Ad Hoc Networks 53 (2016) 17–31

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

# FACID: A trust-based collaborative decision framework for intrusion detection networks

Carol J. Fung[a,*], Quanyan Zhu[b]

[a] Department of Computer Science, Virginia Commonwealth University, Richmond, VA, United States
[b] Department of Electrical and Computer Engineering, New York University, 5 Metrotech Center, Brooklyn, NY, United States

ARTICLE INFO

*Article history:*
Received 27 January 2016
Revised 23 July 2016
Accepted 29 August 2016
Available online 20 September 2016

*Keywords:*
Intrusion detection networks
Resource allocations
Cooperative networks
Distributed algorithms

ABSTRACT

Computer systems evolve to be more complex and vulnerable. Cyber attacks have also grown to be more sophisticated and harder to detect. Intrusion detection is the process of monitoring and identifying unauthorized system access or manipulation. It becomes increasingly difficult for a single intrusion detection system (IDS) to detect all attacks due to limited knowledge about attacks. Collaboration among intrusion detection devices can be used to gain higher detection accuracy and cost efficiency as compared to its traditional single host-based counterpart. Through cooperation, a local IDS can detect new attacks that may be known to other IDSs, which may be from different vendors. However, how to utilize the diagnosis from different IDSs to perform intrusion detection is the key challenge. This paper proposes a system architecture of a collaborative intrusion detection network (CIDN), in which trustworthy and efficient feedback aggregation is a key component. To achieve a reliable and trustworthy CIDN, we present a framework called FACID, which leverages data analytical models and hypothesis testing methods for efficient, distributed and sequential feedback aggregations. FACID provides an inherent trust evaluation mechanism and reduces communication overhead needed for IDSs as well as the computational resources and memory needed to achieve satisfactory feedback aggregation results when the number of collaborators of an IDS is large. Our simulation results corroborate our theoretical results and demonstrate the properties of cost efficiency and accuracy compared to other heuristic methods. The analytical result on the lower-bound of the average number of acquaintances for consultation is essential for the design and configuration of IDSs in a collaborative environment.



## 1. Introduction

In recent years, Internet intrusions have become more sophisticated and harder to detect. Intrusions are usually accomplished with the assistance of malicious code (a.k.a. malware), including worms, viruses, and Spyware. Recent attacks tend to compromise a large number of computers/devices to form a botnet [1], and use those compromised nodes to launch distributed attacks such as Distributed Denial of Service (DDoS) attacks [2] or perform organized crime such as Clickfraud [3] and Spamming.

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 categorized into signature-based or anomaly-based. A signature-based IDS compares suspicious code or behavior with known malware or attack patterns. Therefore it should maintain an up-to-date attack signature database to be effective. Alarms are raised when match or similarity are found. Signature-based IDSs are effective to detect known attacks. On the other hand an anomaly-based IDS raises alarms when unusual behaviors or observations are detected. Anomaly-based IDSs do not require a database of known attacks and it is effective to detect unknown attacks. However, the drawback of the anomaly-based detection is the relative high false positive rate compared to the signature-based method. High false positive rate makes it impractical for system administrators to follow up all the alarms, thus it is desirable to keep false positive rate low. Although most IDSs adopt both technologies to have enhanced detection capability, their detection capability is limited by the amount of knowledge their security vendors have, such as the coverage of signature database. Research [4] shows that a single Antivirus vendor has very limited detection rate (up to 60% for newer malware) and collaborative detection can improve the detection rate significantly. Through collaboration, IDSs can utilize the expertise from various vendors to

* Corresponding author.
*E-mail addresses:* cfung@vcu.edu (C.J. Fung), quanyan.zhu@nyu.edu (Q. Zhu).

http://dx.doi.org/10.1016/j.adhoc.2016.08.014
1570-8705/

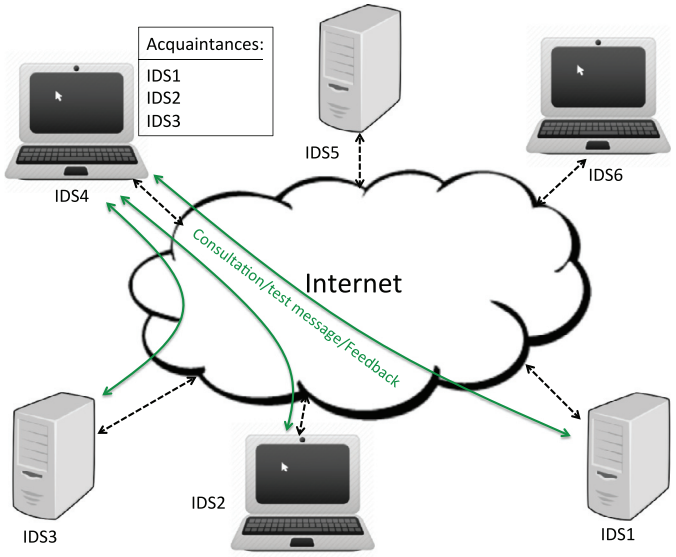



**Fig. 1.** Network overview of a Collaborative Intrusion Detection Network: 6 IDSs are the participants of collaborative intrusion detection network. IDS4 has an acquaintance list containing IDS1, IDS2, and IDS3. It sends a test message or consultation request to its acquaintances for intrusion diagnosis. Acquaintances send their feedback to IDS4.

improve the coverage of known attacks. For example, a malware that cannot be detected by an IDS from vendor A can be detected by an IDS from vendor B. How to utilize multiple IDSs for intrusion detection is the focus of our work in this paper.

Collaborative intrusion detection networks (CIDNs) have distinct features from other types of collaboration networks such as P2P networks and E-commerce networks, where the collaboration is one-time or has a short-term pattern. Collaboration between two parties in an intrusion detection network (IDN) can last for a long period of time. In most social networks, communication for the purpose of testing trust levels is considered ethically inappropriate or costly. However, testing trust level is not an issue in CIDNs because it is "low-cost" and no humans are involved in the loop. For example, to test the trust value of a seller in a e-market, transactions are required so it is costly. However, to test the trust value of an IDS, a host can send *test messages* (a communication overhead generated on purpose to test the reliability of collaborating IDSs. For example, a suspicious attack flow) to the collaborators and compare the feedback with expected answers. This makes feasible using *test messages* (a communication overhead generated on purpose to test the reliability of collaborating IDSs) in CIDN for trust evaluation.

