th transaction is normal}\}=\Pr\{\mathrm{H}(t) = x|D(t) = 0\}$.

$m(x) = \Pr\{M(t) = x\}$, for $x \in S$ and $t \in \{1, 2, \cdots, \}$. That is, $m(x)$ is the probability that the tth transaction to be generated by M is x. By our assumptions, this probability is independent of t and can be interpreted as $\Pr\{t\text{th transaction in the audit trail is } x \mid t\text{th transaction is misuse}\}=\Pr\{\mathrm{H}(t) = x|D(t) = 1\}$.

It follows from our definitions and assumptions that

$$h(x) = \lambda * n(x) + (1 - \lambda) * m(x).$$

We note that in most applications λ is close to 1. One consequence of this fact is that samples of actual misuse activities are rare, necessitating the application of misuse modeling techniques such as those described in Section III-B.

Example 2.1: We illustrate by means of a simple example the previous definitions and concepts. Suppose the transaction template consists of only two attributes, *user* and *command*. Suppose that the possible values for *user* are *Fred* and *Sue*, while the possible values for *command* are *Execute* and *Edit*. Hence, the transaction space S can be represented by the set

{< *Fred, Execute*>, <*Fred, Edit*>, <*Sue, Execute*>, <*Sue, Edit*>} of ordered pairs.

Assume the following probability distributions on S. For all $t \in \{1, 2, \cdots\}$:

$$\begin{aligned}
&p[N(t) =< Fred, Execute >] \\
&\quad = n(< Fred, Execute >) = 0.0400; \\
&p[N(t) =< Fred, Edit >] = n(< Fred, Edit >) = 0.900 \\
&p[N(t) =< Sue, Execute >] \\
&\quad = n(< Sue, Execute >) = 0.0200; \\
&p[N(t) =< Sue, Edit >] = n(< Sue, Edit >) = 0.0400
\end{aligned}$$

$$\begin{aligned}
&p[M(t) =< Fred, Execute >] \\
&\quad = m(< Fred, Execute >) = 0.250; \\
&p[M(t) =< Fred, Edit >] = m(< Fred, Edit >) = 0.250 \\
&p[M(t) =< Sue, Execute >] \\
&\quad = m(< Sue, Execute >) = 0.250; \\
&p[M(t) =< Sue, Edit >] = m(< Sue, Edit >) = 0.250.
\end{aligned}$$

Then, if, for example, $\lambda = 0.900$, we have

$$\begin{aligned}
&p[\mathrm{H}(t) =< Fred, Execute >] \\
&\quad = h(< Fred, Execute >) = 0.0610; \\
&p[\mathrm{H}(t) =< Fred, Edit >] = h(< Fred, Edit >) = 0.835 \\
&p[\mathrm{H}(t) =< Sue, Execute >] \\
&\quad = h(< Sue, Execute >) = 0.0430 \\
&p[\mathrm{H}(t) =< Sue, Edit >] = h(< Sue, Edit >) = 0.0610. \square
\end{aligned}$$

The objective of misuse detection is to identify those transactions $x \in S$ that are likely to be misuse, that is, transactions x for which

$$\Pr\{D(t) = 1 | \mathrm{H}(t) = x\}$$

is above some threshold or is large relative to the probability for other transactions.

We observe that our problem shares many similarities with the problem of signal detection (e.g., [7]). Indeed, many of our error measures and the methods of Section II-B and II-C, which apply when processes N and M are characterized exactly, are consistent with results from the signal detection literature. Approaches to the two problems diverge, however, when these characterizations are poor or unavailable. In particular, the techniques considered in Sections III and IV (e.g., attribute projection, value aggregation, the use of simple misuse model surrogates, and statistical rule generation) appear to have no direct analogues in the signal detection problem domain.

B. Detectors

According to Bayes theorem and our definitions

$$\begin{aligned}
&\Pr\{D(t) = 1 | \mathrm{H}(t) = x\} \\
&= \frac{\{\Pr\{\mathrm{H}(t) = x | D(t) = 1\}(1 - \lambda)}{\Pr\{\mathrm{H}(t) = x | D(t) = 1\}(1 - \lambda) + \Pr\{\mathrm{H}(t) = x | D(t) = 0\}\lambda} \\
&= \frac{m(x)(1 - \lambda)}{m(x)(1 - \lambda) + n(x)\lambda} = \frac{r(x)(1 - \lambda)}{r(x)(1 - \lambda) + \lambda} \\
&= \frac{r(x)}{r(x) + \lambda/(1 - \lambda)} \qquad (1)
\end{aligned}$$

where $r(x) = m(x)/n(x)$ if $n(x) \neq 0$. If $n(x) = 0$ and $m(x) \neq 0$, we define $r(x) = \infty$ and the last quotient of (1) to be identically 1. If $n(x) = 0$ and $m(x) = 0$, we define $r(x) = 1$. We derive immediately that $\Pr\{D(t) = 1 | \mathrm{H}(t) = x\} > \tau$ iff $r(x) > \tau\lambda/(1 - \tau)(1 - \lambda)$.

In practice, the maximum number of transactions that may be flagged during any time interval (the flagging threshold) as well as the minimum value of the ratio $r(x) = m(x)/n(x)$ of interest are specified by the System's Security Officer (SSO). Thus, in practice, what we hope to flag are the transactions x with the largest ratios $r(x)$.

Example 2.2: Assuming the probabilities given in Example 2.1, we can compute, for example,

$$\begin{aligned}
r(< Fred, Edit >) &= 0.250/.900 = 0.278 \\
r(< Sue, Execute >) &= 0.250/.0200 = 12.5.
\end{aligned}$$

It can be verified that among the transactions in the space S, $< Sue, Execute >$ has the largest ratio (and hence should be considered to be the most suspicious transaction), while $< Fred, Edit >$ has the smallest ratio (and hence should be considered to be the least suspicious transaction). In practice, of course, some or all of the probabilities necessary for these calculations will be unknown; estimation techniques are discussed in Section III. □

We define graded MD_g and binary MD_b misuse detectors as functions from S into the nonnegative reals and the binary set {0,1}, respectively. A graded detector MD_g provides a ranking of the transactions in S: The larger the value $MD_g(x)$, the more suspicion detector MD_g attributes to transaction x. A binary detector MD_b provides an absolute classification of each transaction $x \in S$: When $MD_b(x) = 0$, x is classified as normal, while $MD_b(x) = 1$ classifies x as misuse.

Example 2.3: If the required probability distributions were available, we could define a graded detector by

$$MD_g(x) = r(x).$$

Note that, as a consequence of identity (1), $MD_g(x) > MD_g(y)$ iff $\Pr\{D(t) = 1 | \mathrm{H}(t) = x\} > \Pr\{D(t) = 1 | \mathrm{H}(t) = y\}$. Similarly, if the prior probability λ additionally were available, we could define a binary detector by

$$MD_b(x) = \begin{cases} 0 & \text{if } r(x) \leq \lambda/(1 - \lambda) \\ 1 & \text{otherwise} \end{cases}.$$

As a consequence of the equivalence following (1), $MD_b(x) = 1$ iff $\Pr\{D(t) = 1 | \mathrm{H}(t) = x\} > 0.5$. □

Generalizing Example 2.3, we see immediately that to any graded misuse detector MD_g we can associate a (nonunique) ordered finite family $(MD_{b,\tau})$ of binary detectors satisfying

$$MD_{b,\tau}(x) = \begin{cases} 0 & \text{if } MD_g(x) < \tau \\ 1 & \text{otherwise} \end{cases}.$$

C. Detection Effectiveness

In general, processes N and M overlap, that is, S contains transactions x for which both $n(x)$ and $m(x)$ are nonzero. Consequently, since any misuse detector (graded or binary) must map to the same value any two occurrences of identical transaction instances x (one of which might be generated by N and the other by M) some error (incorrect ranking or classification) is unavoidable.

In this section, we demonstrate that the effectiveness of any misuse detector is limited by the disparity of the distributions n and m. When the distributions are highly similar, no detector can reliably distinguish between legitimate and misuse transactions. Conversely, when the distributions are highly dissimilar, in principle, the transactions can be distinguished with high reliability.

Consider a misuse detector MD (either graded or binary). We define weighted symmetric error as the sum

$$\sum_{x\in S}(a\max\{MD(x)-\frac{(1-\lambda)m(x)}{h(x)},0\} + b\max\{\frac{(1-\lambda)m(x)}{h(x)}-MD(x),0\})*h(x). \quad (2)$$

Since it follows from Bayes Theorem (1) that the quantity $((1-\lambda)m(x))/h(x)$ is equivalent to $\Pr\{D(t)=1|\mathrm{H}(t)=x\}$, the first term of the expression (2) is seen to measure overestimation of error and the second term underestimation. Further note that, in particular, when MD is a binary detector and $MD(x)=0$, the error term for $x\in S$ is $\Pr\{D(t)=1|\mathrm{H}(t)=x\}$, while this term is $\Pr\{D(t)=0|\mathrm{H}(t)=x\}$ when $MD(x)=1$. Because frequently these types of errors are not considered to be of equal importance we have weighted them by the constants a and b, respectively.

