A Decentralized Bayesian Attack Detection Algorithm for Network Security

Kien C. Nguyen, Tansu Alpcan, and Tamer Başar

Abstract Decentralized detection has been an active area of research since the late 1970s. Its earlier application area has been distributed radar systems, and more recently it has found applications in sensor networks and intrusion detection. The most popular decentralized detection network structure is the parallel configuration, where a number of sensors are directly connected to a fusion center. The sensors receive measurements related to an event and then send summaries of their observations to the fusion center. Previous work has focused on separate optimization of the quantization rules at the sensors and the fusion rule at the fusion center or on asymptotic results when the number of sensors is very large and the observations are conditionally independent and identically distributed given each hypothesis.

In this work, we examine the application of decentralized detection to intrusion detection with again the parallel configuration, but with joint optimization. Particularly, using the Bayesian approach, we seek a joint optimization of the quantization rules at the sensors and the fusion rule at the fusion center. The observations of the sensors are not assumed to be conditionally independent nor identically distributed. We consider the discrete case where the distributions of the observations are given as probability mass functions. We propose a search algorithm for the optimal solution. Simulations carried out using the KDD'99 intrusion detection dataset show that the algorithm performs well.

Kien C. Nguyen
Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W Main St., Urbana, IL 61801, USA,
e-mail: knguyen4@uiuc.edu

Tansu Alpcan
Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, D-10587 Berlin, Germany,
e-mail: tansu.alpcan@telekom.de

Tamer Başar
Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W Main St., Urbana, IL 61801, USA,
e-mail: tbasar@control.csl.uiuc.edu

1 Introduction

There is pressing need for extensive research and development of novel approaches to address security problems in networked systems. The current cost of security-related issues is on the order of billions of dollars in terms of lost productivity, prevention, and clean-up. This affects individuals, businesses, and organizations on a global scale. For an example, the Code Red worm, which infected some 360,000 servers, cost about $1.2 billion in damage to computer networks [1]. As a result of the general-purpose nature of current computing systems and due to their social underpinnings, network security poses significant challenges that require innovative security architectures.

The problem of decentralized detection has been addressed in many works ([2], [3], [4], [5], [6], and [7]). The concepts and taxonomy of intrusion detection systems can be found in [8] and [9]. Reference [10] provides a survey on intrusion detection for mobile *ad hoc* networks. Furthermore, the authors in [11] have proposed an algorithm for decentralized intrusion detection in the context of wireless sensor networks. The use of Principal Component Analysis (PCA) to detect network anomalies has been examined in [12], [13] and [14], while reference [15] uses a Markov chain model to learn the normal behavior and then detect the anomalies. Also, application of game theory to intrusion detection has been examined in [16] and [17].

A variety of network security issues such as attack and anomaly detection can be addressed within the framework of Bayesian hypothesis testing. In such a framework, one considers networked security systems with multiple virtual sensors (detection units) implemented as software agents that report various measurements or observations. In many cases, sending all this information to a centralized location for processing (attack detection) has several disadvantages such as traffic overhead and need for extensive computing resources at the center. To remedy these issues, we resort in this paper to decentralized hypothesis testing for attack detection.

KDD[1] Cup 1999 [18] is a dataset extracted from the TCP dump data of a Local Area Network (LAN). The LAN was set up to simulate a United States Air Force LAN and speckled with different kinds of attacks. From this dataset, it can be shown that the observations from different sensors (parameters) are not necessarily identically distributed and may also be strongly correlated. Thus the analyses and results developed under the assumption of conditionally independent and identically distributed (*i.i.d.*) observations with a large number of sensors will not be applicable here. We therefore attempt to analyze a sensor network with a finite number of sensors. We do not assume that the observations are conditionally *i.i.d.* We use the Bayesian criterion, i.e., the cost function is the average probability of error at the fusion center.

The main contributions of this paper are: (i) applying decentralized hypothesis testing to intrusion detection, where each sensor observes a parameter of the system or current connection; (ii) proposing a search algorithm for the optimal (Bayesian)

[1] KDD stands for Knowledge Discovery and Data Mining [18].

thresholds for the general case of non-*i.i.d.* observations, provided that the sensors are restricted to use likelihood ratio tests; and (iii) deriving some relationships between the majority vote and the likelihood ratio test for a parallel configuration.

The rest of the paper is organized as follows. The background theory is presented in Section 2. In Section 3, we derive some relationships between the majority vote and the likelihood ratio test at the fusion center. We then propose a search algorithm to find the optimal thresholds for the sensors in Section 4. Section 5 gives a brief overview of the KDD 1999 dataset, discusses the application of hypothesis testing in attack and anomaly detection, and presents the simulation results using the dataset. Finally, some concluding remarks end the paper.

2 Decentralized hypothesis testing with non-*i.i.d.* observations

In this section, we formulate the problem of decentralized hypothesis testing with non-*i.i.d* observations. We first discuss centralized detection before proceeding with the decentralized problem. Extensive discussion on both models can be found in [4]. In Subsection 2.2, we provide details on the fusion rule and the average probability of error at the fusion center.