Based on the aforementioned properties, we design a CIDN which utilizes *test messages* to learn the reliability of others and sends *consultation requests* to seek diagnosis from collaborators. The network overview is shown in Fig. 1, where participating IDSs are connected into a collaboration network. Each IDS maintains a list of *acquaintances* (collaborators), and test messages are sent to acquaintances periodically to update its assessment of peer trustworthiness. When an IDS receives intrusion alerts and lacks confidence to determine their reliability (e.g., alert through anomaly detecting), the observation related to the alert messages is sent to acquaintances for diagnosis. An acquaintance IDS analyzes the received intrusion information and replies with a *feedback* – essentially a positive or negative diagnosis. The ambivalent IDS collects feedback from its acquaintances and decides whether an alarm should be raised or not to the administrator. If an alarm is raised, the suspicious intrusion activity/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, fail to detect intrusion may end of having systems compromised. How to balance these two cost to find a compromised optimal point is a challenge.

Although CIDN provides a platform for IDSs to collaborate with each other to achieve higher detection capability, the participating IDS may have different level of expertise. For example, IDS from vendor 1 may be good at detection intrusions but the other IDS may be not as good. It is important to be able to evaluate the capability of IDSs. Trust management permeates the entire CIDN design, which provides every component of the system to a certain level of information assurance. Feedback aggregation is a critical component in the design of CIDN and system integration because it directly affects the performance of the entire system in spite of successful management of resources, acquaintances and trust. It can be deemed as the "last mile" problem of the system. An efficient, scalable and trustworthy aggregation mechanism is hence essential. In this paper, we design a trustworthy framework for feedback aggregation in CIDN, named FACID.[1] Each IDS in the CIDN evaluates its peer acquaintances based on their false positive and false negative rates obtained from historical data and test messages. Accordingly, the assessments received from an incompetent or malicious insider will have less weight in the final decisions. We establish a theoretical framework for the aggregation problem based on data analytical models and hypothesis testing methods. We design optimal decision rules that minimize Bayesian risks of IDSs in the network. In addition, for real-time applications, a host IDS only needs to consult a subset of its acquaintances until desired levels of performance, such as probabilities of detection and false alarm, are achieved. Accordingly, FACID provides a data-driven distributed sequential algorithm for IDSs to make decisions based on feedbacks from a subset of acquaintances. For example, IDS 1 requests help from 6 acquaintances but only 4 of them have responded after some time, then IDS 1 may have sufficient data to make confident decisions without waiting for the last two responses. This way, we also reduce communication overhead between IDSs as well as the computation and memory resources needed to achieve a satisfactory feedback aggregation result when the number of acquaintances of an IDS is large. We investigate four possible outcomes of a decision: *false positive* (FP), *false negative* (FN), *true positive* (TP), and *true negative* (TN). Each outcome is associated with a cost. Our proposed sequential hypothesis testing based feedback aggregation provides improved cost efficiency as compared to other heuristic methods, such as the simple average model [5] and the weighted average model [6,7]. In addition, our approach reduces the communication overhead as it aggregates feedbacks until a predefined FP and TP goal is reached. Our analytical model effectively estimates the number of acquaintances needed for an IDS to reach its predefined intrusion detection goal. This result is crucial for building an IDS acquaintance list in CIDN.

The remainder of this paper is organized as follows. In Section 2, we review existing CIDNs in the literature and IDS feedback aggregation techniques. Section 3 describes the proposed CIDN framework design. The decision aggregation problem is formulated in Section 4, where we use hypothesis testing to minimize the cost of decisions, and sequential hypothesis testing to form consultation termination policy for predefined goals. In Section 6, we perform simulations to evaluate the effectiveness of our aggregation system and validate the analytical model. Section 7 concludes the paper and identifies directions for future research.

[1] FACID stands for "Feedback Aggregation for Collaborative Intrusion Detection".

## 2. Related work

Various CIDN architectures have been proposed in the past, such as Indra [8], DOMINO [9], Worminator [10], Kademlia [11], NetShield [12] and community-based intrusion detection [13]. Recent trend of CIDN involves modern applications such as smart grids [14–16], cloud networks [17,18], or vehicular network [19]. Most of these works have assumed that all acquaintances are reliable and trustable, which makes their collaboration systems vulnerable to malicious insiders. The goal of this paper is to propose a systematic approach to build CIDNs that have inherent trust evaluation mechanisms and provide trustworthy aggregation decisions even in the presence of dishonest and malicious insiders in the collaboration network.

A few recent works on CIDNs [7,20–22] have proposed to use trust or reputation models to identify dishonest peers. Intrusion assessments from nodes with different trust values are assigned with different weights to improve intrusion detection accuracy. DEFIDNET [23] discussed comprehensive adversary model for CIDN. for example, a malicious node in CIDN can block, fabricate, and modify messages exchanged in the CIDN. However, they do not provide corresponding malicious node detection mechanism. On the other hand, ABDIAS [20] is a community based CIDN where IDSs are organized into groups and exchange intrusion information to 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 in the past. Existing trust or reputation management models for CIDNs include the linear model [6,21,22], and the Bayesian model [7]. However, all these works use heuristic approaches to aggregate consultation results from acquaintances. 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 various approach has been used in distributed detection. Existing works including [24] and [25] use hypothesis testing methodologies to aggregate the results from sensors distributed in a local area network. However, the methodologies are limited to the context where all participants need to engage in every detection case. Whereas in our context, IDSs may not be involved in all intrusion detections and the collected responses may come each time from different groups of IDSs. In the work of RevMatch [4], a machine learning approach is used to make collaborative decision. However, this only works for a centralized collaboration model. Our CIDN is based on a distributed collaboration model. In our previous work, Bayesian decision model [26] has been used to make optimal cost decisions based on feedbacks from IDS peers. In this paper, we adopt a more rigorous hypothesis testing model to aggregate feedback from multiple sources and obtain bounds on the number of acquaintances for achieving certain performance goals.