By the law of large numbers,

$$\frac{(1-\lambda)m(x)}{h(x)}=\lim_{T\to\infty}\frac{1}{T}\sum_{\substack{t=1\\ \ni\mathrm{H}(t)=x}}^{T} D(t)$$

where the notation $\sum_{\substack{t=1\\ \ni\mathrm{H}(t)=x}}$ indicates a sum over a subsequence of transactions, each of whose identity is x. Thus, the sum (2) can be written as

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} a\max\{MD\circ\mathrm{H}(t)-D(t),0\} + b\max\{D(t)-MD\circ\mathrm{H}(t),0\}. \quad (3)$$

Using the identities $r(x)=m(x)/n(x)$ and $h(x)=\lambda*n(x)+(1-\lambda)*m(x)$ we write

$$\frac{(1-\lambda)m(x)}{h(x)}=\frac{(1-\lambda)m(x)}{\lambda n(x)+(1-\lambda)m(x)} = \frac{(1-\lambda)r(x)}{\lambda+(1-\lambda)r(x)}=\frac{r(x)}{r(x)+\lambda/(1-\lambda)}.$$

From this we see that for any binary detector MD_b, weighted symmetric error is minimized whenever

$$MD_b(x)=\begin{cases}0 & \text{if } r(x)\le a\lambda/(1-\lambda)b\\ 1 & \text{otherwise}\end{cases}.$$

Henceforth, we deal only with the equally weighted case; the unequal case follows analogously.

For binary detectors, we can separate symmetric error into two types of error, type I (lack of sensitivity) and type II (false alarm). Analogously to (3), we can write these errors as limits of finite sums

$$\text{type I error: } \lim_{T\to\infty}\frac{1}{T}\sum_{t=1:D(t)=1}^{T} |MD_b\circ\mathrm{H}(t)-1|$$

$$\text{type II error: } \lim_{T\to\infty}\frac{1}{T}\sum_{t=1:D(t)=0}^{T} MD_b\circ\mathrm{H}(t)$$

where the sums are taken over subsequences of transactions generated by M and N, respectively. It follows trivially that symmetric error equals λ*type II error + $(1-\lambda)$ * type I error.

The following theorem establishes that symmetric error is bounded below by a function of the disparity of the distributions m and n.

Theorem 1: For any binary misuse detector MD_b, the symmetric error is bounded below by

$$\sum_{x\in S}\min(\lambda n(x),(1-\lambda)m(x)). \quad (4)$$

Further, this bound on symmetric error is achieved by the binary misuse detector defined as

$$MD_b(x)=\begin{cases}0 & \text{if } r(x)\le\lambda/(1-\lambda)\\ 1 & \text{otherwise}\end{cases}.$$

Proof: $\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}|MD_b\circ\mathrm{H}(t)-D(t)| = (1-\lambda)\sum_{x\in S}(1-MD_b(x))m(x)+\lambda\sum_{x\in S}MD_b(x)n(x)$.

Direct algebraic manipulation verifies that MD_b achieves this lower bound when it is defined as in the theorem's statement. □

III. Detecting Misuse When the Information is Imperfect

The previous sections introduced binary and graded misuse detectors and specified optimality conditions. Unfortunately, the optimal detectors introduced in the previous sections are of somewhat limited practical applicability in that they assume that misuse detection is to be performed with the benefit of perfect information. In particular, the optimal detectors require:

1) Knowledge of the *a priori* values λ and $(1-\lambda)$ on the two processes.
2) Knowledge of the distribution of the normal process, that is, $n(x)$ for all transactions $x\in S$.
3) Knowledge of the distribution of the misuse process, that is, $m(x)$ for all transactions $x\in S$.

We consider now the implications of this information not being readily available.

A. Good Estimates of λ are not Readily Available

In the previous section we defined and analyzed detectors which yield a value—either continuous or binary—indicating whether each transaction encountered should be flagged. Symmetric error measures in terms of deviation from $\Pr\{D(t) = 1|\mathrm{H}(t) = x\}$ how well the detector can be expected to perform.

In this theoretical development, the likelihood ratio $r(x) = m(x)/n(x)$ played a central role in optimal detection. We show now that in practice as well as in theory the ratio is pivotal.

Definition 1: A graded detector MD_g is *consistent* with the ratio r if, for all transactions $x_i, x_j \in S$, $r(x_i) < r(x_j) => MD_g(x_i) < MD_g(x_j)$. A binary detector MD_b is *consistent* with the ratio r if, for all transactions $x_i, x_j \in S$, $r(x_i) < r(x_j) => MD_b(x_i) \leq MD_b(x_j)$.

Results of the previous section imply that for any value of λ, the optimal detector, binary or graded, will be consistent with r. However, without knowledge of λ, it is impossible to construct an optimal MD. For example, in the case of a binary detector, it is impossible to determine, for the unknown λ in question, the optimal threshold of r-values beyond which $MD_b(x) = 1$.

Without knowledge of λ, we must shift slightly our detection goals. While we cannot make a judgement involving the absolute probability that a transaction x is generated by the misuse process M, we aspire to make *relative* judgements regarding the set of transactions encountered over some interval of time. For example, we might desire to do the following:

a) Rank from most to least suspicious the transactions encountered in a given time interval, permitting the SSO to prioritize the transactions he/she considers carefully, and to investigate as many as he/she wishes. Supplied with such information, the SSO, for example, at the end of each business day could investigate in decreasing order of suspicion the day's transactions. In this manner, the SSO could pursue as many transactions as time permits, or could terminate the day's investigation upon reaching a point in the ranking at which the transactions appear normal.

b) Given a fixed k, identify the k most suspicious transactions in a time interval. This information is similar to a), but is based on a security policy that exactly k transactions will be investigated per time interval (e.g., per business day). Note that, in principle, b) is easier to accomplish than a), since the information required for b) is derivable from that of a).

c) Given a fixed α, identify a mass α collection of transactions with the property that no transaction outside the collection is more suspicious than a transaction in the collection. This goal does require that we estimate $h(x)$ which, in turn, requires knowledge of λ. However, it often is reasonable to estimate $h(x)$ as $n(x)$ (i.e., because λ is often believed close to 1). This information supports a more dynamic version of b). For example, a security policy might be to investigate the most suspicious 1% of all computer activities encountered (e.g., $\alpha = .01$).

We present now a second measure of a detector's performance, one which is appropriate in the context of these goals. Though the measure is most appropriately applied to a graded detector, for the sake of full generality, we define it for any detector, graded or binary.

Definition 2: Let MD be a misuse detector and let the processes N, M, H, and D be understood. The *prioritization penalty* for MD's evaluation of the unordered pair $\{x, x'\}$ of transactions is defined as follows.

$$PEN(MD, \{x, x'\}) = \begin{cases} \Pr\{D(t) = 1|\mathrm{H}(t) = x\} * \Pr\{D(t') = 0|\mathrm{H}(t') = x'\} \\ \qquad\qquad\qquad\qquad \text{if } MD(x) < MD(x') \\ 1/2 * \Pr\{D(t) \neq D(t')|\mathrm{H}(t) = x \wedge \mathrm{H}(t') = x'\} \\ \qquad\qquad\qquad\qquad \text{if } MD(x) = MD(x') \end{cases}\cdot$$

MD's prioritization penalty on a transaction pair $\{x, x'\}$ for which $MD(x) < MD(x')$ is the probability that x is generated by the misuse process while x' is generated by the normal process; hence, MD's penalty on such a pair $\{x, x'\}$ measures the probability that the detector incorrectly ranks the suspiciousness of the two transactions, and hence that it misdirects the SSO. The penalty on a transaction pair $\{x, x'\}$ for which $MD(x) = MD(x')$ reflects the fact that in this case MD gives the SSO no guidance, forcing the SSO to use some other means to rank these transactions. Since we do not wish to assume any knowledge regarding what these other means might be, we postulate that the SSO has a 50% chance of misranking a pair of transactions not ordered by MD, whenever these transaction in fact are generated by different processes.

The *prioritization error* of a detector MD is the sum of its prioritization penalties weighted over all transaction pairs, and for ranking serves as the counterpart of symmetric error.