2.1 From centralized to decentralized detection

Centralized detection. First we consider the configuration given in Figure 1. This is a parallel configuration with a finite number of sensors and a data fusion center. The sensors observe two hypotheses, H_0 and H_1, corresponding, for example, to the normal state and an attack, respectively. Let $Y_1, Y_2, \ldots, Y_N$, the observations of the sensors, be N discrete random variables that take values in finite sets $\mathscr{Y}_1, \mathscr{Y}_2, \ldots, \mathscr{Y}_N$, respectively. The observations are not assumed to be conditionally independent nor identically distributed. In this model, we suppose that the fusion center has full access to the observations of the sensors. It then fuses all the data to finally decide whether H_0 or H_1 is true. From the result of centralized Bayesian hypothesis testing [19], the rules can be stated as follows:

$$\gamma_0(y_1, y_2, \ldots y_N) = \begin{cases} 1 & \text{if } \frac{P_1(y_1, y_2, \ldots y_N)}{P_0(y_1, y_2, \ldots y_N)} \geq \frac{\pi_0}{\pi_1} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $P_1(y_1, y_2, \ldots y_N)$ denotes the joint probability of the Y_i's under hypothesis H_1, i.e., $P(Y_1 = y_1, Y_2 = y_2, \ldots Y_N = y_N | H_1)$; $P_0(y_1, y_2, \ldots y_N)$ denotes the joint probability of the Y_i's under hypothesis H_0, i.e., $P(Y_1 = y_1, Y_2 = y_2, \ldots Y_N = y_N | H_0)$; π_0 and π_1 are the prior probabilities of H_0 and H_1, respectively; and γ_0 is the fusion rule at the fusion center. Throughout this paper, we use the indices of the hypotheses (0, 1) to indicate the hypotheses (H_0, H_1) in the equations. Note that the fusion rule

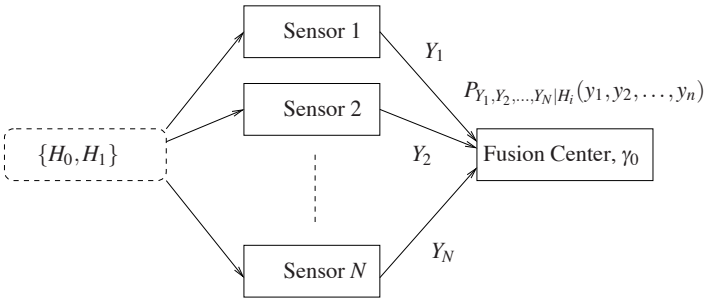

Fig. 1 Centralized detection, where the fusion center has full access to the observations of the sensors.

involves a threshold which is the ratio of π_0 to π_1, and the likelihood ratio (ratio of probabilities under the two hypotheses) is tested against that threshold.

Decentralized detection. In the decentralized detection model, instead of providing the full observation, each sensor only transmits 1 bit of information (which is a local decision whether H_0 or H_1 is true) to the fusion center, which will fuse all the bits to finally decide between H_0 or H_1. The communication channels between the sensors and the fusion center are assumed to be perfect. We seek a joint optimization of the quantization rules of all the sensors $(\gamma_1(.), \ldots, \gamma_N(.))$ and the fusion rule of the fusion center $(\gamma_0(.))$ to minimize the average probability of error of the system. The configuration of N sensors and the fusion center are shown in Figure 2.

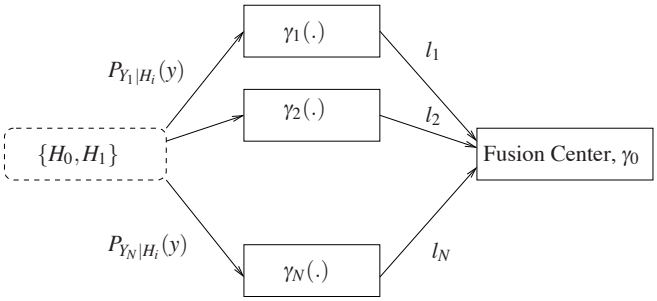

Fig. 2 Decentralized detection model, where each sensor transmits 1 bit of information to the fusion center, which will fuse all the bits to finally decide whether H_0 or H_1 is true.

Naturally, given the same *a priori* probabilities of the hypotheses and conditional joint distributions of the observations, the decentralized configuration will yield an average probability of error that is higher than or equal to that of the centralized configuration. The reason is that we lose some information after the quantization at the sensors [4]. Putting it another way, given the observations of the sensors and assuming the use of a likelihood ratio test at the fusion center in the centralized

configuration, the test in (1) will yield the minimum probability of error. The decentralized configuration, however, can always be considered as a special setup of the fusion center in the centralized case, where the observations from the sensors are quantized before being fused together.

Under the assumption that the observations are conditionally independent, it has been shown in [4] that there exists an optimal solution for the local sensors, which is a deterministic (likelihood ratio) threshold strategy. When the observations are conditionally dependent, however, the threshold rule is no longer necessarily optimal [4]. In this case, obtaining the overall optimal non-threshold rule is a very challenging problem. In view of this, we restrict ourselves to threshold-type rules (which are suboptimal) at the local sensors and seek optimality within that restricted class. The optimal fusion rule, as shown next, will also be a likelihood ratio test.

