Bayesian Decision Aggregation in Collaborative Intrusion Detection Networks

Carol J. Fung[1], Quanyan Zhu[2], Raouf Boutaba[1], and Tamer Başar[2]
[1]David R. Cheriton School of Computer Science,
University of Waterloo, Ontario, Canada, {j22fung,rboutaba}@uwaterloo.ca
[2]Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign, USA. {zhu31,basar1}@illinois.edu

***Abstract*—Cooperation between intrusion detection systems (IDSs) allows collective information and experience from a network of IDSs to be shared to improve the accuracy of detection. A critical component of a collaborative network is the mechanism of feedback aggregation in which each IDS makes an overall security evaluation based on peers opinion and assessment. In this paper, we propose a collaboration framework for intrusion detection networks (CIDNs) and use a Bayesian approach for feedback aggregation by minimizing cost. The proposed model is highly scalable, robust, and cost effective. Experimental results demonstrate an improvement in the true positive detection rate and a reduction in the average cost of our mechanism compared to existing models.**

I. Introduction

In recent years, Internet intrusions are becoming more sophisticated and harder to detect. Intrusions are usually accomplished with the assistance of malicious code (a.k.a malware), including worms, viruses, Trojan, and Spyware. Recent intrusion techniques tend to compromise a large number of nodes to form a botnet [14], and use those compromised nodes to launch distributed attacks such as Distributed Denial of Service attacks [10] or organized attacks such as Fast-Flux service Networks [1].

To protect computer users from malicious intrusions, Intrusion Detection Systems (IDSs) are designed to monitor network traffic or computer activities and alert administrators or computer users about suspicious intrusions. An IDS can be either host-based (HIDS) or network-based (NIDS). Traditional intrusion detection systems work in isolation and can be easily compromised by new or unknown attacks. Collaboration in IDSs enables each IDS to use collective information and experience from other IDSs to achieve more accurate intrusion detections. The overlay network which connects IDSs to exchange information with each other is called a Collaborative Intrusion Detection Network (CIDN).

Several CIDNs [15], [2], [7] have been proposed in the past few years. In most CIDNs, especially worm detection CIDNs (such as Dshield [13] and NetShield [2]), IDSs are distributed in different locations and report intrusion information to the collaborative system. The collected data is usually processed and analyzed in a centralized or distributed manner. However, most CIDNs assume that all collaborators are reliable and trustable [2], [7], [3], which may lead their collaboration system to be vulnerable to malicious insiders. Some recent works on CIDNs [7], [5], [6] propose to use trust models to identify dishonest peers. Intrusion assessments from nodes with different trust values are assigned with different weights to improve intrusion detection accuracy. However, they use a heuristic approach to aggregate feedbacks from others. Some other work [12], [11] proposed to use a Bayesian hypothesis testing methodology to aggregate feedbacks from distributed sensors. Their systems require each sensor to be involved in detecting all intrusions. This condition does not apply to our social network based CIDN.

In this paper, we use a Bayesian approach to devise a decentralized feedback aggregation mechanism for each peer in the CIDN. A Beta distribution is used to model the false positive (FP) rate and true positive (TP) rate of each IDS. We estimate the cost of all possible decisions after feedback aggregation. Only the decision with least cost is chosen. We evaluate the Bayesian aggregation model using a simulation approach and compare it with some other existing aggregation approaches.

The rest of this paper is organized as follows. In Section II, we review some existing CIDNs in the literature and IDS feedback aggregation techniques. Section III is the proposed CIDN framework design. We describe the aggregation problem and propose a Bayesian aggregation solution in Section IV. We show the evaluation results of the Bayesian aggregation model in Section V. Finally, we summarize the main results of the paper and point out some future challenges in Section VI.

II. Related Work

Many architectures have been proposed in the literature, such as Indra [8], DOMINO [15], and NetShield [2]. However, these works did not address the problem that the system might be degraded by some compromised insiders who are dishonest or malicious.

ABDIAS [7] is a community based CIDN where IDSs are organized into groups and exchange intrusion information to

The work of the authors from the University of Waterloo is supported by the Natural Science and Engineering Research Council of Canada under its strategic program. The work of the authors from University of Illinois was in part supported by a grant from Boeing through the Information Trust Institute.

gain better intrusion detection accuracy. A simple majority-based voting system was proposed to detect compromised nodes. However, this voting-based system is vulnerable to colluded voting. Another solution to detect compromised nodes is a trust management system where peers build trust with each other based on personal experience. Existing trust management models for CIDN include the linear model [4], [5] and the Bayesian model [6]. However, all these works use heuristic approaches to aggregate consultation results from other collaborators. In this paper, we propose a Bayesian aggregation model which aims at finding optimal decisions based on collected information.

In the field of intrusion detection, The Bayesian approach has been used in distributed detection. Existing work including [12] and [11] uses Bayesian hypothesis testing methodologies to aggregate the results from sensors distributed in a local area network. However, the methodologies are limited to the context that all participants need to engage in every detection case. While in our context, IDSs may not be involved in all intrusions detection and the collected responses may be from different groups of IDSs each time.

III. Collaboration Framework

The purpose of a CIDN framework is to connect IDSs including HIDSs and NIDSs into a social network. Each IDS can freely choose collaborators on its own benefit. For example, IDSs may choose to collaborate with other IDSs with which they had good experience. We consider that the collaboration participants may have various detection expertise levels and they may act dishonestly or selfishly in collaboration. Therefore, a few features are desirable for an efficient CIDN:

1) It is necessary to have a CIDN endowed with an effective trust evaluation system to reduce the negative impact of dishonest nodes and discover compromised ones.
2) An incentive-compatible collaboration resource allocation mechanism to discourage selfish behaviors and encourage active collaborations.
3) An efficient feedback aggregation algorithm to minimize the cost from false intrusion detection.
4) The system needs to be robust against malicious insiders.
5) Scalability is also a desired feature of the system.