Definition 3: The *prioritization error* of a misuse detector MD is

$$\sum_{\{x,x'\}\subset S} PEN(MD_g, \{x, x'\})h(x)h(x'). \tag{5}$$

The following theorem demonstrates that just as when λ is known and minimization of symmetric error is the goal, a graded detector must be consistent with r to minimize prioritization error. As is discussed in the sections which follow, the result has much practical significance for the design of actual misuse detection algorithms.

Theorem 2: A graded detector MD_g minimizes prioritization error if and only if MD_g is consistent with r.

Proof: First rewrite

$$\begin{aligned} &\sum_{\{x,x'\}\subset S} PEN(MD_g, \{x, x'\})h(x)h(x') \\ &\quad= \sum_{\substack{\{x,x'\}\subset S \\ x\neq x'}} PEN(MD_g, \{x, x'\})h(x)h(x') \\ &\quad\quad+ \sum_{x\in S} PEN(MD_g, \{x, x\})h(x)^2. \end{aligned} \tag{6}$$

The contribution of the second term to the right hand side is independent of MD_g; hence, we can limit our attention to the first term. Since the error is linear over pairs of transactions, we can limit our attention to a single pair $\{x, x'\}$.

Suppose first that $r(x') > r(x)$. Observe that, in this case, it follows from (1) that for any $0 < \lambda < 1$

$$\Pr\{D(t') = 1|\mathrm{H}(t') = x'\} > \Pr\{D(t) = 1|\mathrm{H}(t) = x\}$$

and, equivalently,

$$\Pr\{(D(t) = 0|\mathrm{H}(t) = x\} > \Pr\{(D(t') = 0|\mathrm{H}(t') = x'\}.$$

For the pair $\{x, x'\}$, MD_g is consistent with r if and only if the contribution to the prioritization error by the pair is

$$\Pr\{D(t') = 0|\mathrm{H}(t') = x'\}\Pr\{D(t) = 1|\mathrm{H}(t) = x\}h(x)h(x'). \tag{7}$$

If for the pair $\{x, x'\}$ MD_g is inconsistent with r, then either $MD_g(x') < MD_g(x)$ or $MD_g(x') = MD_g(x)$. In the first case, the contribution to the prioritization error is

$$\Pr\{D(t') = 1|\mathrm{H}(t') = x'\}\Pr\{D(t) = 0|\mathrm{H}(t) = x\}h(x)h(x'). \tag{8}$$

In the second case, it is the average of the terms (7) and (8). Since (8) is larger than (7) whenever $r(x') > r(x)$, the required result follows. A symmetric argument applies when $r(x) > r(x')$. When $r(x) = r(x')$, we need only observe that all detectors are consistent with r on such pairs and since, in this case (7) is equal to (8), such pairs contribute the same error to all detectors. □

While prioritization error is most appropriately applied to a graded detector, the following result is of some interest.

Corollary 1: If a binary detector MD_b has minimal prioritization error among the set of all binary detectors, then MD_b is consistent with r.

Proof: Follows immediately from Theorem 2. □

Notice that, for binary detectors, the implication of Theorem 2 holds in only one direction since, for example, a binary detector MD_b which is identically zero on the transaction space is consistent with r.

B. Unknown Distributions of the Processes

To this point, we have proceeded under the implicit assumption that the distributions n and m are reliably estimable and that therefore r can be determined. The degree to which this assumption is valid has a large potential bearing on our detection capability.

Approaches for estimating n include neural networks (Jones *et al.* [11], Poggio [17]; Qian *et al.* [18]), Parzen windows (Parzen [16]), nonparametric methods (Loftsgaarden and Quesenberry [14]), projection-pursuit techniques (Friedman *et al.* [5]), k-nearest neighbor methods (Duda and Hart [4]), pseudo-Bayes estimators (Bishop *et al.* [1]), modified frequency estimators (Good [8]), and the frequentist estimator (Liepins and Vaccaro [13]). All but the last three of these methods are best suited to continuous variables. For purposes of this paper, we will restrict our attention to the frequentist estimator. The frequentist estimator simply estimates $n(x)$ as the quotient of the number of occurrences of x in the historical database DB (which we assume is generated by process N and has been screened to eliminate contaminants such as missing, erroneous, or imprecise attribute values) divided by the total number of occurrences of all transactions. As we shall see below, characteristics of our problem (e.g., a small sample DB relative to the size of the transaction space S) mandate that the frequentist estimator be used in conjunction with data transformation techniques such as attribute projection and data aggregation. Such estimators treat as equivalent transactions whose values on designated sets of attributes (i.e., sets on which we project) fall into the same aggregate of data values; the hope is that we can obtain meaningful frequency estimates for the induced transaction equivalence classes based on the total number of occurrences of transactions from each class. The subsections which follow contain details of this approach.

In contrast, historical misuse data appear not to be widely available and hence m cannot be estimated directly.[1] Some researchers approach this problem by constructing from semantic information and whatever documented misuse activities may be available application specific misuse models. Instead, we pursue an approach that utilizes simple generic models as ***surrogates*** for actual misuse models. While no claim is made that these simple surrogates resemble closely actual misuse activity, the hope is that each surrogate employed will abstract some significant characteristics that differentiate from normal activity one or more classes of misuse activity. In this way, a detector that estimates a collection $\bar{r}_i(x)$ of ratio values, each using some surrogate M_i to derive an approximation $\bar{m}_i(x)$ for the numerator, is able to differentiate from normal behavior many classes of misuse behavior. This appears to be a reasonable approach, especially in the context of the severely information-limited application environments in which we often must operate.

We generate surrogate misuse models M by applying to the normal process N simple functional relationships. In particular, an estimate $\bar{m}$ is generated from the functional $F[\cdot]$ and the estimate $\bar{n}(\cdot)$ by the relation $\bar{m}(\cdot) = F[\bar{n}(\cdot)]$. The concept is best illustrated by means of example.

Example 3.1: Two promising misuse surrogates are the ***uniform*** and ***independence*** models. The uniform model assumes that all transactions (whether historically seen or unseen) are equally likely. The functional used in this case is simply the constant relation

$$\bar{m}(x) = F[\bar{n}(\cdot)] = \frac{1}{|S|}, \qquad \text{for all } x \in S.$$

The independence model treats as independent the distributions of the individual attributes of a transaction, with the marginal probabilities derived from those of process N. That is, for any L-attribute transaction $x =< x_1, \cdots, x_i, \cdots, x_L >$ and marginal probabilities $n_i(x) = n(y|y_i = x_i)$ (i.e., $n_i(x)$ denotes the probability that the ith attribute A_i of an arbitrary transaction generated by N will have value equal x_i), $m(x)$ is the product

$$m(x) = \Pi_{i=1}^{L} n_i(x).$$

The functional used to approximate the independence model derives as follows from the observed marginal distributions $\bar{n}_i(x)$ the estimate $\bar{m}(x)$.

$$\bar{m}(x) = F[\bar{n}(\cdot)] = \Pi_{i=1}^{L}\bar{n}_i(x). \qquad □$$

[1] Were historical misuse data available, any proposed misuse model would be at least partially testable.

Example 3.2: We illustrate the consequences of the uniform and independence surrogates by means of a small two attribute example. Suppose that transaction templates consist of the attributes *user* and *command*, with respective value sets {Fred, Sue} and {execute, edit}. Suppose further that the historical database DB consists of 100 transactions, yielding observed frequencies of the four possible transactions as shown below.

	execute	edit
Fred	4	90
Sue	2	4

The frequentist estimator $\bar{n}$ of N computes, for example, $\bar{n}$(<Fred, execute>) = 0.0400 and $\bar{n}$(<Sue, execute>) = 0.0200.

When the uniform model is used for the misuse surrogate, we obtain for any $x \in S$ $\bar{m}_U(x)$ = 0.250, and hence $\bar{r}_U$<Fred, execute> = 6.25 and $\bar{r}_U$<Sue, execute> = 12.5. Therefore, under the uniform misuse surrogate, <Sue,execute> is deemed to be the most suspect transaction simply because it is the least frequent in the sample. More generally, the uniform misuse surrogate has the property that it is consistent with *anomaly detection*, where anomaly detection is defined to be the problem of flagging a collection of the transactions least likely to have been generated by N, without reference to a misuse process M. Many detectors reported in the literature (e.g., [10], [23]) appear to be performing anomaly detection, without making explicit the misuse model.