2.2 The fusion rule and the average probability of error

For each combination of the thresholds at the sensors $\{\tau_1, \tau_2, \ldots, \tau_N\}$, the fusion rule ($\gamma_0$) is determined based on the likelihood ratio test at the fusion center:

$$\gamma_0(l_1, l_2, \ldots, l_N) = \begin{cases} 1 & \text{if } \frac{P_1(l_1,l_2,\ldots,l_N)}{P_0(l_1,l_2,\ldots,l_N)} \geq \frac{\pi_0}{\pi_1} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Here $P_i(l_1, l_2, \ldots, l_N)$ is the conditional joint probability mass function (*pmf*) given $H_i, i = 0, 1$.

This result can be derived from the solution of the one-sensor Bayesian detection problem [19], where the fusion center is considered as a sensor with the local decisions (from the connected sensors) as its observations [4].

The average probability of error at the fusion center is then given by:

$$\begin{aligned} P_e &= \pi_0 P_0\left(\frac{P_1(l_1, l_2, \ldots, l_N)}{P_0(l_1, l_2, \ldots, l_N)} \geq \frac{\pi_0}{\pi_1}\right) + \pi_1 P_1\left(\frac{P_1(l_1, l_2, \ldots, l_N)}{P_0(l_1, l_2, \ldots, l_N)} < \frac{\pi_0}{\pi_1}\right) \\ &= \pi_0 \sum_{l_1, l_2, \ldots, l_N : L_a \geq \frac{\pi_0}{\pi_1}} P_0(l_1, l_2, \ldots, l_N) + \pi_1 \sum_{l_1, l_2, \ldots, l_N : L_a < \frac{\pi_0}{\pi_1}} P_1(l_1, l_2, \ldots, l_N) \end{aligned}$$

$$\text{where } L_a = \frac{P_1(l_1, l_2, \ldots, l_N)}{P_0(l_1, l_2, \ldots, l_N)}. \tag{3}$$

As we are considering the discrete case, where the conditional joint distributions are given as *pmf*s, the conditional joint distributions of the local decisions can be written as:

$$P_i(l_1, l_2, \ldots, l_N) = \sum_{Y_N \in R_{Ni_N}} \cdots \sum_{Y_1 \in R_{1i_1}} P_i(Y_1, Y_2, \ldots, Y_N) \tag{4}$$

where $i_n = 0, 1$, and R_{ni_n} is the region where Sensor n decides to send bit i_n, $n = 1, \ldots, N$:

$$R_{n1} = \left\{ Y_n \in \mathscr{Y}_n : L_{Y_n} = \frac{P_1(Y_n)}{P_0(Y_n)} \geq \tau_n \right\} \tag{5}$$

$$R_{n0} = \left\{ Y_n \in \mathscr{Y}_n : L_{Y_n} = \frac{P_1(Y_n)}{P_0(Y_n)} < \tau_n \right\}. \tag{6}$$

where $L_{Y_n} = P_1(Y_n)/P_0(Y_n)$ is the likelihood ratio at Sensor n.

Our goal is to find the combination $\{\tau_1, \tau_2, \ldots, \tau_N\}$ that yields the minimum probability of error at the fusion center. If the number of threshold candidates for every sensor is finite, the number of combinations of thresholds will also be finite. Then there is an optimal solution, i.e., a combination of thresholds $\{\tau_1, \tau_2, \ldots, \tau_N\}$ that yields the minimum probability of error. In Section 4, we show how to pick the threshold candidates for each sensor.

3 The majority vote versus the likelihood ratio test

In this section, we first show that if the observations of the sensors are conditionally independent, given the set of thresholds at the local sensors, any sensor switching from decision 0 to decision 1 will increase the likelihood ratio at the fusion center. Furthermore, if the observations are conditionally *i.i.d.* and the sensors all use the same threshold for the likelihood ratio test, the likelihood ratio test at the fusion center becomes equivalent to a majority vote. In the general case, where the observations are not *i.i.d.*, this property no longer holds; we provide towards the end of the section an example where the likelihood ratio test and the majority vote yield different results.