To achieve the preceding goals, we propose a social network-like CIDN (Figure 1). The topology as shown in Figure 1(a) consists of IDSs (nodes) including NIDSs and HIDSs. Nodes are connected if they have a collaborative relationship. Each node maintains a list of other nodes which it currently collaborates with. We call such a list of nodes *acquaintances*. Each node in the CIDN has the freedom to choose its acquaintances based on their trustworthiness. The communication between collaborating nodes are intrusion evaluation requests and corresponding feedbacks. There are two types of requests: *intrusion consultations* and *test messages*. The architecture of the CIDN is shown in Figure 1(b). The collaboration system is composed of seven components, namely, intrusion detection system, communication overlay, trust management, acquaintance management, resource management, feedback aggregation, and test message generator. In the following subsections, we will elaborate on the consultation and test messages and the functionality of each component in the architecture.

A. Consultation message

When an IDS detects some suspicious alerts but does not have enough experience to make a decision whether it should raise an alarm or not, it may send alerts to its acquainted IDSs for diagnosis. Feedbacks from the acquaintances are aggregated and a final alarm decision is made based on the aggregated results. The alert information provided to acquaintances depends on the trust level of each acquaintance. For example, a node may want to share all alert information including data payload with nodes inside its local area network. Some intrusion information might be digested or even removed when sent to acquaintances from the Internet.

B. Test Message

In order for the nodes in the CIDN to gain experience with each other, we propose that IDSs use test messages to evaluate the trustworthiness of others. Test messages are "bogus" consultation requests which are sent to measure the trustworthiness of another node in the acquaintance list. It is sent out in a way that makes it difficult to be distinguished from a real consultation request. The testing node knows the true diagnosis result of the test message and uses the received feedback to derive a trust value for the tested node. This technique can discover inexperienced and/or malicious nodes within the collaborative network.

C. Communication Overlay

Communication overlay is the component which handles all the communications from the host node with other peers in the collaborative network. The messages passing through the communication overlay include: test messages from host node to its acquaintances; intrusion consultations from host node to its acquaintances; feedback from acquaintances; consultation requests from acquaintances; feedback to acquaintances.

D. Trust Management

The trust management component allows IDSs in the CIDN to evaluate the trustworthiness of others based on their personal experience with them. The host node can use test messages to gain experience quickly. Indeed, the verified consultation results can also be used as experience. In our proposed CIDN, we have adopted a Dirichlet-based trust management model [6] to evaluate the trustworthiness of IDSs. In this trust model, IDSs evaluate the trustworthiness of others based on the quality of their feedbacks. The confidence of trust estimation is modeled using Bayesian statistics and the results show that the frequency of test messages is proportional to the confidence level of trust estimation.

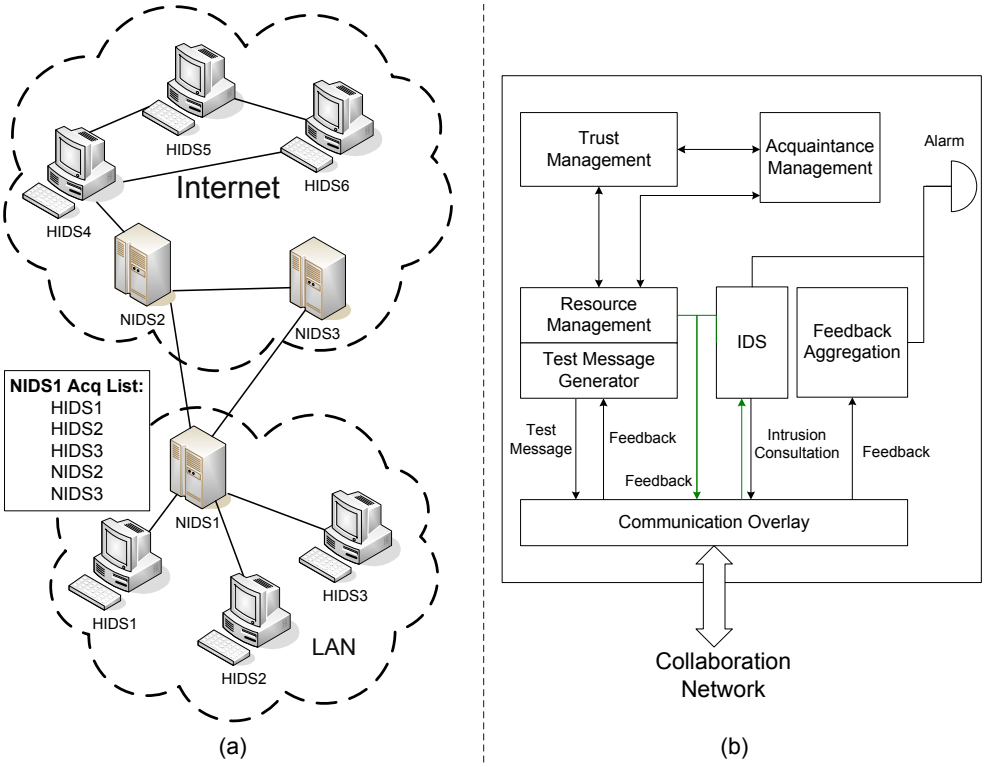

Fig. 1. Design of a Collaborative Intrusion Detection Network: (a) IDN Topology (b) Collaboration Architecture

E. Acquaintance Management

Since each IDS needs to send test messages to its acquaintances to maintain the confidence of trust evaluation, the acquaintance list needs to be limited for the system to be scalable. Other than acquaintances, our system also maintains a consultation list. The nodes on the consultation list are randomly selected from the acquaintances which have passed the probation period. Test messages are sent to all acquaintances while consultation requests are only sent to the nodes in the consultation list. The acquaintance list is updated regularly to recruit new nodes or remove unwanted ones. In our system, we use a fixed length acquaintance list and only keep the most trusted peers in the list and remove the relatively less trusted peers.