The trustworthiness of CIDNs can be ensured at many levels of the system architecture. In [27,28], a communication protocol with the property of reciprocal incentive compatibility has been used to provide IDS nodes incentives to send feedbacks to their peers, and hence to prevent malicious free-riders, denial-of-service attacks and dishonest insiders. However, this mechanism only ensures the reliability and trustworthiness at the CIDN communication overlay, and does not directly consider the content of the feedback. In [29,30], a knowledge sharing mechanism has been proposed to allow expert nodes to disseminate knowledge within the CIDN to prevent zero-day attacks. The communication protocols in [29] are implemented at the higher application layers of the collaboration network. In this work, we aim at designing a framework that can ensure security and trustworthiness at the "last mile" problem of the system. The design of FACID has inherent trust evaluation compatibility and can provide an additional layer of security as part of the defense-in-depth strategies in CIDN layered architecture design [31,32].

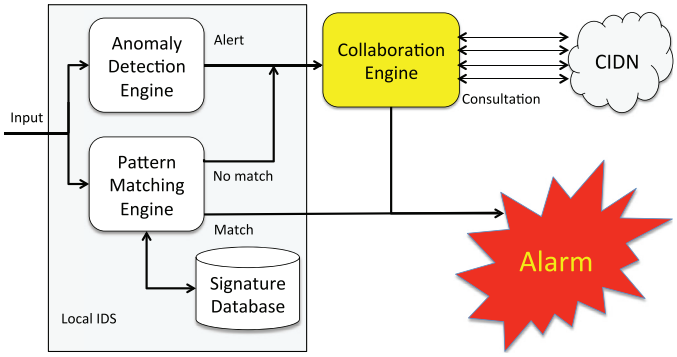



**Fig. 2.** The logic flow on intrusion detection with CIDN: An IDS scans the local input using local signature-based and anomaly-based detection engines. If a signature match is found, the alarm is triggered. If no signature is found but anomaly is detected, then send the input to the collaboration engine which utilizes CIDN for detection. If the CIDN suggests attacks found, then trigger the alarm.

## 3. Collaboration framework

As we mentioned, one major challenge is how to utilize the diagnosis from different IDSs to perform intrusion detection. In this work, we will first propose a Collaborative Intrusion Detection Network (CIDN) design based on an overlay network connecting collaborating IDSs. We then present the collaborative decision model which can be used to effectively aggregate the decision from multiple IDSs.

As shown in Fig. 1, IDSs from different security vendors are connected in a peer-to-peer manner. Each IDS selectively maintains a list of "good" acquaintances based on its own interest. For example, IDSs may choose to collaborate with other IDSs that they have had good experience with in the past. We consider that the collaboration participants may have various detection expertise levels and they may act dishonestly or selfishly in collaboration. This assumption is particularly true when collaboration happens between different vendors. Consequently, the following features are desirable for an efficient CIDN:

(1) An effective trust evaluation to reduce the negative impact of dishonest nodes and discover malicious ones.
(2) An incentive-compatible resource allocation to discourage selfish behaviors and encourage active collaborations.
(3) An efficient feedback aggregation to minimize the cost of false intrusion detection.
(4) A robust behavior against malicious insiders.
(5) Ability to scale with the network size.

We design our CIDN to satisfy all the above features through component integration. The topology as shown in Fig. 1 consists of peer IDSs (nodes). Nodes are connected if they have a collaboration relationship. Each node maintains a list of other nodes with which it currently collaborates with. We name such a list of nodes *acquaintances* list. Each node in the CIDN has the freedom to choose its acquaintances based on their own interest. The communications between collaborating nodes are requests for intrusion alert evaluation and their corresponding feedbacks. There are two types of requests: *intrusion consultations* and *test messages* (definitions in following sub sections). As shown in the logic flow in Fig. 2, when a local IDS detects anomaly but no local matching attack signature is found, it can turn to its acquaintances for opinions. The feedback from acquaintances will be used to decide whether to raise

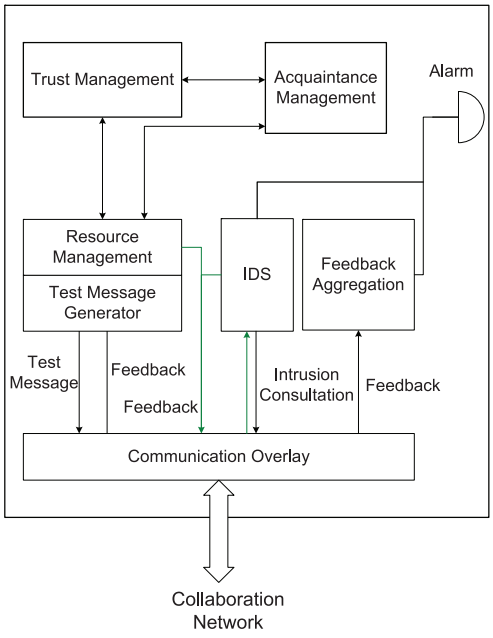



**Fig. 3.** Architecture Design of a CIDN: All system components in the design can be implemented in a distributed way. Each IDS can generate a test message or a request for intrusion consultation and send it to its peers in the acquaintance list through a communication overlay. The resource management component allocates communication and computational resources, and updates the acquaintance list. The trust management component estimates the trust values of acquaintances. Feedback aggregation is the essential part of the CIDN which makes critical intrusion detection decisions based on collected feedbacks. Alltogether, FACID provides an inherent trust evaluation mechanism, and enables efficient, distributed and trustworthy sequential feedback aggregations.

an alarm or not. One may argue that an attacker may compromise an IDS but keeps responding to consultation messages in a normal way in order to profile the target system. However, profiling target system through inspecting consultation requests is not practical since the target system also sends test messages that are indistinguishable from real consultation messages. This way the profiling results may not be correct. Therefore, there is no incentive for attacker to compromise an IDS but remain the same behavior in consultation.

The architecture of the CIDN is shown in Fig. 3. 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 of the architecture.