Recall that the fusion rule at the fusion center is given by (2). If the observations of the sensors are conditionally independent, the likelihood ratio at the fusion center becomes:

$$\frac{P_1(l_1, l_2, \ldots, l_N)}{P_0(l_1, l_2, \ldots, l_N)} = \frac{\prod_{n=1}^{N} P_1(l_n)}{\prod_{n=1}^{N} P_0(l_n)} = \prod_{n=1}^{N} \frac{P_1(l_n)}{P_0(l_n)}.$$

Let us denote by $\mathscr{N}$ the set of all local sensors (represented by their indices). We divide $\mathscr{N}$ into two partitions: $\mathscr{N}_0$, the set of local sensors that send 0 to the fusion center, and $\mathscr{N}_1$, the set of local sensors that send 1 to the fusion center. Then we have $\mathscr{N}_0 \bigcup \mathscr{N}_1 = \mathscr{N}$ and $\mathscr{N}_0 \bigcap \mathscr{N}_1 = \emptyset$. Note that, given the conditional joint probabilities of the observations, $\mathscr{N}_0$ and $\mathscr{N}_1$ are set-valued functions of the thresholds $\{\tau_1, \tau_2, \ldots, \tau_N\}$. Let N_0 and N_1 denote the cardinalities of $\mathscr{N}_0$ and $\mathscr{N}_1$, respectively. Obviously, $N_0, N_1 \in \mathscr{Z}$ (where $\mathscr{Z}$ is the set of all integers), $0 \leq N_0, N_1 \leq N$, and $N_0 + N_1 = N$. Now the likelihood ratio can be written as:

$$\frac{P_1(l_1, l_2, \ldots, l_N)}{P_0(l_1, l_2, \ldots, l_N)} = \prod_{n \in \mathscr{N}_0} \frac{P_1(l_n = 0)}{P_0(l_n = 0)} \prod_{m \in \mathscr{N}_1} \frac{P_1(l_m = 1)}{P_0(l_m = 1)}. \tag{7}$$

From the definitions of the decision regions in (5), (6) we have that

$$P_1(l_n=1)=\sum_{Y_n:L_{Y_n}\geq\tau_n}P_1(Y_n)\ \text{ and }\ P_0(l_n=1)=\sum_{Y_n:L_{Y_n}\geq\tau_n}P_0(Y_n).$$

Consider the region where Sensor n decides 1 (defined in (5)), $\{R_{n1}:Y_n\in\mathscr{Y}_n:L_{Y_n}=P_1(Y_n)/P_0(Y_n)\geq\tau_n\}$. We have that

$$P_1(l_n=1)=\sum_{Y_n:L_{Y_n}\geq\tau_n}P_1(Y_n)\geq\tau_n\sum_{Y_n:L_{Y_n}\geq\tau_n}P_0(Y_n)\geq\tau_nP_0(l_n=1),$$

or

$$\frac{P_1(l_n=1)}{P_0(l_n=1)}\geq\tau_n. \tag{8}$$

Similarly, summing over the region where Sensor n decides 0 (defined in (6)), $\{R_{n0}:Y_n\in\mathscr{Y}_n:L_{Y_n}=P_1(Y_n)/P_0(Y_n)<\tau_n\}$, we have that

$$\frac{P_1(l_n=0)}{P_0(l_n=0)}<\tau_n. \tag{9}$$

From (7), (8) and (9), we can see that any sensor switching from decision 0 to decision 1 will increase the likelihood ratio at the fusion center.

Now, if the observations are conditionally *i.i.d.* and all the sensors use the same threshold then

$$P_i(l_n=1)=\sum_{Y_n:L_{Y_n}\geq\tau}P_i(Y_n)=P_i(l_m=1)$$

where $i=0,1;\ 0\leq m,n\leq N$. Thus we can write (7) as follows:

$$\frac{P_1(l_1,l_2,\ldots,l_N)}{P_0(l_1,l_2,\ldots,l_N)}=\left(\frac{P_1(l=0)}{P_0(l=0)}\right)^{N-N_1}\left(\frac{P_1(l=1)}{P_0(l=1)}\right)^{N_1}. \tag{10}$$

The fusion rule compares the likelihood ratio in (10) with the ratio π_0/π_1. Again, using (8) and (9), it can be seen that the likelihood ratio is a non-decreasing function of N_1. Therefore the likelihood ratio test becomes equivalent to a majority vote rule in this case.

In what follows, we give an example where $L(001)>L(110)$ for the case of three sensors. The observations are supposed to be conditionally independent but not conditionally identically distributed. If we use the majority vote, the fusion center will output H_1 if it receives $(1,1,0)$ and H_0 if it receives $(0,0,1)$. On the contrary, we will show that, if the likelihood ratio test is used, the fusion center will pick $(0,0,1)$ against $(1,1,0)$ for H_1. Using the independence assumption, we have that:

$$L(110)=\frac{P_1(110)}{P_0(110)}=\frac{P_1(l_1=1)}{P_0(l_1=1)}\frac{P_1(l_2=1)}{P_0(l_2=1)}\frac{P_1(l_3=0)}{P_0(l_3=0)},$$

$$L(001) = \frac{P_1(001)}{P_0(001)} = \frac{P_1(l_1=0)}{P_0(l_1=0)} \frac{P_1(l_2=0)}{P_0(l_2=0)} \frac{P_1(l_3=1)}{P_0(l_3=1)}.$$