When the independence model is used for the misuse surrogate, we obtain

$$\bar{m}_I(<\text{Fred,execute}>) = (0.940) * (0.0600) = 0.0564$$

$$\bar{m}_I(<\text{ Sue,execute}>) = (.0600) * (0.0600) = 0.00360$$

$$\bar{r}_I(<\text{ Fred, execute}>) = 1.41; \bar{r}_I<\text{Sue, execute}> = 0.180.$$

Therefore, under the independence surrogate, <Fred,execute> is deemed the most suspicious transaction because Fred is doing something *relatively unusual, for him.* Clearly, each of these misuse surrogates detects better than the other certain classes of misuse. □

Preliminary experimental results indicate that the use of these surrogates in our detection algorithms allow us to distinguish rather well between normal and misuse transactions, for a wide range of simulated N and M processes. While we believe simple misuse surrogates such as those described above to be reasonable starting points, we envision a framework in which several surrogates are used as the basis for competing detectors which are combined to yield a single evaluation. We envision further an approach in which learning techniques are applied to modify the models in response to feedback on the detectors' performance. The hope is that for each target application, a collection of simple, generic models can evolve into models most appropriate for the application.

Consider now the accuracy of the estimate $\bar{r} = \bar{m}/\bar{n}$ under the simplifying assumption that the exact functional relationship $M = F[N]$ is known. In this case, the major obstacle to accurately estimating r is that posed by a finite sample.

a) *The mass of transactions not represented in the historical database DB is large.* All $x \notin DB$ will have equal estimated densities, i.e., $\bar{n}(x) = 0$, forcing the conclusion that either $\bar{r}(x) = \infty$ or, in the case that $\bar{m}(x) = 0$, that $\bar{r}(x) = 1$. Hence, we cannot rank the transactions within each of these two categories, though the r-values of transactions not in DB actually may be quite different. Further, in the case the ratio is estimated as ∞, we are forced to classify the unseen transactions as among the most suspicious. If the total mass of this collection of transactions is large, we are led to flag an unacceptably large number of suspicious transactions.

b) More generally, *the difference between* $\bar{n}(x)$ and $\bar{n}(x')$ may be small, even for x, x' represented in the historical transactions. Consequently, we may have little confidence in the prioritization by $\bar{r}$-values.

In principle, an infinite sample of historical transactions addresses both of these problems. In practice, a sample of sufficient size to be representative of the event space in its raw form ($|S|$ might be on the order of 10^{100} potential transactions) generally is unobtainable. We suggest that attribute projection and value aggregation (considered in the subsections which follow) are two approaches to transforming the raw event space in order to overcome sample size limitations and the attendant inability to estimate the ratio r. In general, there are many sets of attributes that could be selected and many ways to aggregate their values. We desire our solutions to result in: a) distinguishability between the induced measures n and m; b) good spread of the r-values; c) preservation of the probability structure of S (were full information available); and, d) small mass of unseen transactions. Estimating the mass of the unseen transactions has been addressed by a number of authors including Good [8] and Robbins [19]. Robbins showed that for independent trials generating a sample Z of cardinality k, an unbiased estimator of the mass of the unseen transactions is $1/(1+k)$ times the number of singleton transactions in Z', where $Z' = Z$ augmented with one additional randomly drawn transaction.

Attribute Selection: Loosely speaking, the more (nondependent) attributes, the more likely that each transaction is unique, and the less likely that any fixed sized historical database represents a substantial mass of all transactions. That is, any density estimator becomes less reliable. We seek "characterizing" subsets B of the set A of attributes present in the raw transactions, and project onto these subsets. This induces an equivalence class structure[2] on the set S and the detection problem is subsequently solved solely in terms of the equivalence classes: transactions x and x' are defined to be equivalent if $x[B] = x'[B]$, that is, if x and x' agree on their values of all the attributes in the projection set B.

As we have indicated, we are interested in projecting onto characterizing subsets B, that is, subsets with respect to which densities are reliably estimatable, the two processes N and M distinguishable, and which preserve the *probability structure* of the original transaction space in the sense that for all $x, x' \in S$,

$$r(x) < r(x') \leftrightarrow r(x[B]) < r(x'[B]).$$

In practice, these objectives may not be simultaneously satisfiable; in this paper, we suggest them only as heuristics.

[2] with induced probability measures.

Example 3.3: Consider transactions with two fields (or sets of fields), J and K. Suppose that process N is such that J and K are generated independently and that:

K has 2^k possible values, and the distribution of these values is uniform.

J has five values $v_1, \cdots, v_4, v_5$. J assumes v_i with probability $1/2^i$, $i = 1, 2, 3, 4$, and v_5 with probability with probability $1/2^4$.

Notice that the distribution of transactions is such that $n(x) = 1/2^{(k+i)}$, if $x[J] = v_i$, $i = 1, 2, 3, 4$. Suppose further that process M is governed by the uniform model.

For k sufficiently large and reasonably-sized samples DB, these underlying processes exhibit undesirable properties:

a) A high proportion of the space will be unseen in DB. Notice that for all $x \notin DB$, $\bar{r}(x) = \infty$.

b) A high proportion of those transactions that are in DB are likely to be singleton and doubletons. Hence, we have little separation of the $\bar{r}$-values, even for transactions which we have seen, and therefore little confidence in the resulting rankings.

Consider now the effect of projecting onto J, that is, identifying transactions with the same values in the attributes J, ignoring the K-values. Now even a modest-sized DB will yield good estimates of the distribution of N relative to the projection; observe that the projection set preserves the probability structure in the sense defined above. Hence, we can better solve the transformed detection problem. □

Of course, we have performed a bit of reverse engineering by assuming specific N and M models and then identifying good projection sets under these assumptions; in practice, we must identify good sets without prior knowledge of N and M. The point, however, is that for many seemingly natural processes, attribute projection is of tremendous value. We shortly shall consider heuristics for selecting projections when N and M are unknown.

Value Aggregation: Another technique for forming equivalence classes is value aggregation. Value aggregation partitions the domain of one or more attributes A into collections V_i of values. The limiting case of a single value class for an attribute is equivalent to projecting on the complement of that attribute, i.e., projecting so that the attribute is eliminated. Wisdom and Sense implements value aggregation (see Vaccaro and Liepins [23] and Section V of the current paper) but only for one attribute at a time and only for numeric attributes. In contrast, we consider the fully general case.

Example 3.4: As an example of the utility of value aggregation, observe that only rarely will two transactions have exactly the same values for login times monitored by month, day, hour, minutes, and hundredths-of-a-second. Hence, the distribution of these transactions is not well estimable. However, login times monitored by month, day, and hour would probably yield estimable distributions. Furthermore, it seems reasonable to conjecture that for any misuse model of interest, the rankings of r-values is invariant under such aggregation. More generally, when the result of attribute projection leaves attributes which assume continuous (e.g., floating point) values, value aggregation must partition the real number line in a manner that reduces the number of singleton transactions, while preserving the probability structure of the original transaction space. □

Heuristics to find Good Equivalence Relations: The determination of good equivalence relations first requires practical criteria to evaluate candidate relations, and second, a practical means to search the space of possible relations for the superior ones (in terms of the given criteria). We expect that both steps will depend on heuristics.

One approach to the selection of equivalence relations is to use different combinations of criteria to generate equivalence classes which later are analyzed more closely. It is quite plausible that a useful procedure would be to proceed with multiple misuse detectors, one for each equivalence class and misuse model considered, and combine the results through an adaptively weighted combination.

For projections, reasonable criteria might include some of those previously suggested: a) distinguishability between the induced measures n and m; b) good spread of the r-values; c) preservation of the probability structure; d) small estimated mass of unseen transactions.

Even given clear, unambiguous criteria, the problem of finding the best equivalence relations is far from trivial. For example, below we define a *singleton reduction problem* which requires the construction of a set of attributes with respect to which projection most effectively reduces the number of singleton transactions. We demonstrate that singleton reduction is NP-complete, implying that its solution, and therefore the optimization of criteria which depend on singleton reduction (e.g., reducing the mass of unseen transactions as estimated by the techniques of Good and Robbins [8], [19]), almost certainly will be limited to heuristic approximation.

Theorem 3: The *Singleton Reduction Problem* (*SRP*) defined as follows is NP-complete.

INSTANCE: Database DB of D transactions, each consisting of L attributes, and positive integers $K \leq L$, $s \leq D$.

QUESTION: Is there a subset A consisting of K or more attributes such that the number of singleton transactions, when DB is projected onto A, is s or fewer?

Proof: To see that SRP is in NP, observe that in time polynomial in the length of the problem instance we can guess a size-K collection A of attributes and count the number of singletons with respect to the projection onto A. To establish that SRP is NP-complete, we reduce to it *Balanced Complete Bipartite Subgraph* (BCBS) [6;GT24]. BCBS is defined as follows.

INSTANCE: Bipartite graph $G = (V, E)$, positive integer $K \leq |V|$.