### 3.1. Consultation messages

When an IDS detects suspicious activities but does not have enough experience to make a decision whether it should raise an alarm or not, it sends 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. A node may want to share all alert information including data payload with the nodes inside its local area network. Some intrusion information might be digested or even removed when sent to acquaintances from the Internet. For example, a suspicions file can be hashed and only the hashed signature is sent for diagnosis [33]. Payload can also be removed and only the headers are used for intrusion detection such as some lightweight IDS [34]. This may compromise the precision of intrusion detection due to limited information provided but a higher processing rate can be achieved since lower amount of data is involved [35].

### 3.2. Test messages

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. They are sent out in a way that makes them 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 collaboration network.

### 3.3. Communication overlay

The communication overlay is the component which handles all the communications between the host node and other peers in the collaboration 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. The communication overlay dispatches received requests to corresponding components in the system and routes requests and messages from local hosts to their destinations. For example, when the communication layer receives a consultation request, it calls local IDS component for diagnosis and returns the received feedback (diagnosis result) back to the sender.

### 3.4. Trust management

The trust management component allows IDSs in the CIDN to evaluate the trustworthiness of others based on personal experience with them. The definition of trust in our context is the likelihood that an IDS is reliable in terms of giving useful feedback. 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 [7,36] 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. Trust management model is closely connected to the resource management and acquaintance management since the evaluation results of collaborator are essential inputs for the latter. Trust value is a critical criteria for IDSs to decide whether to maintain a collaboration relationship with another or not [37].

### 3.5. 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 a 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 dynamic acquaintance management system [37] to recruit higher quality peers and remove less helpful

peers based on their trustworthiness and expertise in intrusion detection.

### 3.6. 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, the resource management component decides 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 [27] for IDSs in the CIDN.

### 3.7. 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. A test message can be a permutation of the previous consultation message with which the ground truth has been verified, or a random pick taken from its knowledge database. A permutation algorithm can be used to automatically generate unique test messages each time based on existing ones. This way adversaries will not be able to identify test messages.

### 3.8. Network join process

Before joining the network, a IDS needs to register to a trusted digital certificate authority and get a public and private key pair which uniquely identifies it. Note that we identify the (machine, user) tuple. This is because a different machine means a different IDS instance. When a new IDS joins a network, it will contact the seed node in the network, which can be publicized or privately communicated. After that it is provided with a preliminary acquaintance list through a random traversing in the network. This list is customizable and contains identities (or public keys) of other peers within the network along with their trust values. Communication between IDSs are encrypted using the public/private key pair. All acquaintances will need to pass a probation period for being truly trusted. Note that for a good acquaintance the initial trust can be low but it will build up after responding to test messages and consultation messages during the probation period. Acquaintances with low trust values after probation period will be replaced by new ones to keep the quality of the acquaintance list.

### 3.9. Feedback aggregation

When the IDS of the host computer cannot provide a confident intrusion diagnosis result of a given alert, the host node may consult with other IDSs in the collaboration network for opinions/diagnosis. The received opinion (feedback) set is then used to make a decision whether the host IDS should raise an alarm to its administrator or not. The feedback aggregation component is responsible for making a decision based on the feedback set. It decides not only on which criterion to use to measure the quality of decisions, but also on how to reach such a decision in an efficient way. This is one of the most important components since it has a direct impact on the accuracy of the collaborative intrusion detection. 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 and spread in the network. In this paper, we design a feedback aggregation system, FACID, to make final intrusion detection decisions. We measure the rate of false alarms, i.e., false positive (FP) rate, and the rate of missing intrusions, i.e., false negative (FN) rate, of participating IDSs based on collected experience with them in the past. FACID offers two modes of operation. One is a sequential mode in which acquaintances are requested one after another until performance has been achieved. Another mode is non-sequential, where requests are sent to all known acquaintances. The former one is useful for resource-constrained large-scale networks and time-insensitive IDS applications. The latter one, however, is used for small networks where nodes have a small number of acquaintances.

In the next section, we will explain the mathematical model for collaborative decision making. We analyze the false positive cost and false negative cost. We then provide a hypothesis testing model to find a decision which leads to minimum overall cost.

## 4. System model

In a CIDN, how to utilize the collected responses (feedback) from other collaborating IDSs is the key component of the design. In this section, we establish a framework for feedback aggregation in CIDNs. Consider a set of $N$ nodes, $\mathcal{N} := \{1, 2, \cdots, N\}$, connected in 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. An IDS node $i \in \mathcal{N}$ has a set of $n_i$ acquaintances, denoted by $\mathcal{N}_i \subseteq \mathcal{N}$, with $n_i = |\mathcal{N}_i|$. When node $i$ observes suspicious activities and does not have enough experience to make an accurate evaluation of potential intrusion, 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.

Let $Y_j^i, j \in \mathcal{N}_i$, be a random variable denoting the decision of peer IDS $j$, $j \in \mathcal{N}_i$, on its acquaintance list $\mathcal{N}_i$ of node $i$. The random variable $Y_j^i$ takes binary values in $\mathcal{Y} := \{0, 1\}$ for all $j \in \mathcal{N}_i, i \in \mathcal{N}$. In the intrusion detection setting, $Y_j^i = 0$ means that IDS $j$ decides and reports to IDS $i$ that there is no intrusion, while $Y_j^i = 1$ means that IDS $j$ raises an alarm of possible detection of intrusion to IDS $i$. Each IDS makes its decision based upon its own experience of the previous attacks and its own sophistication of detection. We let $p_j^i$ as the probability mass function defined on $\mathcal{Y}$ such that $p_j^i(Y_j^i = 0)$ and $p_j^i(Y_j^i = 1)$ denote the probability of reporting no intrusion and the probability of reporting intrusion from IDS $j$ to IDS $i$, respectively.