Consider the ratio

$$\begin{aligned} \frac{L(001)}{L(110)} &= \frac{P_1(l_1=0)P_0(l_1=1)}{P_1(l_1=1)P_0(l_1=0)} \frac{P_1(l_2=0)P_0(l_2=1)}{P_1(l_2=1)P_0(l_2=0)} \frac{P_1(l_3=1)P_0(l_3=0)}{P_1(l_3=0)P_0(l_3=1)} \\ &= \frac{[1-P_1(l_1=1)][1-P_0(l_1=0)]}{P_1(l_1=1)P_0(l_1=0)} \frac{[1-P_1(l_2=1)][1-P_0(l_2=0)]}{P_1(l_2=1)P_0(l_2=0)} \\ &\quad \frac{P_1(l_3=1)P_0(l_3=0)}{[1-P_1(l_3=1)][1-P_0(l_3=0)]}. \end{aligned} \tag{11}$$

As l_1, l_2, and l_3 are conditionally independent given each hypothesis, we can choose their conditional probabilities such that the ratio in (11) is larger than 1. For example, we can choose the conditional probabilities as follows:

$$\begin{aligned} &P_1(l_1=1) = P_0(l_1=0) = P_1(l_2=1) = P_0(l_2=0) = 0.6, \\ &P_1(l_3=1) = P_0(l_3=0) = 0.9. \end{aligned}$$

Such conditional probabilities can be obtained if we choose P_0 and P_1 as in Figure 3 with $k=2.5$ for Sensor 1 and Sensor 2, and $k=10$ for Sensor 3; and the thresholds for all three quantizers satisfy $1/(k-1) < \tau < k-1$.

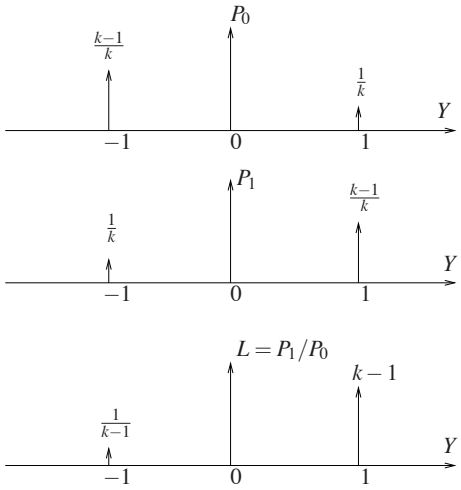

Fig. 3 The majority vote versus the likelihood ratio test: If P_0 and P_1 of each sensor is as shown, the thresholds for all three quantizers satisfy $1/(k-1) < \tau < k-1$ with $k=2.5$ for Sensor 1 and Sensor 2 and $k=10$ for Sensor 3, then $L(001) > L(110)$. A majority vote will output H_1 if it receives $(1,1,0)$ and H_0 if it receives $(0,0,1)$, while the likelihood ratio test favors $(0,0,1)$ for H_1.

4 An algorithm to compute the optimal thresholds

As mentioned in the introduction, the binary decentralized detection problem with two sensors, binary messages, and the fusion rule fixed *a priori* is NP-complete [20]. We thus propose in this section a brute-force search algorithm to solve the optimization problem. (For a discussion on the complexity of this kind of algorithms, see [4], [20].) This algorithm is suitable for small sensor networks. Suppose that we are given a training dataset each record of which has been labeled with either "Normal" or "Attack". Suppose further that each record consists of N parameters, each of which takes values in a finite set. We do not assume that the observations of the sensors (the parameters) are conditionally independent nor identically distributed. The *a priori* probabilities and the conditional joint *pmf*s given each hypothesis then can be learnt from the training dataset. The search algorithm for the optimal thresholds is as follows.

The algorithm to compute the optimal thresholds at the sensors:

1. Group all possible values of each parameter into equally spaced bins with the number of bins for the n-th parameter denoted by b_n. In general, b_n's do not have to be equal. This operation is done for both "Normal" and "Attack" modes.
2. Compute the *a priori* probabilities of "Normal" and "Attack", π_0 and π_1.
3. Compute the conditional joint *pmf*s and the conditional marginal *pmf*s for each hypothesis.
4. Compute the likelihood ratio for each parameter. There are b_n possible values of likelihood ratio for the n-th parameter, $0 \leq \tau_n^1 \leq \tau_n^2 \ldots \leq \tau_n^{b_n} \leq \infty$.
5. The threshold candidates for the local likelihood ratio test of each parameter are

$$\tau_n^0 = 0 < \tau_n^1 < \tau_n^2 \ldots < \tau_n^{b'_n} < \tau_n^{b'_n+1} = \infty, \tag{12}$$

   where $\tau_n^1, \tau_n^2, \ldots, \tau_n^{b'_n}$ are the b'_n values of likelihood ratio of the n-th parameter from Step 4, where duplications have been removed ($b'_n \leq b_n$).
6. For each combination $\{\tau_1, \tau_2, \ldots, \tau_N\}$ where τ_n takes a value in $\{\tau_n^0, \tau_n^1, \ldots, \tau_n^{b'_n+1}\}$, determine the fusion rule (γ_0) based on the likelihood ratio test at the fusion center given in (2).
7. For each combination $\{\tau_1, \tau_2, \ldots, \tau_N\}$, evaluate the average probability of error P_e using (3) and (4).
8. Choose the combination that minimizes P_e.