F. Resource Management

To prevent some peers from taking advantage of the system and launching a Denial-of-Service attack by sending too many consultation messages to overwhelm the targeted IDSs, a resource management system is required to decide whether the host should allocate resources to respond to each consultation request. An incentive-compatible resource management can assist IDSs to allocate resources to their acquaintances so that other IDSs are fairly treated based on their past assistance to the host IDS. Therefore, an IDS which abusively uses the collaboration resource will be penalized by receiving fewer responses from others. The resource allocation system also decides how often the host should send test messages to its acquaintances, protecting the system from being overloaded. In our CIDN, we use an incentive-compatible resource allocation system [16] for IDSs in the CIDN.

G. Test Message Generator

The functionality of this component is to generate random "bogus" consultation requests for which the results are known beforehand. The feedback of test messages can be used to evaluate the trustworthiness of the feedback sender. It should be difficult to distinguish the generated test messages from regular consultation requests.

H. Feedback Aggregation

Feedback aggregation is a critical component and it has a direct impact on the accuracy of the collaborative intrusion detection. After the host IDS sends out a consultation request to its acquaintances, the collected feedbacks are used to decide whether the host IDS should raise an alarm to the administrator or not. If an alarm is raised, the suspicious intrusion flow will be suspended and the system administrator investigates the intrusion immediately. On one hand, false alarms may waste human resources. On the other hand, undetected intrusions may cause damages. In this paper, we use a Bayesian approach to model the probabilities of false alarms and missing intrusions based on collected information. We then make a decision that leads to the lowest potential cost.

IV. Problem Formulation and Solution

When an IDS observes suspicious activities and does not have enough experience to make an accurate evaluation of

TABLE I
SUMMARY OF NOTATIONS

Symbol	Meaning
$\mathcal{N}$	Set of IDSs in the collaborative network
$\mathcal{A}$	Set of acquaintances of a node
$\mathbf{Y}$	Random vector of complete feedback from a node's acquaintances
$\mathbf{y}$	An instance of complete feedback from a node's acquaintances
F_k, T_k	False positive probability and trust positive probability for acquaintance k
$\mathcal{F}_k, \mathcal{T}_k$	Probability density function of F_k, T_k
π_0, π_1	Prior probability of no-attack and under-attack
τ	Probability threshold for final decision
P	Probability of "under-attack"
X	Random variable to state whether the host IDS is under attack or not
U_g	The utility goal of average cost
C_{fp}, C_{fn}	Marginal cost of making false positive and false negative decisions

potential intrusions, it can send out its observed intrusion information to its acquaintances to ask for diagnosis. The feedback from its acquaintances can be used to make a final decision. The input to the host IDS is the past history of each acquaintance regarding their detection accuracy, as well as their current feedbacks. The output is a decision on whether to raise an alarm or not.

We formulate the feedback aggregation as a Bayesian optimization problem. Consider a set of nodes $\mathcal{N}$ connected into a network, which can be represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The set $\mathcal{E}$ contains the undirected links between nodes, indicating the acquaintances of IDSs in the network.

Let $\mathbf{Y}_i := [Y_j]_{j \in \mathcal{A}_i}$ be an observation vector of an IDS i that contains the feedback from its peers in the acquaintance list $\mathcal{A}_i$. For the convenience of presentation, we drop the subscript i in the notations appearing later in this paper. It should be clear that each IDS i has the same set of characteristic parameters. Suppose node i receives a list of diagnosis results $\mathbf{y} = \{\mathbf{y}_1, ..., \mathbf{y}_{|\mathcal{A}|}\}$ from its acquaintances, where $\mathbf{y}_j \in \{0, 1\}$, $j = 1, 2, \cdots, |\mathcal{A}|$. $\mathbf{y}_j = 0$ means that the j-th acquaintance suggests an intrusion related to the alert, whereas $\mathbf{y}_j = 1$ indicates that the j-th acquaintance suggests no intrusion related to the alert. Our goal is to decide whether the system should raise an alarm to the system administrator based on the current received feedbacks. Table I summarizes the notations we used in this section for readers' convenience.

In the following subsections, we will first model the past behavior of acquaintances. Then, we model the decision problem using Bayesian risk function. In the last subsection, we propose an algorithm to find the minimum number of feedbacks from acquaintances to make a satisfactory decision.

A. Modeling of Acquaintances

Let random variables F_k and T_k denote the false positive (FP) probability and true positive (TP) rate of acquaintance $k \in \mathcal{A}$. FP is the probability that the IDS gives a positive diagnosis (under-attack) given the condition that there is no attack. TP is the probability that the IDS gives correct positive diagnosis under the condition that there is an attack.