QUESTION: Are there two disjoint subsets $V_1, V_2 \subseteq V$ such that $|V_1| = |V_2| = K$ and such that $u \in V_1$ and $v \in V_2$ implies that $u, v \in E$?

Transform an arbitrary instance of BCBS to the following instance of SRP. (Without loss of generality, assume that $K > 1$; if $K = 1$, simply transform to any "YES" instance of SRP if $E \neq \emptyset$ and to any "NO" instance if $E = \emptyset$.)

DB consists of $|V|$ transactions, each over $|V|$ attributes, i.e., $D = L = |V|$. We denote the transactions by $t_1, \cdots, t_D$ and the attributes by $A_1, \cdots, A_D$.

For $1 \leq i,j \leq D$ and $i \neq j$, $t_i[A_j] = 1$ iff $\{v_i, v_j\}$ is an edge in E. For each other $1 \leq i,j \leq D$ (including whenever $i = j$), assign $t_i[A_j]$ a value that appears nowhere else in DB.

K has the same value as in the BCBS instance.

s is $D - K$.

Intuitively, DB is obtained from the $D \times D$ adjacency matrix M of G, once M is modified so that each entry of M which is zero, and each entry on the diagonal, is changed to an integer value appearing nowhere else in M. The resulting matrix M' corresponds directly to DB: each row t_i is a transaction with $t_i[A_j] = m'_{i}j$. s is chosen so that the given instance of BCBS is a "YES" instance iff M' contains a $K \times K$ submatrix that is all 1's iff the target instance of SRP is a "YES" instance. The argument below establishes formally this correspondence.

First we show that the given instance of BCBS is a "YES" instance implies that the target instance of SRP is a "YES" instance. Let V_1 and V_2 be the required size K subsets of vertices. Construct the projection set A by placing A_j in A iff $v_j \in V_1$. Claim that for each $A_j \in A$, $t_i[A_j] = 1$ whenever $v_i \in V_2$. To see this, observe that $\{v_i, v_j\}$ is an edge in E (since V_1 and V_2 are as required by the instance of BCBS) and recall the construction of the transactions. Therefore, these K transactions are not singletons under the projection onto A, and thus at most $s = D - K$ transactions can be singletons.

We now show that the target instance of SRP is a "YES" instance implies the given instance of BCBS is a "YES" instance. Let A be the required set of attributes and, without loss of generality, assume that A contains exactly K attributes. (If A contains more than K attributes, any size K subset of A has the required property.) Observe that, by the construction of DB in the target instance, t_i not a singleton implies that $t_i[A_j] = 1$ for every $A_j \in A$. Let T be any collection of K transactions that are not singletons under the projection onto A, i.e., T is any size K collection of t such that $t[A_j] = 1$ for $A_j \in A$. Such a collection is guaranteed to exist, since A satisfies the conditions for the SRP instance and $K = D - s$, by the instance's construction. Observe that, by the construction of DB, if $A_j \in A$, it cannot be the case that $t_j \in T$ (recall $t_j[A_j] \neq 1$). Construct the subsets V_1 and V_2 of V by placing v_j in V_1 whenever $A_j \in A$ and placing v_i in V_2 whenever $t_i \in T$. The previous observation ensures that V_1 and V_2 are disjoint. Finally, observe that $v_j \in V_1$ and $v_i \in V_2$ implies $v_i, v_j \in E$, since $t_i[A_j] = 1$. □

Note that the NP-hardness of the optimization version of the above decision problem (i.e., construct a cardinality K subset of attributes which induces the smallest possible number of singleton transactions) follows immediately from Theorem 3. Hence, Theorem 3 and its proof imply that no algorithm that solves exactly the singleton reduction problem can run in time bounded by a polynomial in the number L of attributes, unless $P = NP$. While a simple exhaustive search of attribute subsets is feasible for small values of L (e.g., $L < 20$), audit trails which we have encountered in practice often define transaction templates over a large number (e.g., often, L is near 100) of attributes. On the more positive side, we have had some experimental success with greedy and branch-and-bound heuristics that find good, though not necessarily optimal, solutions to the singleton reduction problem.

IV. Nonmodeling Approaches

The approaches to misuse detection that we discussed in the previous sections all required estimation of the likelihood ratio r and, therefore, estimation of n and m. We call these *modeling approaches*. The approaches have distinct advantages and disadvantage relative to other approaches. On the plus side, they attack directly the misuse detection problem in terms of the very parameters in which it is formulated, that is, in terms of explicit N and M models and the minimization of well-quantified error measures. On the minus side, modeling approaches undoubtedly require reduction of the original data through the formation of equivalence classes and are likely to be sensitive to the operative misuse models or the surrogates employed in an attempt to differentiate misuse behavior from normal behavior. Consequently, each implementation for a specific application can be expected to require specialized design.

As an alternative, in this section we discuss *nonmodeling* approaches. Nonmodeling approaches do not explicitly estimate n or m. Instead, they use various heuristics, clustering algorithms, and statistical measures to flag "bad" transactions. Simonian *et al.* [21] based their approach on Kohonen self-organizing nets. Clithrow [2] assigned users to projects and used backpropogation networks to test for user adherence to historical usage patterns. Lunt *et al.*'s. original formulation [15] used the Mahalanobis distance to detect outliers. Vaccaro and Liepins [23] generates a heuristic rule forest that specifies acceptable values in one attribute, conditioned on the values in other attributes.

Proponents of the nonmodeling approach hope that these systems will be broadly applicable to a variety of installations without the detailed analysis and design required by modeling approaches. They hope to have gained a degree of robustness by a less direct attack on the misuse problem. Nonetheless, they cannot avoid totally all the analysis associated with modeling approaches; nonmodeling approaches too require attribute projection and value aggregation, and their performance with respect to standard measures of error must be analyzed. We contend that the effectiveness of any detection system—modeling or nonmodeling—ultimately is tied to the accuracy with which the system approximates the optimal detector based on the likelihood ratio r.

While systems which adopt a nonmodeling approach are a diverse lot, the technique of statistical rule generation is central to many such systems. Section IV-A describes statistical rule generation and Section IV-B then analyzes the consistency of such rule-based systems with respect to the optimal detectors that were studied in Sections II and III. Finally, Section V provides limited test results comparing the performance of a pair of systems, one based on a modeling approach and the other on a nonmodeling approach.

A. Statically Generated Rule Bases

Systems that employ statistically generated rule bases process the historical data and generate *rules* which specify relationships between the values of groups of attributes. For example, a rule might state that if user is Smith, then port is one of tty1, tty2, or tty3. Such a rule can be represented by the implication

(user=Smith) → (port=tty1) or (port=tty2) or (port=tty3).

A rule's antecedent is called its left hand side (LHS) and its conclusion is called its right hand side (RHS). In the above example, the LHS is (user=Smith) and the RHS is (port=tty1) or (port=tty2) or (port=tty3).

We have the following definitions.

Definition 4: Let $A_1, \cdots, A_k$ be the the set of attributes over which transaction templates are defined. A *rule* is an implication

$$R : LHS => RHS$$

where each of LHS and RHS is a Boolean formula over the set $A_1, \cdots, A_k$ of attributes. A transaction $x =< A_1 = v_1, \cdots, A_k = v_k >$ *matches* rule R if each attribute value $A_i = v_i$ is consistent with LHS. If x matches R, it also either *passes* or *fails* R; x passes R if each attribute value $A_i = v_i$ is consistent with RHS, and otherwise x fails R.

Example 4.1: Suppose transactions are defined over the set {user, port, command, time_of_day} of attributes, and consider the rule

R : (user=Smith) and (port=tty4) => (command=edit)

and the transactions

$$x =< user = Smith, porty = tty2, command = compile,$$
$$time\_of\_day = 06:00 >$$
$$y =< user = Smith, porty = tty4, command = compile,$$
$$time\_of\_day = 06:00 >$$
$$z =< user = Smith, porty = tty4, command = edit,$$
$$time\_of\_day = 06:00 > .$$

Transaction x is inconsistent with the LHS (user=Smith) and (port=tty4); hence x does not match rule R. Transaction y is consistent with the LHS (user=Smith) and (port=tty4), but is inconsistent with the RHS (command=edit); hence y matches and fails rule R. Transaction z is consistent with the LHS (user=Smith) and (port=tty4), and is consistent also with the RHS (command=edit); hence z matches and passes rule R.