Once the optimal thresholds for the sensors have been computed (off-line), we can carry out the following steps to detect attacks in the system.

Using the optimal thresholds for attack detection:

1. For each record, each local sensor quantizes the parameter into a single bit (indicating whether an attack exists or not).

2. The fusion center collects all the bits from the local sensors and computes the likelihood ratio using (4) (the joint conditional *pmf*'s are drawn from the training data).
3. The fusion center makes the final decision using (2).

If we have a labeled dataset where each record has been marked as "Normal" or "Attack", we can compute the error probabilities as follows:

Computing the probabilities of error:

1. Compute the actual *a priori* probabilities ($\overline{\pi_0}$ and $\overline{\pi_1}$), the false alarm probability ($P_f = P_0(\gamma_0(.) = 1)$ and the misdetection probability ($P_m = P_1(\gamma_0(.) = 0)$.
2. Compute the average probability of error using the equation:

$$P_e = \overline{\pi_0} \times P_f + \overline{\pi_1} \times P_m. \tag{13}$$

5 KDD Cup 1999 data and simulation results

In this section, we first introduce the KDD Cup 1999 data and discuss the application of decentralized detection to these data. We then present the results of the simulation of the algorithm proposed in the previous section using the KDD data.

5.1 KDD Cup 1999 data

As mentioned in the introduction, KDD Cup 1999 [18] is a dataset extracted from the TCP dump data of a LAN. The network was set up to simulate a U.S. Air Force LAN and was speckled with different types of attacks. Each connection (record) consists of 41 parameters and is labeled with either "Normal" or some type of attack. Table 1 describes some parameters of a TCP connection. To apply hypothesis

Feature name	**Description**	**Type**
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of "wrong" fragments	continuous
urgent	number of urgent packets	continuous

Table 1 Basic features of individual TCP connections [18].

testing for network intrusion systems, we can consider the state "Normal" as hypothesis H_0 and a particular type of attack as hypothesis H_1. (For a more general setting, we can group all types of attack into one hypothesis "Attacks" or deal with "Normal" and all types of attacks separately as a multiple hypothesis testing problem with the number of hypotheses, $M > 2$.) We can use the labeled data to learn the conditional distributions of the parameters given each hypothesis. These conditional distributions will then be used to decide the rules for the "sensors" (each of which represents a parameter) and the fusion center. Here, instead of observing the same event, each sensor looks at an aspect of the same event.

For example, we extracted all the records labeled with "Normal" and "Smurf" (which means the connection is a Smurf attack) in the 10% portion of the data given in [18]. We examined the following parameters of all the normal and Smurf connections:

- **duration**: Length (in seconds) of the connection (Table 1).
- **src_bytes**: Number of data bytes from source to destination (Table 1).
- **dst_bytes**: Number of data bytes from destination to source (Table 1).
- **count**: Number of connections to the same host as the current connection in the past two seconds.
- **srv_count**: Number of connections to the same service as the current connection in the past two seconds.

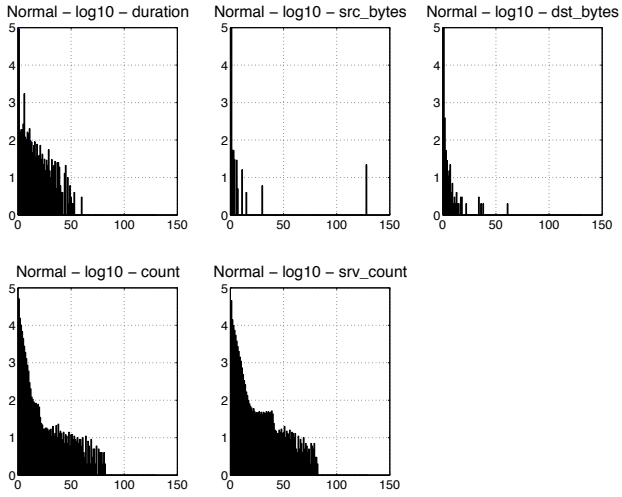

Fig. 4 Probability distributions of some parameters when the LAN is normal. A base-10 logarithmic scale is used for the Y-axis.

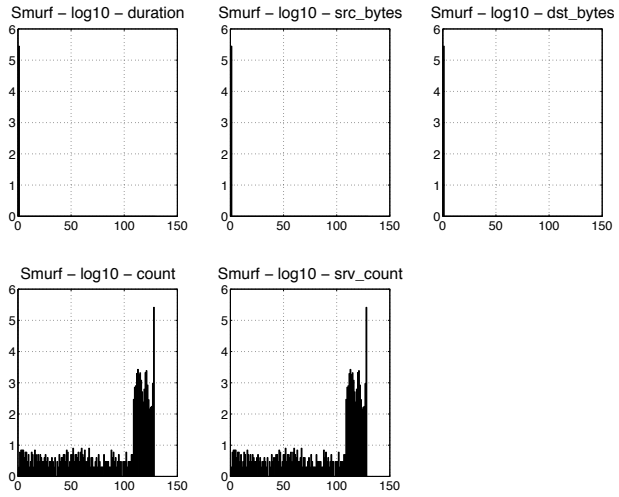

Fig. 5 Probability distributions of some parameters when there are Smurf attacks. A base-10 logarithmic scale is used for the Y-axis.