Let $\mathcal{F}_k$ and $\mathcal{T}_k$ be the probability density functions of F_k and T_k whose support is on the interval $[0, 1]$. Using the past records as sample data from each acquaintance, we can apply Beta function to estimate $\mathcal{F}_k$ and $\mathcal{T}_k$ as follows:

$$\mathcal{F}_k \sim \quad \text{Beta}(x_k|\alpha_k^0, \beta_k^0) = \frac{\Gamma(\alpha_k^0+\beta_k^0)}{\Gamma(\alpha_k^0)\Gamma(\beta_i^0)} x_k^{\alpha_k^0-1}(1-x_k)^{\beta_k^0-1}, \quad (1)$$

$$\mathcal{T}_k \sim \quad \text{Beta}(y_k|\alpha_k^1, \beta_k^1) = \frac{\Gamma(\alpha_k^1+\beta_k^1)}{\Gamma(\alpha_k^1)\Gamma(\beta_i^1)} y_k^{\alpha_k^1-1}(1-y_k)^{\beta_k^1-1}, \quad (2)$$

where $\Gamma(\cdot)$ is the gamma function [9], parameters α_k^0 and α_k^1 are given by

$$\alpha_k^0 = \textstyle\sum_{j=1}^{u} \lambda^{t_{kj,}^0} r_{k,j}^0 \quad \beta_k^0 = \displaystyle\sum_{j=1}^{u} \lambda^{t_{k,j}^0}(1 - r_{k,j}^0) \quad (3)$$

$$\alpha_k^1 = \textstyle\sum_{j=1}^{v} \lambda^{t_{k,j}^1} r_{k,j}^1 \quad \beta_k^1 = \displaystyle\sum_{j=1}^{v} \lambda^{t_{k,j}^1}(1 - r_{k,j}^1); \quad (4)$$

$r_{k,j}^0 \in \{0, 1\}$ is the j-th diagnosis data from acquaintance k under no intrusion: $r_{k,j}^0 = 1$ means the diagnosis from k is positive while $r_{k,j}^0 = 0$ means otherwise. Similarly, $r_{k,j}^1 \in \{0, 1\}$ is the j-th diagnosis data from acquaintance k under intrusion: $r_{k,0}^1 = 1$ means that the diagnosis from k is positive while $r_{k,0}^1 = 0$ means otherwise. Parameters t_{kj}^0 and $t_{k,j}^1$ denote the time elapsed since the j-th feedback is received. $\lambda \in [0, 1]$ is the forgetting factor on the past experience. $\lambda = 0$ represents a memoryless situation while $\lambda = 1$ indicates the situation where all the past experiences are taken into account on equal basis. u is the total number of non-intrusion cases among the past records and v is the total number of intrusion cases.

To make the parametric updates scalable to data storage and memory, we can use the following recursive formulae to update α_k^0, α_k^1 and β_k^0, β_k^1:

$$\alpha_k^l(t_j) \quad = \quad \lambda^{(t_{k,j}^l - t_{k,j-1}^l)} \alpha_k^l(t_{k,j-1}^l) + r_{k,j}^l; \quad (5)$$

$$\beta_k^l(t_j) \quad = \quad \lambda^{(t_{k,j}^l - t_{k,j-1}^l)} \beta_k^l(t_{k,j-1}^l) + r_{k,j}^l, \quad (6)$$

where $l = 0, 1$ and $j - 1$ indexes the previous data point used for updating α_k^l or β_k^l.

B. Feedback Aggregation

A node receives a feedback vector $\mathbf{y}$ from its acquaintances. Let random variable $X \in \{0, 1\}$ denote the scenario of "no-attack" or "under-attack". The probability of a host IDS being

"under-attack" given the diagnosis results from all acquaintance IDSs can be written as $\mathbb{P}[X=1|\mathbf{Y}=\mathbf{y}]$. Using Bayes' Theorem, we have

$$\mathbb{P}[X=1|\mathbf{Y}=\mathbf{y}]$$

$$=\frac{\mathbb{P}[\mathbf{Y}=\mathbf{y}|X=1]\mathbb{P}[X=1]}{\mathbb{P}[\mathbf{Y}=\mathbf{y}|X=1]\mathbb{P}[X=1]+\mathbb{P}[\mathbf{Y}=\mathbf{y}|X=0]\mathbb{P}[X=0]} \tag{7}$$

Assume that the acquaintances provide diagnoses independently and their FP rate and TP rates are known; then (7) can be further written as

$$\mathbb{P}[X=1|\mathbf{Y}=\mathbf{y}]$$

$$=\frac{\pi_1\prod_{k=1}^{|\mathcal{A}|}T_k^{\mathbf{y}_k}(1-T_k)^{1-\mathbf{y}_k}}{\pi_1\prod_{k=1}^{|\mathcal{A}|}T_k^{\mathbf{y}_k}(1-T_k)^{1-\mathbf{y}_k}+\pi_0\prod_{k=1}^{|\mathcal{A}|}F_k^{\mathbf{y}_k}(1-F_k)^{1-\mathbf{y}_i}},$$

where $\pi_0=\mathbb{P}[X=0]$, $\pi_1=\mathbb{P}[X=1]$, where $\pi_0+\pi_1=1$, are the prior probabilities of the scenarios of "no-attack" and "under-attack". $\mathbf{y}_k$ is the k-th element of vector $\mathbf{y}$.

Since T_k and F_k are both random variables with distributions as in (3), we can see that the conditional probability $\mathbb{P}[X=1|\mathbf{Y}=\mathbf{y}]$ is also a random variable. We use a random variable P to denote the conditional probability $\mathbb{P}[X=1|\mathbf{Y}=\mathbf{y}]$. Then P takes a continuous value over domain $[0,1]$. We denote by $f_P(p)$ the probability density function of P.