We let $\mathbf{Y}^i := [Y_j^i]_{j \in \mathcal{N}_i} \in \mathcal{Y}^{n_i}$ be an observation vector of IDS $i$ that contains feedbacks from its peers in the acquaintance list, where $\mathcal{Y}^{n_i}$ is a binary space of length $n_i$. Each IDS has two hypotheses $H_0$ and $H_1$. $H_0$ hypothesizes that no intrusion is detected whereas $H_1$ forwards a hypothesis that an intrusion is detected and an alarm needs to be raised. Note that we intentionally drop the superscript $i$ on $H_0$ and $H_1$ because we assume that each IDS attempts to make the same type of decisions. Denote by $\pi_0^i, \pi_1^i$ the apriori probabilities on each hypothesis such that $\pi_0^i = \mathbb{P}[H_0], \pi_1^i = \mathbb{P}[H_1]$ and $\pi_0^i + \pi_1^i = 1$, for all $i \in \mathcal{N}$. Let $p^i$ be the probability measure on $\mathcal{Y}^{n_i}$, for all $i \in \mathcal{N}$. The conditional

**Table 1**
Summary of notations.

| Symbol | Meaning |
|---|---|
| $\mathcal{N}$ | Set of IDSs in the collaboration network |
| $\mathcal{N}_i$ | Set of acquaintances of IDS $i$, $i \in \mathcal{N}$ |
| $\mathcal{G}$ | The network of CIDN. |
| $\mathcal{E}$ | The edges connecting acquaintances in CIDN. |
| $n_i$ | Number of acquaintances of IDS $i$, $i \in \mathcal{N}$ |
| $Y_j^i$ | Reported decisions from IDS $j$ to IDS $i$, $i \in \mathcal{N}$, $j \in \mathcal{N}_i$ |
| $\mathbf{Y}^i$ | Vector of complete feedbacks from IDS $i$'s acquaintances |
| $H_0$ | Hypothesis that there is no intrusion |
| $H_1$ | Hypothesis that there is an intrusion |
| $p_{j,M}^i$, $p_{j,D}^i$ | False negative rate and true positive rate from $j$ to $i$. |
| $p_{j,F}^i$ | False positive rate from $j$ to $i$. |
| $\lambda_F^i$, $\lambda_D^i$ | Discount factor for false positives and true positives. |
| $r_{j,F,k}^i$ | The diagnosis result at time $k$ from acquaintance $j$ to IDS $i$ given that there is no intrusion |
| $r_{j,D,k}^i$ | The diagnosis result at time $k$ from acquaintance $j$ to IDS $i$ given that there is an intrusion |
| $\pi_0^i$, $\pi_1^i$ | The prior probability of no-attack and under-attack |
| $\tilde{\tau}^i$ | The probability threshold for final decision |
| $L^i$ | Likelihood ratio for IDS $i$'s decision |
| $L_n^i$ | Likelihood ratio for IDS $i$'s sequential decision at stage $n$ |
| $R^i$ | Bayesian risk of IDS $i$ |
| $\delta^i$ | Aggregation decision rule of IDS $i$ |
| $\phi^i$ | Stopping decision rule of IDS $i$ |
| $D_{KL}(p_1 \| p_2)$ | Kullback-Leibler divergence between distributions $p_1$ and $p_2$ |
| $C_{10}^i$, $C_{01}^i$ | Cost of making false positive and false negative decisions for IDS $i$ |
| $C_{00}^i$, $C_{11}^i$ | Cost of making correct decisions for IDS $i$ |

probabilities $p^i(\mathbf{Y}^i = \mathbf{y}^i|H_l), l = 0, 1,$ denote the probabilities of a complete feedback being $\mathbf{y}^i \in \mathcal{Y}^{n_i}$ given the hypothesis $H_0$, $H_1$, respectively. Assuming that peers make decisions independently (this is reasonable if acquaintances are appropriately selected), we can rewrite the conditional probability as

$$p^i(\mathbf{Y}^i = \mathbf{y}^i|H_l) = \prod_{j \in \mathcal{N}_i} p_j^i(Y_j^i = y_j^i|H_l), \ \ i \in \mathcal{N}, l = 0, 1. \tag{1}$$

Our goal is to decide whether the system should raise an alarm to the system administrator based on the current received feedbacks. We need to point out that the feedback aggregation does not exclude the diagnosis of the local host IDS itself. If an IDS is capable of making its independent diagnosis, it should be aggregated together with the feedbacks from its peers in the acquaintance list. Table 1 summarizes the notations we use in this section for readers' convenience.

In the following subsections, we first model the past behavior of acquaintances and then model the decision problem using Bayesian risk function.

### 4.1. Modeling of acquaintances

The conditional probabilities $p_j^i(Y_j^i|H_l), i \in \mathcal{N}, j \in \mathcal{N}_i, l \in \{0, 1\}$, are often unknown to IDS nodes and they need to be learned from previous data. In this section, we use the beta distribution and its Gaussian approximation to find these probabilities. We let $p_{j,M}^i := p_j^i(Y_j^i = 0|H_1)$ be the probability of miss of an IDS $j$'s diagnosis to node $i$'s request, also known as the false negative (FN) rate; and let $p_{j,F}^i := p_j^i(Y_j^i = 1|H_0)$ be the probability of false alarm or false positive (FP) rate. The probability of detection, or true positive (TP) rate, can be expressed as $p_{j,D}^i = 1 - p_{j,M}^i$.

Each IDS in the network maintains a history of data containing the diagnosis data from past consultations. The accuracy of peer diagnosis will be revealed after an intrusion happens. As mentioned in Section 3, test messages can also be used to assess the effectiveness of IDSs even though no intrusion history has been collected. IDS $i$ can use these collected data from its peers to assess the distributions over its peer IDS $j$'s probabilities of detection and false alarm using beta functions, denoted by $p_{j,D}^i$ and $p_{j,F}^i$, respectively. The total reported diagnosis data from peer IDS $j$, $j \in \mathcal{N}_i$, to IDS $i$ is denoted by the set $\mathcal{M}_j^i$, and they are classified into two groups: one is where the result is either false positive or true negative under no intrusion, denoted by the set $\mathcal{M}_{j,0}^i$; and the other is where the result is either false negative or true positive under intrusion, denoted by the set $\mathcal{M}_{j,1}^i$. Both sets are disjoint satisfying $\mathcal{M}_{j,0}^i \cup \mathcal{M}_{j,1}^i = \mathcal{M}_j^i$ and $\mathcal{M}_{j,0}^i \cap \mathcal{M}_{j,1}^i = \emptyset$.