Figures 4 and 5 show that the conditional distributions of the parameters given either hypothesis can be very different. Also, some parameters are strongly correlated (for example, *count* and *srv_count* given a Smurf attack). Thus, as mentioned earlier, the asymptotic results for large values of N will not be applicable.

5.2 Simulation results

In these simulations, we employ the algorithm and procedures given in Section 4 to detect Smurf attacks against Normal connections in the KDD data ([18])[2].

We use the 10% portion of the dataset (given in [18]) as the training data. The proportion of Normal connections is $\pi_0 = 0.2573$, and the proportion of Smurf connections is $\pi_1 = 0.7427$. Four parameters (*duration*, *src_bytes*, *dst_bytes*, and *count*) are used. The number of bins for each of the parameters is 8.

The threshold candidates for the four parameters *duration*, *src_bytes*, *dst_bytes*, and *count* are given in Table 2. The minimum probability of error computed using the algorithm is $9.3369E-4$. The results show that this probability of error is obtained at different combinations of thresholds, one of which, for example, is $\{1.0082, 1.0003, 1.0004, 1.67\}$.

[2] A Smurf attack can be detected using rule-based detection [21], however, here we just use the dataset as a demonstrative example to illustrate our approach.

duration	0								1.0082	∞
src_bytes	0								1.0003	∞
dst_bytes	0								1.0004	∞
count	0	2.81E-4	3.88E-2	9.60E-2	2.04E-1	2.65E-1	1.67	2.21E2	1.37E4	∞

Table 2 The threshold candidates computed for each parameter. The threshold duplications in the first three parameters have been removed.

The detection procedures are then applied to the whole KDD dataset, which is divided into 10 files for ease of handling. Table 3 provides the simulation results. The probabilities of misdetection, probabilities of false alarm, and the average probabilities of error are plotted in Figures 6 and 7.

File	No Normal	No Smurf	$\overline{\pi_0}$	$\overline{\pi_1}$	P_m	P_f	P_e
1	379669	105556	0.7825	0.2175	0.0061	1.1326E-4	0.0014
2	182718	86493	0.6787	0.3213	0.0028	5.4729E-6	9.1007E-4
3	149880	117038	0.5615	0.4385	0.0035	8.0064E-5	0.0016
4	0	489843	0	1	0.0013	n/a	0.0013
5	0	489843	0	1	0	n/a	0
6	0	489843	0	1	0	n/a	0
7	31046	456829	0.0636	0.9364	0	0	0
8	36798	8189	0.8180	0.1820	0.1260	0	0.0229
9	4061	478090	0.0084	0.9916	6.6724E-4	0	6.6162E-4
10	188609	86162	0.6864	0.3136	0.0037	9.7026E-4	0.0018

Table 3 Probabilities of error for 10 portions (files) of the KDD dataset. We only consider Normal and Smurf connections. *No Normal*: Number of Normal connections in the file; *No Smurf:* Number of Smurf connections in the file. We use *n/a* (not available) for the entries of P_f corresponding to the files with no Normal connections.

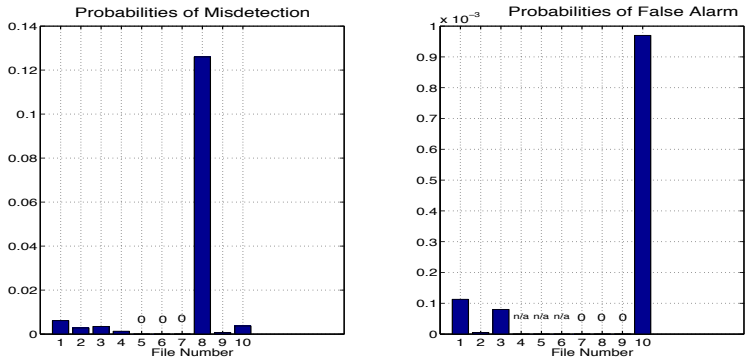

Fig. 6 Misdetection probabilities (left) and false alarm probabilities (right) against file indices (data from Table 3).

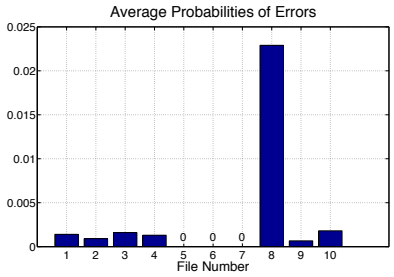

Fig. 7 Average probabilities of error against file indices (data from Table 3).

From the simulation results, we can see that, as expected, the probabilities of error change from file to file, depending on how close the *a priori* probabilities and the conditional joint probabilities of each file are to those of the training data (the simulation of detection using the training data provides exactly the error probability computed from the algorithm, which is $9.3369E-4$). Also, it can be noted that the minimum probability of error should also depend on the number of bins and the way of binning for each parameter. The overall results of the simulation are good, which shows that the algorithm performs well with this dataset.