Let C_{fp} and C_{fn} denote the marginal cost of a FP decision and a FN decision. We assume there is no cost when a correct decision is made. We use marginal cost because the cost of a FP may change in time depending on the current state. C_{fn} largely depends on the potential damage level of the attack. For example, an intruder intending to track a user's browsing history may have lower C_{fn} than an intruder intending to modify a system file. We define a decision function $\delta(\mathbf{y})\in\{0,1\}$, where $\delta=1$ means raising an alarm and $\delta=0$ means no alarm. Then, the Bayes risk can be written as,

$$\begin{aligned}R(\delta) &= \int_0^1 (C_{fp}(1-x)\delta + C_{fn}x(1-\delta))f_P(x)dx\\ &= \int_0^1 C_{fn}xf_P(x)dx\\ &\quad +\delta\left(C_{fp}-(C_{fp}+C_{fn})\int_0^1 xf_P(x)dx\right)\\ &= C_{fn}\mathbb{E}[P]+\delta(C_{fp}-(C_{fp}+C_{fn})\mathbb{E}[P]),\end{aligned} \tag{8}$$

where $f_P(p)$ is the density function of P. To minimize the risk $R(\delta)$, we need to minimize $\delta(C_{fp}-(C_{fp}+C_{fn})\mathbb{E}[P])$. Therefore, we raise an alarm (i.e. $\delta=1$) if

$$\mathbb{E}[P]\geq\frac{C_{fp}}{C_{fp}+C_{fn}}. \tag{9}$$

Let $\tau=\frac{C_{fp}}{C_{fp}+C_{fn}}$ be the threshold. If $\mathbb{E}[P]\geq\tau$, we raise an alarm, otherwise no alarm is raised. This decision rule can be written as follows:

$$\delta=\begin{cases}1\text{ (Alarm)} & \text{if } \mathbb{E}[P]\geq\tau,\\ 0\text{ (No alarm)} & \text{otherwise.}\end{cases} \tag{10}$$

The corresponding Bayes risk for the optimal decision is:

$$R(\delta)=\begin{cases}C_{fp}(1-\mathbb{E}[P]) & \text{if } \mathbb{E}[P]\geq\tau,\\ C_{fn}\mathbb{E}[P] & \text{otherwise.}\end{cases} \tag{11}$$

C. Gaussion Approximation

When α_i and β_i are sufficiently large, Beta distribution can be approximated by Gaussion distribution according to

$$\text{Beta}(\alpha,\beta)\approx N\left(\frac{\alpha}{\alpha+\beta},\sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}\right).$$

The density function of P can also be approximated using Gaussion distribution. According to Gauss's approximation formula, we have,

$$\begin{aligned}\mathbb{E}[P] &\approx \frac{1}{1+\frac{\pi_0\prod_{k=1}^{|\mathcal{A}|}\mathbb{E}[F_k]^{\mathbf{y}_k}(1-\mathbb{E}[F_k])^{1-\mathbf{y}_k}}{\pi_1\prod_{k=1}^{|\mathcal{A}|}\mathbb{E}[T_k]^{\mathbf{y}_k}(1-\mathbb{E}[T_k])^{1-\mathbf{y}_k}}}\\ &= \frac{1}{1+\frac{\pi_0}{\pi_1}\prod_{k=1}^{|\mathcal{A}|}\frac{\alpha_k^1+\beta_k^1}{\alpha_k^0+\beta_k^0}(\frac{\alpha_k^0}{\alpha_k^1})^{\mathbf{y}_k}(\frac{\beta_k^0}{\beta_k^1})^{1-\mathbf{y}_k}}.\end{aligned} \tag{12}$$

D. Determination of Number of Aggregated Acquaintances

In the consultation period, the number of acquaintances involved in diagnosis may change. A host IDS may consult only part of it acquaintances based on needs. Therefore, we propose that each IDS set a cost goal and consult only sufficient acquaintances to reach the goal. We define the utility goal to be U_g. Upon receiving the i-th feedback, the host IDS compares the expected cost $R(\delta)$ with U_g. If $R(\delta)\leq U_g$, the host IDS stops further consultation and a decision is made immediately. Otherwise, the host IDS needs to consult more acquaintances. The decision rule is presented below.

$$\delta=\begin{cases}\text{Alarm} & \text{if } \mathbb{E}[P]>\tau \text{ and } C_{fp}(1-\mathbb{E}[P])\leq U_g,\\ \text{No alarm} & \text{if } \mathbb{E}[P]\leq\tau \text{ and } C_{fn}\mathbb{E}[P]\leq U_g,\\ \text{Sample more data} & \text{otherwise.}\end{cases}$$

However, when all the acquaintances are used and the utility goal is still not achieved, the host IDS will make a decision according to (10). We describe this dynamic decision making in Algorithm 1.

V. Experiments and Results

In this section, we use a simulation approach to evaluate the efficiency of the Bayesian-based feedback aggregation scheme. We compare the Bayesian aggregation mechanism with other heuristic approaches, such as the simple average aggregation and the weighted average aggregation (to be explained in more detail in this section).