We let the random variables $p_{j,F}^i$ and $p_{j,D}^i$ take the form of beta distributions as follows:

$$p_{j,F}^i \sim \text{Beta}(x_j^i|\alpha_{j,F}^i, \beta_{j,F}^i) = \frac{\Gamma(\alpha_{j,F}^i + \beta_{j,F}^i)}{\Gamma(\alpha_{j,F}^i)\Gamma(\beta_{j,F}^i)} (x_j^i)^{\alpha_{j,F}^i - 1}(1 - x_j^i)^{\beta_{j,F}^i - 1}, \tag{2}$$

$$p_{j,D}^i \sim \text{Beta}(y_j^i|\alpha_{j,D}^i, \beta_{j,D}^i) = \frac{\Gamma(\alpha_{j,D}^i + \beta_{j,D}^i)}{\Gamma(\alpha_{j,D}^i)\Gamma(\beta_{j,D}^i)} (y_j^i)^{\alpha_{j,D}^i - 1}(1 - y_j^i)^{\beta_{j,D}^i - 1}, \tag{3}$$

where $\Gamma(\cdot)$ is the Gamma function; $x_j^i, y_j^i \in [0, 1]$; $\alpha_{j,F}^i, \alpha_{j,D}^i$ and $\beta_{j,F}^i, \beta_{j,F}^i$ are beta function parameters that are updated according to historical data as follows.

$$\alpha_{j,F}^i = \sum_{k \in \mathcal{M}_{j,0}^i} (\lambda_F^i)^{t_{j,k}^i} r_{j,F,k}^i, \ \ \beta_{j,F}^i = \sum_{k \in \mathcal{M}_{j,0}^i} (\lambda_F^i)^{t_{j,k}^i} (1 - r_{j,F,k}^i); \tag{4}$$

$$\alpha_{j,D}^i = \sum_{k \in \mathcal{M}_{j,1}^i} (\lambda_D^i)^{t_{j,k}^i} r_{j,D,k}^i, \ \ \beta_{j,D}^i = \sum_{k \in \mathcal{M}_{j,1}^i} (\lambda_D^i)^{t_{j,k}^i} (1 - r_{j,D,k}^i). \tag{5}$$

The introduction of the discount factors $\lambda_F^i, \lambda_D^i \in [0, 1]$ above puts more weights on recent data from IDSs than the old ones. The discount factors on the data can be different for false negative and false positive rates. The parameter $t_{j,k}^i$ denotes the time when $k$-th diagnosis data is generated by IDS $j$, $j \in \mathcal{N}_i$, to its peer IDS $i$. The parameters $r_{j,F,k}^i, r_{j,M,k}^i \in \{0, 1\}$ are the revealed results of the $k$-th diagnosis data: $r_{j,F,k}^i = 1$ suggests that the $k$-th diagnosis data from peer $j$ yields an undetected intrusion while $r_{j,F,k}^i = 0$ means otherwise; similarly, $r_{j,D,k}^i = 1$ indicates the data from the peer $j$ results in a correct detection under intrusion, and $r_{j,D,k}^i = 0$ means otherwise. The choice of discount factors is often chosen

based on two considerations. One is related to system constraints on memory and storage. If the system has a sufficiently large scale storage or memory, then it can store all the data and make decisions based on all the past data. Another consideration is on the relevance of the past data on the decision- making. When the environment changes quickly, the past data may have less relevance on the current decision-making. In general the discount factor can take any values between 0 and 1.

The parameters $\alpha^i_{j,F}, \beta^i_{j,F}, \alpha^i_{j,D}, \beta^i_{j,D}$ in the distribution above also provide an empirical assessment of the trustworthiness of each peer of IDS $i$. They can be also seen as the trust values of the acquaintances. Accordingly, a peer who is either malicious or incompetent will have low values of $\alpha^i_{j,D}$ and higher values $\alpha^i_{j,F}$. To make the parametric updates scalable to data storage and memory, we can use the following recursive formulae to update these parameters as follows:

$$\alpha^i_{j,e,k} = (\lambda^i_e)^{t^i_{j,k}-t^i_{j,k-1}}\alpha^i_{j,e,k-1} + r^i_{j,e,k}, \quad k \geq 1, \tag{6}$$

$$\beta^i_{j,e,k} = (\lambda^i_e)^{t^i_{j,k}-t^i_{j,k-1}}\beta^i_{j,e,k-1} + r^i_{j,e,k}, \quad k \geq 1, \tag{7}$$

where $e \in \{F, D\}$; $\alpha^i_{j,D,k}, \beta^i_{j,D,k}$, are parameter values up to $k$-th data point in their corresponding data set $\mathcal{M}^i_{j,1}$; $\alpha^i_{j,F,k}, \beta^i_{j,F,k}$, are parameter values up to $k$-th data point in their corresponding data set $\mathcal{M}^i_{j,0}$. We can see that when $\lambda^i_e = 0$, the system becomes memoryless, and when $\lambda^i_e = 1$, all past experiences are taken into account on equal basis. The online iterative calculations also provide a method to assess the trust values with real time data.

When parameters of the beta functions $\alpha$ and $\beta$ in (2) are sufficiently large, i.e., enough data are collected, beta distribution can be approximated by a Gaussian distribution as follows:

$$\text{Beta}(\alpha, \beta) \approx \text{G}\left(\frac{\alpha}{\alpha+\beta}, \sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}\right), \tag{8}$$

where the arguments of $G(\cdot, \cdot)$ are the mean value and the standard deviation, respectively. Note that we have dropped the superscripts and subscripts in (8) for generality as it can be applied to all $i$ and $j$ in (2). Hence, using the Gaussian approximation and (4), the expected values for $p^i_{j,D}$ and $p^i_{j,F}$ are given by

$$\mathbb{E}[p^i_{j,F}] = \frac{\alpha^i_{j,F}}{\alpha^i_{j,F}+\beta^i_{j,F}}, \quad \mathbb{E}[p^i_{j,D}] = \frac{\alpha^i_{j,D}}{\alpha^i_{j,D}+\beta^i_{j,D}}. \tag{9}$$

The mean values in (9) under large data can be intuitively interpreted as the proportion of results of false alarm and detection in the set $\mathcal{M}^i_{j,0}$ and $\mathcal{M}^i_{j,1}$, respectively. They can thus be used in (1) as the assessment of the conditional probabilities.