6 Concluding remarks

In this paper, we have considered the problem of decentralized hypothesis testing with non-*i.i.d.* observations. We have presented the theoretical background for the joint optimization of the likelihood ratio thresholds at the sensors and the fusion rule at the fusion center. We have also derived some relationships between the majority vote and the likelihood ratio test at the fusion center. Building on the theoretical background, we have proposed a search algorithm to compute the optimal thresholds for the sensors. Simulations carried out using the KDD'99 dataset have shown that the algorithm performs well as expected.

Some possible extensions are as follows. First, we can consider the case where the sensors send multiple-bit summaries to the fusion center. Second, multiple-hypotheses testing ($M > 2$) can be used to detect more types of attack. Next, when more parameters are used in detection, PCA can be used to reduce the number of dimensions of the problem. Finally, the tree structure with non-*i.i.d.* observations is an intriguing research direction.

Acknowledgements This work was supported by Deutsche Telekom Laboratories and in part by the Vietnam Education Foundation (VEF). The opinions, findings, and conclusions stated herein are those of the authors and do not necessarily reflect those of VEF. We are grateful to the anonymous reviewers for their valuable comments. The first author would also like to thank Dr. Akshay Kashyap and Jayakrishnan Unnikrishnan for helpful discussions on this subject matter.

References

1. Reuters, The cost of 'code red': $1.2 billion, *USA Today*, Aug. 1, 2001. Available at http://usatoday.com/tech/news/2001-08-01-code-red-costs.htm.
2. J. N. Tsitsiklis, Decentralized detection by a large number of sensors, *Math. Control Signals Syst.*, Vol. 1, No. 2, 1988, pp. 167-182.
3. J. N. Tsitsiklis, Extremal properties of likelihood-ratio quantizers, *IEEE Transactions on Communications*, Vol. 41, No. 4, 1993, pp. 550–558.
4. J. N. Tsitsiklis, Decentralized detection, *Advances in Signal Processing*, JAI Press, 1993.
5. J. Chamberland, V. V. Veeravalli, Asymptotic results for decentralized detection in power constrained wireless sensor networks, *IEEE Journal on Selected Areas in Communication*, Vol. 22, No. 6, 2004, pp. 1007-1015.
6. A. Kashyap, T. Başar, R. Srikant, Asymptotically optimal quantization for detection in power constrained decentralized networks, *Proceedings of the American Control Conference*, Minnesota, USA, 2006.
7. I. Y. Hoballah, P. K. Varshney, Distributed Bayesian signal detection, *IEEE Trans. Inform. Theory*, Vol. 35, 1989, pp. 995–1000.
8. A. K. Jones, R. S. Sielken, Computer system intrusion detection: A survey, Technical Report, Computer Science Department, University of Virginia, 2000.
9. S. Axelsson, Intrusion detection systems: A survey and taxonomy, Technical Report 99–15, Department of Computer Engineering, Chalmers University, 2000.
10. T. Anantvalee, J. Wu, A survey on intrusion detection in mobile ad-hoc networks, *Wireless/Mobile Network Security*, Y. Xiao, X. Shen, D. -Z. Du (eds.), Springer, 2007.
11. A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, H. C. Wong, Decentralized intrusion detection in wireless sensor networks, *Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks*, ACM Press, 2005, pp. 16-23.
12. A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Portland, Oregon, USA, 2004.
13. L. Huang, X. L. Nguyen, M. Garofalakis, M. Jordan, A. D. Joseph, N. Taft, In-network PCA and anomaly detection, *Advances in Neural Information Processing Systems 19*, MIT Press, Cambridge, MA.
14. L. Huang, X. L. Nguyen, M. Garofalakisy, J. M. Hellerstein, M. I. Jordan, A. D. Joseph, N. Tafty, Communication-efficient online detection of network-wide anomalies, *Proceedings of the 26th Annual IEEE Conference on Computer Communications*, Anchorage, Alaska, 2007.
15. N. Ye, X. Li, A Markov chain model of temporal behavior for anomaly detection, *Proceedings IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, 2000.
16. T. Alpcan, T. Başar, A game theoretic analysis of intrusion detection in access control systems, *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, 2004, pp. 1568–1573.
17. T. Alpcan, T. Başar, An intrusion detection game with limited observations, *Proceedings of the 12th Int. Symp. on Dynamic Games and Applications*, Sophia Antipolis, France, 2006.
18. KDD Cup 1999 Data. Available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
19. V. H. Poor, *An Introduction to Signal Detection and Estimation*, 2nd Ed., Springer, 1994.
20. J. Tsitsiklis, M. Athans, On the complexity of decentralized decision making and detection problems, *IEEE Transaction on Automatic Control*, Vol. AC–30, No. 5, 1985, pp. 440–446.
21. W. Lee, S. J. Stolfo, A framework for constructing features and models for intrusion detection systems, *ACM Transactions on Information and System Security*, Vol. 3, No. 4, 2000, pp. 227-261.