Algorithm 1 Optimal_Decision(U_g, $\mathcal{A}$)

Require: $U_g \geq 0 \vee \mathcal{A} \neq \emptyset$
Ensure: $\delta(U_g, \mathcal{A})$
  $U \Leftarrow \infty$ {U is the current cost.}
  $Q \Leftarrow \frac{\pi_0}{\pi_1}$ {Note that $\mathbb{E}[P] = \frac{1}{1+Q}$ from (12).}
  while $\mathcal{A} \neq \emptyset \wedge U > U_g$ **do**
    $a \Leftarrow$ firstElementOf($\mathcal{A}$)
    $\mathcal{A} \Leftarrow \mathcal{A} \setminus a$
    $r \Leftarrow$ getFeedback(a)
    if $r = 0$ **then**
      $Q \Leftarrow Q \cdot \frac{1-F(a)}{1-T(a)}$
    else
      $Q \Leftarrow Q \cdot \frac{F(a)}{T(a)}$
    end if
    $U \Leftarrow \min\left(\frac{C_{fp}Q}{1+Q}, \frac{C_{fn}}{1+Q}\right)$
  end while
  if $\frac{1}{1+Q} > \frac{C_{fp}}{C_{fp}+C_{fn}}$ **then**
    Raise Alarm
  else
    No Alarm
  end if

We present a set of experiments to evaluate the average cost of the collaborative detection using the Bayesian-based aggregation model in comparison with the simple average and the weighted average models. Each experimental result presented in this section is derived from the average of a large number of replications with an overall negligible confidence interval.

A. Simulation Setting

The simulation environment uses an IDN of n peers. Each IDS is represented by two parameters, expertise level l and decision threshold τ_p. At the beginning, each peer receives an initial acquaintance list containing all the other neighbor nodes. In the process of the collaborative intrusion detection, a node sends out intrusion information to its acquaintances to request for an intrusion assessment. The feedbacks collected from others are used to make a final decision, i.e., whether to raise an alarm or not. Different feedback aggregation schemes can be used to make such decisions. We implement three different feedback mechanisms, namely, simple average aggregation, weighted average aggregation, and Bayesian aggregation. We compare their efficiency by the average cost of false decisions.

1) Simple Average Model: If the average of all feedback is larger than a threshold then raise an alarm.

$$\delta_{SA} = \begin{cases} 1 \text{ (Alarm)} & \text{if } \frac{\sum_{k=1}^{n} \mathbf{y}_k}{n} \geq \tau_{SA}, \\ 0 \text{ (No alarm)} & \text{otherwise,} \end{cases} \tag{13}$$

where τ_{SA} is the decision threshold for the simple average algorithm. It is set to be 0.5 if no cost is considered for making false decisions.

TABLE II
EXPERIMENTAL PARAMETERS

Parameter	Value	meaning
τ_{SA}	0.5	decision threshold of the simple average model
τ_{WA}	0.5	decision threshold of the weighted average model
n	10	number of IDSs in the network
d	0.5	difficulty levels of intrusions and test messages
λ	0.9	forgetting factor
π_0, π_1	0.5	probability of no-attack and under-attack

2) Weighed Average Model: Weights are assigned to feedbacks from different acquaintances to distinguish their detection capability. For example, high expertise IDSs are signed with larger weight compared to low expertise IDSs. In [4], [5], and [6], the weights are the trust values of IDSs:

$$\delta_{WA} = \begin{cases} 1 \text{ (Alarm)} & \text{if } \frac{\sum_{k=1}^{n} w_k \mathbf{y}_k}{\sum_{k=1}^{n} w_k} \geq \tau_{WA}, \\ 0 \text{ (No alarm)} & \text{otherwise,} \end{cases} \tag{14}$$

where w_k is the weight of the feedback from acquaintance k, which is the trust value of acquaintance k in [4], [5], and [6]. τ_{WA} is the decision threshold for the weighted average algorithm. It is fixed to be 0.5 since no cost is considered for FP and FN. In this simulation, we adopt trust values from [6] to be the weights of feedbacks.

3) Bayesian aggregation Model: As described in section IV-B, the Bayesian aggregation approach models each IDS with two features (FP and TP) instead of a single trust value. It also considers the costs of false positive and false negative decisions. A Bayesian decision model investigates the cost of all possible decisions and chooses a decision which leads to a minimal expected cost. The parameters we use are shown in Table II.

B. Modeling of a single IDS

To reflect the intrusion detection capability of each peer, we use a Beta distribution to simulate the decision model of an IDS. A Beta density function is given by:

$$\begin{aligned} f(\bar{p}|\bar{\alpha}, \bar{\beta}) &= \frac{1}{B(\bar{\alpha}, \bar{\beta})} \bar{p}^{\bar{\alpha}-1}(1-\bar{p})^{\bar{\beta}-1}; \\ B(\bar{\alpha}, \bar{\beta}) &= \int_0^1 t^{\bar{\alpha}-1}(1-t)^{\bar{\beta}-1} dt, \end{aligned} \tag{15}$$

where $\bar{p} \in [0, 1]$ is the probability of intrusion assessed by the host IDS. $f(\bar{p}|\bar{\alpha}, \bar{\beta})$ is the probability that a peer with expertise level l answers with a value of $\bar{p}$ to an intrusion assessment of difficulty level $d \in [0, 1]$. Higher values of d are associated with attacks that are difficult to detect, i.e., many peers may fail to identify them. Higher values of l imply a higher probability of producing correct intrusion assessment.