Notice that since attribute time_of_day does not appear in rule R, a transaction's value on this attribute has no effect on whether it matches, passes, or fails R. □

A statistical rule-based system uses the database DB to associate with each rule it generates measures of the rule's historical significance. Systems with which we are familiar measure significance in terms of (1) the rule's pass/match ratio, that is, the ratio

$$\frac{(\text{number } x \in DB \text{ which pass the rule})}{(\text{number } x \in DB \text{ which match the rule})},$$

and (2) the number $x \in DB$ that match the rule. A rule's pass/match ratio can be interpreted as specifying a empirically observed estimate of the conditional probability $\Pr\{RHS|LHS\}$, while the number matched specifies the sample size used in the approximation of this conditional probability.

When transaction x is presented for analysis, the detection system determines which rules x matches and, of these, which rules x passes and which it fails. The detection system then employs a *scoring function* that interprets the evidence yielded by the rules matched by a transaction to produce a score characterizing the overall suspicion that is to be attributed to the transaction. While many specific scoring functions are possible, the classes defined below appear to be both natural and general.

A scoring function sf evaluates transaction x against a rule base and historical database DB by constructing a vector of information. This vector contains a triple for each of the M rules in the rule base. Component j of the vector specifies whether x matches rule R_j, and if so whether x passes or fails the rule, and summarizes R_j's historical significance as measured in terms of r_j, its pass/match ratio in DB, and m_j, the number of transactions in DB which match it. More formally, we have the following.

Definition 5: A *scoring function* sf over a rule base of M rules is a function

$sf :< \{-1, 0, 1\}x[0, 1]xN >^M \rightarrow R.$

sf's evaluation of transaction x against the rule base and database DB of transactions is determined by a vector of M triples (I_j, r_j, m_j) $(j = 1, 2, \cdots, M)$ derived for x from the rule base and DB, where

$$I_j = \begin{cases} -1 & \text{if } x \text{ fails the rule } R_j \\ 0 & \text{if } x \text{ does not match the rule } R_j \\ 1 & \text{if } x \text{ passes the rule } R_j \end{cases}.$$

r_j= the pass/match ratio in DB of R_j, where 0/0 is defined to be 0.

m_j= the number of matches in DB of R_j

The class of *well-behaved* scoring functions consists of those continuous scoring functions satisfying:

$$I_j * \frac{\partial(sf)}{\partial r_j} < 0, \qquad \text{for } I_j \in \{-1, 1\}$$
$$\text{and } r_j \in (0, 1), j = 1, 2, \cdots, M.$$

The condition on the jth partial derivative characterizes the manner in which the scoring function interprets changes in the historical compliance of transactions with R_j, while other historical information is held constant. In particular, the condition on the jth partial derivative implies:

1) If transaction x fails rule R_j (and hence $I_j = -1$), scoring function sf's evaluation of x is required to be *monotonically increasing* in suspicion as the number of historical transactions which pass R_j increases, while all other historical information is held constant. In terms of the condition on the partial derivatives, if x fails rule R_j, I_j is -1, hence requiring $\partial(sf)/\partial r_j > 0$.

2) If transaction x passes rule R_j (and hence $I_j = 1$), scoring function sf's evaluation of x is required to be *monotonically decreasing* in suspicion as the number of historical transactions which pass R_j increases, while all other historical information is held constant. In terms of

the condition on the partial derivatives, if x passes rule R_j, I_j is 1, hence requiring $\partial(sf)/\partial r_j < 0$.

Example 4.2: A simple, well-behaved scoring function evaluates transaction x by adding the historical pass/match ratios for rules failed by x and subtracting the pass/match ratios for rules passed by x. It is immediately evident that the condition on the partial derivatives is satisfied. □

B. A Consistency Result for Rule-Based Systems

In this section, we specify necessary conditions that nonmodeling approaches must satisfy if they are to minimize prioritization error. Our consistency results apply to all techniques that utilize rules as described in Section IV-A.

Before deriving our results, we first must state additional technical definitions and prove some technical propositions.

Definition 6: Let $F[\,]$ be a functional that relates normal processes N to misuse processes M by the relation

$$m(\cdot) = F[n(\cdot)]$$

$F[\,]$ is *piecewise monotonic* if, for any pair N_1 and N_2 of normal processes with associated densities n_1 and n_2, whenever transaction $x \in S$ is such that

$$n_1(x[A]) \leq n_2(x[A]) \textit{for all subsets } A \text{of attributes}$$

it must follow that

$$m_1(x) \leq m_2(x)$$

where m_i is obtained from n_i as $m(\cdot) = F[n(\cdot)]$, $i = 1, 2$.

It is easily seen that the independence and uniform models are piecewise monotonic. Further, we have the following.

Lemma 1: Any monotonic composition of piecewise monotonic functionals is a piecewise monotonic functional. That is, suppose for $1 \leq i \leq k$ $m^i = F^i[n(\cdot)]$ for piecewise monotonic F^i. If functional F generates the density

$$m(x) = g(m^1(x), \cdots, m^k(x))$$

where g is nondecreasing in each of its arguments, F is piecewise monotonic.

Proof: Let n_1 and n_2 be any pair of densities. For $i = 1, 2$ let $m_i(\cdot)$ be generated from $n_i(\cdot)$ by

$$m_i(x) = g(m_i^1(x), \cdots, m_i^k(x))$$

where g is monotonic and each $m_i^j(\cdot)$ is generated from $n_i(\cdot)$ by a piecewise monotonic $F^j[\,]$. Suppose transaction x is such that

$n_1(x[A]) \leq n_2(x[A])$ for all subsets A of attributes.

Then, since each $m_i^j(x)$ is produced by application to $n_i(\cdot)$ of a piecewise monotonic $F^j[\,]$,

$m_1^j(x) \leq m_2^j(x)$, for $j = 1, 2, \cdots, k$.

It then follows from the monotonicity of g that

$$m_1(x) \leq m_2(x)$$ □

Definition 7: R is a *conjunctive rule* if it can be represented as

$$R : (a_1 \in V_1) \ldots (a_{i-1} \in V_{i-1}) => (a_i \in V_i)$$

where each V_j is a set of values.

Definition 8: Conjunctive rule R is *nonmaximal* if there exist one or more attributes over which transaction templates are defined which appear in neither the LHS nor RHS of R.

The main result of this section is to establish that when a scoring function is well behaved and a piecewise monotonic $F[]$ relates N and M, nonmaximal rules can lead to transaction scoring that is inconsistent with the optimal detectors defined in Sections II and III. Before presenting this result, we require two additional definitions and a lemma.

Definition 9: Let transactions consist of L attributes. Let x be an arbitrary but fixed transaction, let $1 \leq i \neq j \leq L$ be some fixed pair of attributes, and let v be a value from the domain of attribute i such that $x[i] \neq v$.

1) y is a *candidate* for a (x, i, j, v) transform if $y[i] = x[i]$ and $y[j] \neq x[j]$.
2) If y is a candidate for a (x, i, j, v) transform, y's (unique) *target* under this transform is the transaction z identical to y on all attributes, except that $z[i] = v$.

Note that a (x, i, j, v) transform defines a 1–1 correspondence between its candidate and target transactions. Hence, we may speak of the target [candidate] transaction that is related by a (x, i, j, v) transform to a given candidate [target] transaction. When the (x, i, j, v) transform is understood, such a related pair of transaction will be denoted as an ordered pair (y, z), where y is the candidate and z is its target.

Definition 10: Let x, i, j, and v be as in the previous definition. Stochastic process N_2 is a (x, i, j, v) *perturbation* of stochastic process N_1 if:

1) For each pair (y, z) of candidate and target transactions related by an (x, i, j, v) transform, there exists $\delta \geq 0$ such that $n_2(y) = n_1(y) - \delta$ and $n_2(z) = n_1(z) + \delta$.
2) $n_2(w) = n_1(w)$ if w is neither a candidate nor a target of a (x, i, j, v) transform.

Intuitively, a (x, i, j, v) perturbation increases the frequency of some transactions that differ from x on both the attributes i and j at the expense of the frequency of some transactions that differ from x on j but agree with x on i.

Example 4.3: Let the attribute set be as in Example 4.1, let

$$x = <user = Smith, porty = tty2, command = compile,$$
$$time\_of\_day = 06:00>,$$

and consider $(x, port, command, tty4)$ transforms. Transaction

$$y = <user = Jones, porty = tty2, command = edit,$$
$$time\_of\_day = 06:00>$$

is a candidate for a $(x, port, command, tty4)$ transform, and transaction

$$z = <user = Jones, porty = tty4, command = edit,$$
$$time\_of\_day = 06:00>$$

is y's unique target under this transform. If N_1 is any stochastic process generating transactions, an example of a N_2 which can be obtained via $(x, port, command, tty4)$ perturbation of N_1 is given by

$$n_2(q) = n_1(q) - 0.01\ n_2(q') = n_1(q') + 0.01,$$

whenever q is a candidate of the the form $<$ $user = Smith, porty = tty2, command = edit, time\_of\_day = * >$ or $< user = Jones, porty = tty2, command = edit, time\_of\_day = * >$, and q' is the corresponding target. For all other u, $n_2(u) = n_1(u)$. □

The following lemma demonstrates that the optimal detector interprets any (x, i, j, v) perturbation as making transaction x no more suspicious (and possibly less suspicious), assuming a piecewise monotonic $F[\,]$ relates the normal and misuse processes.