### 4.2. Feedback aggregation

The feedback aggregation problem of IDS $i$ can be seen as a hypothesis testing problem in which one finds a decision function $\delta^i(\mathbf{Y}^i) : \mathcal{Y}^{n_i} \to \{0, 1\}$ to minimize the Bayes risk of IDS $i$

$$R^i(\delta^i) = R^i_0(\delta^i|H_0)\pi^i_0 + R^i_1(\delta^i|H_1)\pi^i_1, \tag{10}$$

where $R^i(\delta^i|H_0)$ is the cost of false alarm and $R^i(\delta^i|H_1)$ is the cost of missed detection. An optimal decision function partitions the observation space $\mathcal{Y}^{n_i}$ into two disjoint sets $\mathcal{Y}^i_0$ and $\mathcal{Y}^i_1$, where $\mathcal{Y}^i_0 = \{\mathbf{y}^i : \delta^i(\mathbf{y}^i) = 0\}$, and $\mathcal{Y}^i_1 = \{\mathbf{y}^i : \delta^i(\mathbf{y}^i) = 1\}$.

To find an optimal decision function according to some criterion, we introduce the cost function $C^i_{ll'}, l, l' = 0, 1$, which represents IDS $i$'s cost of deciding that $H_l$ is true when $H_{l'}$ holds. More specifically, $C^i_{01}$ is the cost associated with a missed intrusion or attack and $C^i_{10}$ refers to the cost of false alarm, while $C^i_{00}, C^i_{11}$ are the incurred costs when the decision meets the true situation. Let

$$R^i_0(\delta^i|H_0) = C^i_{10}p^i[\delta^i = 1|H_0] + C^i_{00}p^i[\delta^i = 0|H_0], \tag{11}$$

$$R^i_1(\delta^i|H_0) = C^i_{01}p^i[\delta^i = 0|H_1] + C^i_{11}p^i[\delta^i = 1|H_1]. \tag{12}$$

It can be shown that the optimal decision functions is a function of the likelihood ratio given by $L^i(\mathbf{y}^i) = \frac{p^i(\mathbf{y}^i|H_1)}{p^i(\mathbf{y}^i|H_0)}$ (see [24,25]).

A threshold Bayesian decision rule is expressed in terms of the likelihood ratio and is given by

$$\delta^i_B(\mathbf{y}^i) = \begin{cases} 1 & \text{if } L^i(\mathbf{y}^i) \geq \tau^i \\ 0 & \text{if } L^i(\mathbf{y}^i) < \tau^i \end{cases}, \tag{13}$$

where the threshold $\tau^i$ is defined by

$$\tau^i = \frac{(C^i_{10} - C^i_{00})\pi^i_0}{(C^i_{01} - C^i_{11})\pi^i_1}. \tag{14}$$

If the costs are symmetric and the two hypothesis are equal likely, then the rule in (13) reduces to the maximum likelihood (ML) decision rule

$$\delta^i_{ML}(\mathbf{y}) = \begin{cases} 1 & \text{if } p^i(\mathbf{y}^i|H_1) \geq p^i(\mathbf{y}^i|H_0) \\ 0 & \text{if } p^i(\mathbf{y}^i|H_1) < p^i(\mathbf{y}^i|H_0) \end{cases}, \tag{15}$$

Assume that $C^i_{00}, C^i_{11} = 0$. Using the acquaintance models in Section 4.1, we can obtain the following distributed decision rule for each IDS. The application of the optimal decision rules is summarized in Algorithm 1.

**Algorithm 1** Optimal decision rule for an IDS $i$.

**Step 1**: Send out requests to all acquaintances of IDS $i$ and collect their feedback results.
**Step 2**: Use (16) to decide whether an intrusion occurs or not, and take corresponding actions.
**Step 3**: Update the data sets $\mathcal{M}^i_{j,0}$, $\mathcal{M}^i_{j,1}$, with the diagnosis results of each peer $j$, $j \in \mathcal{N}_i$ when the fact has been revealed a posteriori.
**Step 4**: Calculate beta function parameters $\alpha^i_{j,F}, \alpha^i_{j,D}$ and $\beta^i_{j,F}, \beta^i_{j,F}$ using iterative schemes (6) and (7).
**Step 5**: Go to **Step 1** when new decisions need to be made or the trustworthiness of new acquaintances need to be evaluated using test messages.

**Proposition 4.1.** *Let* $\bar{\tau}^i := \frac{C^i_{10}}{C^i_{10}+C^i_{01}}$ *and assume that historical data is relatively large. The optimal decision rule of IDS* $i$, $i \in \mathcal{N}$, *is a threshold rule*

$$\delta^i = \begin{cases} 1 \text{ (Alarm)} & \text{if } \bar{P}^i \geq \bar{\tau}^i, \\ 0 \text{ (No alarm)} & \text{otherwise.} \end{cases} \tag{16}$$

*where* $\bar{P}^i$ *can be obtained by Gaussian approximation as follows:*

$$\bar{P}^i \approx \frac{1}{1 + \frac{\pi^i_0}{\pi^i_1}\prod_{j=1}^{n_i}\frac{\alpha^i_{j,D}+\beta^i_{j,D}}{\alpha^i_{j,F}+\beta^i_{j,F}}\left(\frac{\alpha^i_{j,F}}{\alpha^i_{j,D}}\right)^{y^i_j}\left(\frac{\beta^i_{j,F}}{\beta^i_{j,D}}\right)^{1-y^i_j}}.$$

*The corresponding Bayes risk for the optimal decision is*

$$R^i(\delta^i) = \begin{cases} C^i_{10}(1-\bar{P}^i) & \text{if } \bar{P}^i \geq \bar{\tau}^i, \\ C^i_{01}\bar{P}^i & \text{otherwise.} \end{cases} \tag{17}$$