τ_p is the decision threshold of $\bar{p}$. If $\bar{p} > \tau_p$, a peer sends feedback 1 (i.e., under-attack); otherwise, feedback 0(i.e., no-attack) is generated. Let $r \in \{0, 1\}$ be the expected result

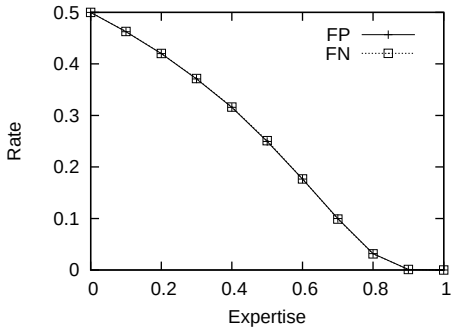

Fig. 2. FP and FN vs. Expertise Level

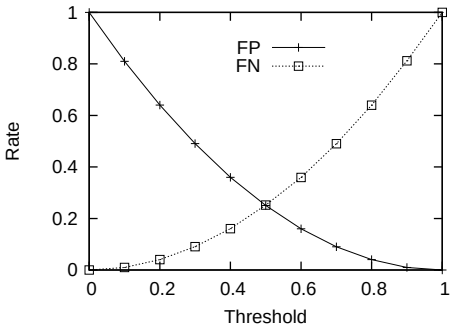

Fig. 3. FP and FN vs. Threshold τ_p

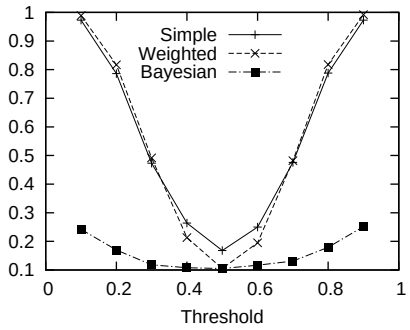

Fig. 4. Average Cost vs. Threshold τ_p

of detection. $r = 1$ indicates that there is an intrusion and $r = 0$ indicates that there is no intrusion. We define $\bar{\alpha}$ and $\bar{\beta}$ as follows.

$$\begin{aligned} \bar{\alpha} &= 1 + \frac{l(1-d)}{d(1-l)} r, \\ \bar{\beta} &= 1 + \frac{l(1-d)}{d(1-l)} (1-r). \end{aligned} \tag{16}$$

For a fixed difficulty level, the preceding model assigns higher probabilities of producing correct intrusion diagnosis to peers with higher level of expertise. A peer with expertise level l has a lower probability of producing correct intrusion diagnosis for intrusions of higher detection difficulty ($d > l$). $l = 1$ or $d = 0$ represent extreme cases where the peer can always accurately detect the intrusion. This is reflected in the Beta distribution by $\bar{\alpha}, \bar{\beta} \to \infty$.

Figure 2 shows that both the FP and FN decrease when the expertise level of an IDS increases. We notice that the curves of FP and FN overlap. This is because the IDS detection density distributions are symmetric under $r = 0$ and $r = 1$. Figure 3 shows that the FP decreases with the decision threshold while the FN increases with the decision threshold. When the decision threshold is 0, all feedbacks are positive; when the decision threshold is 1, all feedbacks are negative.

C. Detection Accuracy and Cost

One of the most important metrics to evaluate the efficiency of a feedback aggregation is the average cost of incorrect decisions. We take into consideration the fact that the costs of FP decisions and FN decisions are different. In the following subsections, we evaluate the cost efficiency of the Bayesian-based aggregation algorithm compared with other models under homogeneous and heterogeneous network settings. Then we study the relation between decision cost and the consulted number of acquaintances.

1) Cost Under Homogeneous Environment: In this experiment, we study the efficiency of the three aggregation models under a homogeneous network setting, i.e., all acquaintances have the same parameters. We fix the expertise levels of all nodes to be 0.5 (i.e., medium expertise) and set $C_{fp} = C_{fn} = 1$ for the fairness of comparison, since the simple average and the weighted average models do not account for the cost difference between FP and FN. We fix the decision threshold for each IDS (τ_p) to 0.1 for the first batch run and then increase it by 0.1 in each following batch run until it reaches 1.0. We measure the average cost of the three aggregation models. As shown in Figure 4, the average costs yielded by Bayesian aggregation remains the lowest among the three under all threshold settings. The costs of the weighted average aggregation and the simple average aggregation are close to each other. This is because under such a homogeneous environment, the weights of all IDSs are the same. Therefore, the difference between the weighted average and the simple average is not substantial. We also observe that changing the threshold has a big impact on the costs of the weighted average model and the simple average model, while the cost of the Bayesian model changes only slightly with the threshold. All costs reach a minimum when the threshold is 0.5 and increase when it deviates from 0.5.

2) Cost Under Heterogeneous Environment: In this experiment, we fix the expertise level of all peers to 0.5 and assign decision thresholds ranging from 0.1 to 0.9 to node 1 to 9 respectively with an increment of 0.1. We set $C_{fp} = 1$ and $C_{fn} = 5$ to reflect the cost difference between FP and FN. We observe the detection accuracy in terms of FP and FN rates and the average costs of false decisions at node 0 when three different feedback aggregation models are used.

Figure 5 shows that the average costs of the three different models converge after a few days of learning process. The cost of Bayesian model starts with a high value and drops drastically in the first 10 days, and finally converges to a stable value on day 30. We then plot in Figure 6 the steady state FP, FN, and the cost. We observe that the weighted average model shows significant improvement in the FP and FN rates and cost compared to the simple average model. The Bayesian aggregation model has a higher FP rate and a lower FN rate compared to the other two models. However, its cost is the lowest among the three. This is because the Bayesian model trades some FP with FN to reduce the overall cost of false decisions.