Lemma 2: Let N_2 be a (x, i, j, v) perturbation of N_1. For $d = 1, 2$, let the densities $n_d(\cdot)$ and $m_d(\cdot) = F[n_d(\cdot)]$ be generated from stochastic process N_d, where $F[\,]$ is piecewise monotonic. Then

$$r_1(x) \geq r_2(x)$$

Proof: First observe that $n_1(x) = n_2(x)$, since x is neither a candidate nor a target of the transform. Let A be any subset of the attributes. We claim $n_1(x[A]) \geq n_2(x[A])$. To see this, suppose $z \in x[A]$ is such that $n_1(z) < n_2(z)$. Then it must be the case that z is a target of a (x, i, j, v) transform and hence that $i, j \notin A$. Consequently, the candidate transaction y which corresponds to z under this transform also is a member of $x[A]$. Since $n_1(\{y, z\}) = n_2(\{y, z\})$ for each such (candidate,target) pair (y, z), we have $n_1(x[A]) \geq n_2(x[A])$. It then follows from the fact that $F[]$ is piecewise monotone that $m_1(x) \geq m_2(x)$ and hence that $r_1(x) \geq r_2(x)$. □

Note that, under the conditions of the previous lemma, if $n_1(w) > n_2(w)$ for some $w \in S$, w must be a candidate for a (x, i, j, v) transform and hence $w[i] = x[i]$. Since no u such that $u[i] = x[i]$ is a target, it follows that $n_1(x[i]) > n_2(x[i])$. Consequently, if functional $F[]$ generates the independence model (see Example 3.1) and if $n_1(w) \neq n_2(w)$ for at least one $w \in S$, then the strict inequality

$$r_1(x) > r_2(x)$$

obtains.

We now use Lemma 2 to exhibit an inconsistency in the scoring of transactions yielded by rule bases containing nonmaximal conjunctive rules. In particular, we demonstrate that if a detection system employs a well-behaved scoring function as defined in Definition 5 then a nonmaximal conjunctive rule will exert influence on the scoring function that is contrary to the result of Lemma 2.

Theorem 4: Let R be any nonmaximal conjunctive rule

$$R : (a_1 \in V_1) \cdots (a_{i-1} \in V_{i-1}) => (a_i \in V_i),$$

and let a_j be any attribute not appearing in R. Assume that a piecewise monotone functional $F[]$ generates the misuse process from the normal process. Then for any normal processes N_1 and any transaction x failing R, there exists a normal process N_2, obtained from N_1 by a (x, i, j, v) perturbation (where v is any member of the set of values V_i), such that $r_1(x) \geq r_2(x)$ yet R has a lower pass/match ratio in any sample DB_1 generated by N_1 and containing at least one transaction that fails R than in any nonempty DB_2 generated by N_2.

Proof: By Lemma 2, the condition on the ratios r_d holds for any (x, i, j, v) perturbation. Consider in particular the (x, i, j, v) perturbation that changes the probabilities of a (candidate, target) pair (y, z) iff y fails R; in this case, the perturbation is by $\delta = n_1(y)$ and hence $n_2(y) = 0$ and $n_2(z) = n_1(z) + n_1(y)$. Consequently, any DB_2 which N_2 generates contains no transactions which fail R. □

Discussion: The transformation described in Theorem 4 increases the pass/match ratio of rule R, while leaving unchanged its expected number of matches. Since x fails R, R exerts on any well-behaved scoring function influence that is contrary to the result of Lemma 2, when a piecewise monotone misuse model is assumed. Further, historical databases DB can be constructed so that the number of transactions matching R is sufficiently large that R's pass/match ratio dominates the transformation's effect on members of many natural subclasses of well-behaved scoring functions [9]. This result, combined with Theorem 4, imply that in the presence of nonmaximal rules, such functions and the optimal detector diverge as to the relative suspicion attributed to x in the context of certain DB_1 and DB_2 pairs obtained as in the proof of the theorem. That is, for such pairs DB_1 and DB_2 we have $sf_1(x) < sf_2(x)$ yet $r_1(x) \geq r_2(x)$.

On the surface, the results appear to suggest that nonmaximal rules not be included in rule bases; however, practice demonstrates that rules which include all attributes are of little value, because reasonably-sized historical databases will contain few transactions matching any such rule (see the discussion in Section III-B on attribute projection). The result should be viewed in terms of "characterizing" attribute sets, as described in Section III-B and Example 3.3. If S is a characterizing attribute set, then $a \notin S$ for our purposes is *irrelevant* and may be (and, as Example 3.3 illustrates, should be) omitted from consideration. (Note that a_j in the process N_2 used in the proof of Theorem 4 is not irrelevant in our sense, since transactions which differ only on this attribute, in general, could be ranked differently depending on whether a_j is included in the projection set.)

What interpretation should be attached to these results, in the context of rule base design, where characterizing attribute sets, at least initially, are not known? We contend that, in the steady state, a rule base should provide scoring that is consistent with ratio tests computed with respect to a consistent collection of hypotheses on the identity of good characterizing sets of attributes. Given this, the above results imply that, when the implicit misuse model is piecewise monotone, a rule base should not contain *nested* rules, i.e., a pair of rules such that the set of attributes referenced by one rule is properly contained in the set of attributes referenced by another rule. If nested rules are present, than either the smaller rule is nonmaximal or the larger rule contains irrelevant attributes.

V. Testing: Wisdom and Sense Versus a Frequentist Detector

In prior sections, we have developed theoretical bounds for detection effectiveness, a consistency condition for rule-generation, nonmodeling approaches, and suggestions on how to group data into equivalence classes when the granularity of the raw data together with limitations of practical sample sizes prevent reliable estimates of the density n. In this section, we suggest the form of a simple, quantitative, and controlled test designed to determine whether a detection system is operating properly. In particular, we describe here a test comparing the performance of Wisdom and Sense and a simple frequentist detector. The parameters of the test are such (e.g., small transaction space, known form of misuse model) that the frequentist detector can be expected to approximate optimal detection. Hence, the test provides a benchmark against which the effectiveness of Wisdom and Sense (or any other detection system) can be measured. Current research is developing more sophisticated testing procedures, and results will be reported in a forthcoming paper.

The design of the test is as follows. We randomly split the historical database in two (not necessarily equal parts) and use one part for training and the other for testing. We next assume a particular misuse model M, and allow the frequentist detector to estimate from the training data and the assumed M the ratio $r(x)$, for any x in the transaction space. The detection system being benchmarked also trains on the training data and M—in the case of Wisdom and Sense, a statistical rule base is generated against the training data. Provided that the training set is sufficiently large relative to the transaction space, the frequentist detector will approximate optimal detection, and hence the quality of the system in question can be assessed.

In tests we performed comparing Wisdom and Sense and the frequentist detector, we assumed the uniform misuse model and considered a two attribute transaction template (user and command), which allows the training set to be large relative to the transaction space. Note that this choice of misuse model implies that transactions rare in the test data should be flagged. Typical frequentist detector results are given in Tables I and II. These results were generated from about 30 000 historical transactions, split into training and testing sets in the ratio 2:1. The sensitivity table is to be interpreted as follows: the 88 in the (2,1) position indicates that 88% of the transactions that appeared less than three times in the test set also appeared less than twice in the training set. Thus, the (2,1) sensitivity was 88% and similarly, the (2,2) sensitivity was 96%. The corresponding interpretation of the false alarm table indicates that for the (2,1) entry, 4% of the transactions that appeared more than twice in the test set also appeared less than twice in the training set. Although not readily apparent from the small set of test results reproduced here, the level curves for (fixed) sensitivity are roughly linear as are the level curves for false alarms. For example, the set of points (x, y) associated with 95% are roughly co-linear; (3,2), (5,3), and (7,4) lie on approximately the same line. A further observation is that the level curves for higher (sensitivity or false alarm) values have somewhat greater slope than those for lower values. Nonetheless, the relationship between sensitivity and false alarms remains relatively constant. A false alarm rate of about 10% corresponds to a sensitivity of about 99%, 5% false alarm to 93–95% sensitivity, and 1% false alarm to 45%–60% sensitivity.