3) Cost and the Number of Acquaintances: In this experiment, we study the relation between average cost due to false decisions and the number of acquaintances that the host IDS

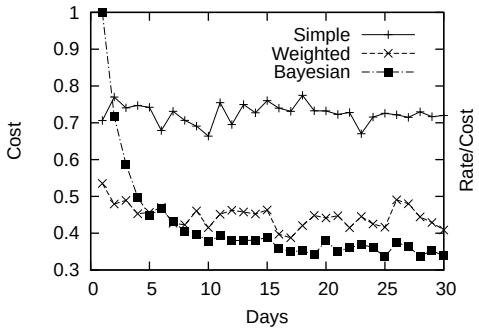

Fig. 5. Average Costs for three different aggregation models

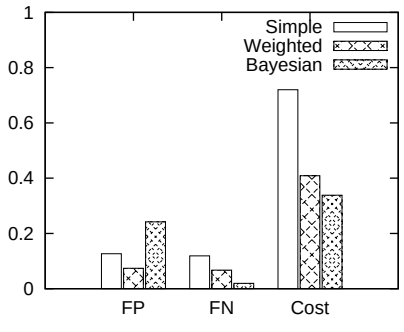

Fig. 6. Comparison of three aggregation models

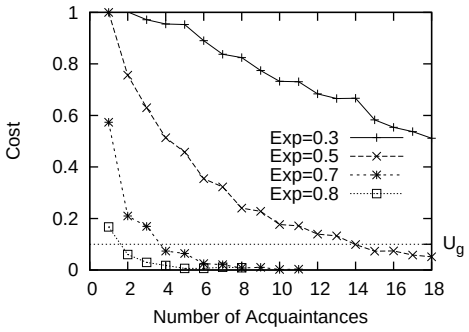

Fig. 7. Average cost vs. Number of Acquaintances Consulted

consults. We fix the expertise level of all IDSs in the network to $0.3, 0.5, 0.7, 0.8$ respectively for different batch runs. Every IDS decision threshold is fixed to 0.5 in all cases. We observe in Figure 7 that, under all cases, the average cost decreases when more acquaintances are consulted. We also notice that for higher expertise acquaintances, fewer consultations are needed to reach the cost goal. For instance, in our experiments, the host IDS only needs to consult 2 acquaintances on the average to reach a cost of 0.1, under the case where all acquaintances are with high expertise level 0.8. Correspondingly, the number of acquaintances needed are 4 and 15 on the average when the acquaintance expertise levels are 0.7 and 0.5 respectively. In the case that all acquaintances are 0.3 (i.e., of low expertise), the utility goal can not be reached after consulting a small number (i.e., < 20) of acquaintances.

D. Robustness and Scalability of the System

Robustness and scalability are two important features of a CIDN. Our proposed CIDN is robust to malicious insiders since it inherits the robust trust management model from [6] where malicious insiders can be quickly discovered and removed from the acquaintance list. The use of probation period in acquaintance management also effectively avoids the impact from malicious newcomers. This CIDN is scalable since the number of acquaintances needed for consultation only depends on the expertise level of those acquaintances rather than the size of the network. Hence the message rate from/to each IDS does not grow with the number of nodes in the network. Furthermore, the dynamic consultation algorithm minimizes the number of consultation messages needed for collaborative intrusion detections.

VI. Conclusion

In this paper, we have described an architecture for a collaborative intrusion detection network. We have then proposed a Bayesian decision based feedback aggregation algorithm. The experimental results indicate that the Bayesian approach reduces the cost of risks from false decisions in comparison to the simple average and weighted average aggregation models. As part of future work, we intent to develop and deploy a real life CIDN using existing intrusion detection systems. Furthermore, we plan to design an effective acquaintance management system to enable robust and effective collaborations among IDSs with a low communication overhead.

References

[1] The honeynet project. know your enemy: Fast-flux service networks, 13 July, 2007. http://www.honeynet.org/book/export/html/130.

[2] M. Cai, K. Hwang, Y. Kwok, S. Song, and Y. Chen. Collaborative internet worm containment. *IEEE Security & Privacy*, 3(3):25–33, 2005.

[3] F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework. In *EEE Symposium on Security and Privacy*, pages 202–215, 2002.

[4] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni. A trust-aware, p2p-based overlay for intrusion detection. In *DEXA Workshops*, 2006.

[5] C. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba. Trust management for host-based collaborative intrusion detection. In *19th IFIP/IEEE International Workshop on Distributed Systems*, 2008.

[6] C. Fung, J. Zhang, I. Aib, and R. Boutaba. Robust and scalable trust management for collaborative intrusion detection. In *11th IFIP/IEEE International Symposium on Integrated Network Management*, 2009.

[7] A. Ghosh and S. Sen. Agent-based distributed intrusion alert system. In *Proceedings of the 6th International Workshop on Distributed Computing (IWDC04)*. Springer, 2004.

[8] R. Janakiraman and M. Zhang. Indra: a peer-to-peer approach to network intrusion detection and prevention. *Proceedings of the 12th IEEE International Workshops on Enabling Technologies*, 2003.

[9] A. Jøsang and R. Ismail. The beta reputation system. *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.

[10] J. Mirkovic and P. Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *SIGCOMM Comput. Commun. Rev.*, 34(2):39–53, 2004.

[11] K. Nguyen, T. Alpcan, and T. Başar. A Decentralized Bayesian Attack Detection Algorithm for Network Security. In *Proceedings of the 23rd International Information Security Conference*, 2005.

[12] J. Tsitsiklis. Decentralized detection. *Advances in Statistical Signal Processing*, pages 297–344, 1993.

[13] J. Ullrich. DShield. http://www.dshield.org/indexd.html.

[14] R. Vogt, J. Aycock, and M. Jacobson. Army of botnets. In *ISOC Symp. on Network and Distributed Systems Security*, 2007.

[15] V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the domino overlay system. In *Proceedings of Network and Distributed System Security Symposium*, 2004.

[16] Q. Zhu, C. Fung, R. Boutaba, and T. Başar. A game-theoretical approach to incentive design in collaborative intrusion detection networks. In *Proceedings of the International Symposium on Game Theory for Networks (GameNets)*, May, 2009.