TABLE I
Selection of Sensitivity Results for Naive Detector with 2/3 Training Sample

Train					
5	100	100	100	99	99
4	99	100	100	98	96
3	99	98	98	97	95
2	97	96	95	92	89
1	92	88	84	80	77
0	74	65	59	55	50
	1	2	3	4	5

Test

TABLE II
Selection of False Alarm Results for Naive Detector with 2/3 Training Sample

Train					
5	13	12	11	10	10
4	11	10	9	9	8
3	8	7	7	6	5
2	6	6	5	4	3
1	5	4	3	3	2
0	2	2	2	1	1
	1	2	3	4	5

Test

Tables III and IV present counterpart results for Wisdom and Sense (W&S) expressed in terms of its scoring function, labeled FOM (Figure of Merit); the larger the FOM, the more suspicious the transaction. We note that Wisdom and Sense's FOM is a well-behaved scoring function as defined in Definition 5. As can be seen, the Wisdom and Sense granularity for the two field case is much larger than for the frequentist detector. Also, the interpretation is somewhat different. The (1,0) entry of 84 in Table III means that 84% of the transactions that appeared less than twice in the test set had

TABLE III
SELECTION OF SENSITIVITY RESULTS FOR W&S WITH 1/2 TRAINING SAMPLE

FOM							
7	67	58	57	51	49	44	41
6	83	75	71	62	58	53	50
5	84	76	71	63	59	54	51
4	84	76	71	63	59	54	51
3	84	76	71	63	59	54	51
2	84	76	71	63	59	54	51
1	84	76	71	63	59	54	51
0	84	76	71	63	59	54	51
	1	2	3	4	5	6	7

Test

TABLE IV
SELECTION OF FALSE ALARM RESULTS FOR W&S WITH 1/2 TRAINING SAMPLE

FOM							
7	4	3	2	2	1	1	1
6	5	4	3	2	2	1	1
5	5	4	3	2	2	1	1
4	5	4	3	2	2	1	1
3	5	4	3	2	2	1	1
2	5	4	3	2	2	1	1
1	5	4	3	2	2	1	1
0	5	4	3	2	2	1	1
	1	2	3	4	5	6	7

Test

FOM's greater than 0 (and greater than 5). Similarly, the (1,4) entry of 5 in Table IV means that 5% of the transactions that appeared more than once had FOM's greater than 4. Because of the granularity, one has a choice in how to report the results. For the 1:1 and 2:1 training set-test set splits, a 4–5% false alarm rate corresponds to about 84% sensitivity (in the best case), and a 1% false alarm rate corresponds to 54–66% sensitivity.

Since the frequentist detector in these experiments can be expected to approximate optimal detection, we conclude that for at least the experiments undertaken, Wisdom and Sense is performing well.

VI. CONCLUSION

We have specified misuse detection objectives in terms of a probability model. Given this model we defined performance measures and established theoretical performance bounds in terms of the disparity of the underlying distributions of normal and misuse transactions. We further demonstrated that consistency with likelihood ratio estimation is a prerequisite for minimizing prioritization error, an error that measures the amount of misdirection a SSO can expect from a misuse detector. We further established that sampling error can be a major cause of poor detector performance, and suggested attribute projection and value aggregation as solutions. We presented several criteria for determining good attribute projections and value aggregations and proved that exactly optimizing even the simplest of these criteria is NP-hard. We demonstrated that rule-building, nonmodeling approaches that use nonmaximal conjunctive rules run the risk of being inconsistent with likelihood ratio estimation. Finally, we presented limited test results comparing Wisdom and Sense and the frequentist detector.

Current research is extending this work both theoretically and practically. Theoretically, we wish to relax several of the simplifying modeling assumptions. This includes a refinement of the binary partition of activities so that subcategories of normal and misuse activities can be defined and analyzed. We wish to relax also the assumption that the processes are stationary, thereby supporting the analysis of transaction sequences. Practically, we are developing heuristics for attribute projection and value aggregation, and assessing their effectiveness on simulated detection problems. Additionally, we are studying, both theoretically and experimentally, properties of misuse surrogate models in order to discover how they can best be utilized in detection.

ACKNOWLEDGMENT

This paper is dedicated to the memory of G. Liepins. We wish to thank the referees for many useful suggestions.

REFERENCES

[1] Y. M. M. Bishop, S. E. Fienberg, and P. W. Holland, *Discrete Multivariate Analysis.* Cambridge, MA: M.I.T. Press, 1975.

[2] P. Clitherow and R. Herrara, "A connectionist approach to monitoring computer audit trails," Bellcore, Piscataway, NJ, 1989.

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

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[5] J. H. Friedman, W. Stuetzle, and A. Schroeder, "Projection pursuit density estimation," *JASA,* vol. 79, no. 387, pp. 599–608, 1984.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability.* San Francisco, CA: Freeman, 1979.

[7] D. M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics.* New York: Wiley, 1976.

[8] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika,* vol. 40, parts 3 and 4, pp. 237–264, 1953.

[9] P. Helman, "Rule base design criteria," Technical Report, Los Alamos National Laboratory, Los Alamos, NM, 1990.

[10] H. S. Javitz, and A. Valdes, "The SRI IDES statistical intrusion detector," in *Proc. IEEE Symp. Research in Security and Privacy,* 1990, pp. 316–326.

[11] R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, K. Lee, P. S. Lewis, and S. Qian, "Function approximation and time series prediction with neural networks," LA-UR-90_21, Los Alamos National Laboratory, 1989.

[12] G. E. Liepins and H. S. Vaccaro (). "Anomaly detection: Purpose and framework," in *Proc. 12th Nat. Comput. Security Conf.,* 1989, pp. 495–504.

[13] ——, "Intrusion detection: Its role and validation," *Computers and Security J.,* 1991.

[14] D. O. Loftsgarden and C. P. Quesenberry, "A nonparametric estimate of a multivariate density function," *Ann. Math. Stat.,* vol. 36, pp. 1049–1051, 1965.

[15] T. F. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, P. G. Neuman, H. S. Javitz, and A. Valdes, "IDES: The enhanced prototype," SRI International, SRI-CSL-88-12, 1988.

[16] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.,* vol. 33, pp. 1065–1076, 1962.

[17] T. Poggio and F. Girosi "A theory of networks for approximation and learning," AI Memo no. 1140, C.B.I.P. Paper no. 31, M.I.T., Cambridge, MA, 1989.

[18] S. Qian, Y. C. Lee, R. D. Joncs, C. W. Barnes, and K. Lee, "Function approximation with orthogonal basis net," LALP-90-04, Los Alamos National Laboratory, Los Alamos, NM, 1990.

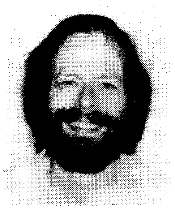
[19] H. E. Robbins, "Estimating the total probability of the unobserved outcomes of an experiment," *Ann. Math. Statist.,* vol. 39, no. 1, pp. 256–257, 1968.

[20] M. M. Sebring, E. W. Shellhouse, M. E. Hann, and R. A. Whitehurst, "Expert systems in intrusion detection," in *Proc. 11th Nat. Comput. Security Conf.,* 1988, pp. 74–81.

[21] R. P. Simonian, P. R. Henning, J. H. Reed, and K. L. Fox, "An AI approach toward computer virus detection and removal," Harris Corporation, Government Information Systems Division, Melbourne, FL, 1989.

[22] W. T. Tenner, "Discovery: An expert system in the commercial data security environment," TRW Information Services Division, Orange, CA, 1988.

[23] H. S. Vaccaro and G. E. Liepins, "Detection of anomalous computer session activity," in *Proc. IEEE Symp. Research in Security and Privacy,* 1989, pp. 280–289.



Paul Helman was born in Brooklyn, NY, in 1954. He received the B.A. degree in mathematics from Dickinson College in 1976, the M.S. degree in operations research from Stanford University in 1977, and the Ph.D. degree in computer science from the Univeristy of Michigan in 1982.

He is currently an Associate Professor of Computer Science at the University of New Mexico. His research interests include computer security, combinatorial optimation, and database theory and applications. In addition to his research contributions, he has authored two computer science textbooks, *Intermediate Problem Solving and Data Structures: Walls and Mirrors*, and *The Science of Database Management.*

Gunar Liepins received the Ph.D. degree in mathematics from Dartmouth College in 1974 and the M.S. degree in engineering/economic systems from Stanford University in 1977. At the time of his death in 1992, Dr. Liepins was a Research Staff Member at Oak Ridge National Laboratory and an Adjunct Associate Professor of Computer Science at the University of Tennessee. His research interests included statistics, operations research, genetic algorithms, and computer security.