**Proof.** The likelihood ratio can be rewritten as

$$L^i(\mathbf{y}^i) = \prod_{j=1}^{n_i}\frac{p^i_j(y^i_j|H_1)}{p^i_j(y^i_j|H_0)} \tag{18}$$

Using the Gaussian approximations of beta distributions under the assumption of large data sets, we obtain

$$L^i(\mathbf{y}^i) \approx \frac{\prod_{j=1}^{n_i}(\mathbb{E}[p^i_{j,F}])^{y^i_j}(1-\mathbb{E}[p^i_{j,F}])^{y^i_j}}{\prod_{j=1}^{n_i}(\mathbb{E}[p^i_{j,D}])^{y^i_j}(1-\mathbb{E}[p^i_{j,D}])^{y^i_j}} \tag{19}$$

The result then follows from using the threshold rule (13) and (14). □

## 5. Sequential hypothesis testing

The optimal decision rule in Section 4 requires each IDS to send requests to all the acquaintances. As the number of acquaintances increases, it creates a lot of communication overhead and consumes a large amount of computational power to implement Algorithm 1. Instead, it is desirable that IDSs can choose a sufficient number of acquaintances to guarantee a certain level of confidence in the final feedback aggregation. In this section, we use sequential hypothesis testing to make decisions with minimum number of feedbacks from the peer IDSs, [38,39]. An IDS asks for feedback from its acquaintance list until a sufficient number of answers are collected. Let $\Omega^i$ denote all the possible collections of feedback in the acquaintance list to an IDS $i$ and $\omega^i \in \Omega^i$ denotes a particular collection of feedback. Let $N^i(\omega^i)$ be a random variable denoting the number of feedbacks used until a decision is made. A sequential decision rule is formed by a pair $(\phi, \delta)$, where $\phi^i = \{\phi^i_n, n \in \mathbb{N}\}$ is a stopping rule and $\delta^i = \{\delta^i_n, n \in \mathbb{N}\}$ is the terminal decision rule. Introduce a stopping rule with $n$ feedback, $\phi^i_n : \mathcal{Y}^i_n := \prod_{j\in\mathcal{N}_{i,n}} \mathcal{Y} \to \{0, 1\}$, where $\mathcal{N}_{i,n}$ is the set of nodes an IDS $i$ asks up to time $n$. $\phi^i_n = 0$ indicates that IDS $i$ needs to take more samples after $n$ rounds whereas $\phi^i_n = 1$ means to stop asking for feedback and a decision can be made by the rule $\delta^i_n$. The minimum number of feedbacks is given by

$$N^i(\omega^i) = \min\{n : \phi^i_n = 1, n \in \mathbb{N}\}. \tag{20}$$

Note that $N^i(\omega^i)$ is the stopping time of the decision rule. The decision rule $\delta^i$ is not used until $N$. We assume that no cost has incurred when a correct decision is made while the cost of a missed intrusion is denoted by $C^i_{01}$ and the cost of a false alarm is denoted by $C^i_{10}$. In addition, we assume each feedback incurs a cost $D^i$. We introduce an optimal sequential rule that minimizes Bayes risk given by

$$R^i(\phi^i, \delta^i) = R(\phi^i, \delta^i|H_0)\pi^i_0 + R(\phi^i, \delta^i|H_1)\pi^i_1, \tag{21}$$

where $R(\phi^i, \delta^i|H_l), l = 0, 1,$ are the Bayes risks under hypotheses $H_0$ and $H_1$, respectively:

$$R^i(\phi^i, \delta^i|H_0) = C^i_{10}p^i[\delta_N(Y^i_j, j \in \mathcal{N}_{i,N}) = 1|H_0] + D^i\mathbb{E}[N|H_0],$$
$$R^i(\phi^i, \delta^i|H_1) = C^i_{01}p^i[\delta_N(Y^i_j, j \in \mathcal{N}_{i,N}) = 0|H_1] + D^i\mathbb{E}[N|H_1].$$

Let $V^i(\pi^i_0) = \min_{\phi^i,\delta^i} R^i(\phi^i, \delta^i)$ be the optimal value function. It is clear that when no feedback is obtained from the peers, the Bayes risks reduce to

$$R^i(\phi^i_0 = 1, \delta^i_0 = 1) = C^i_{10}\pi^i_0, \tag{22}$$

$$R^i(\phi^i_0 = 1, \delta^i_0 = 0) = C^i_{01}\pi^i_1. \tag{23}$$

Hence, $H_1$ is chosen when $C^i_{10}\pi^i_0 < C^i_{01}\pi^i_1$ or $\pi^i_0 < \frac{C^i_{01}}{C^i_{10}+C^i_{01}}$, and $H_0$ is chosen otherwise. The minimum Bayes risk under no feedback is thus obtained as a function of $\pi^i_0$ and is denoted by

$$T^i(\pi^i_0) = \begin{cases} C^i_{10}\pi^i_0 & \text{if } \pi_0 < \dfrac{C^i_{01}}{C^i_{10}+C^i_{01}}, \\ C^i_{01}(1-\pi^i_0) & \text{otherwise.} \end{cases} \tag{24}$$

The minimum cost function (24) is a piecewise linear function. For $\phi^i$ such that $\phi^i_0 = 0$, i.e., at least one feedback is obtained, let the minimum Bayes risk be denoted by $J^i(\pi^i_0) = \min_{\{(\phi^i,\delta^i):\phi^i_0=0\}} R^i(\phi^i, \delta^i)$. Hence, the optimal Bayes risk needs to satisfy

$$V^i(\pi^i_0) = \min\{T^i(\pi^i_0), J^i(\pi^i_0)\}. \tag{25}$$

Note that $J^i(\pi^i_0)$ must be greater than the cost of one sample $D^i$ as a sample request incurs $D^i$ and $J^i(\pi^i_0)$ is concave in $\pi^i_0$ as a consequence of minimizing the linear Bayes risk (21). If the cost $D^i$ is high enough so that $J^i(\pi^i_0) > T^i(\pi^i_0)$ for all $\pi^i_0$, then no feedback will be requested. In this case, $V^i(\pi^i_0) = T^i(\pi^i_0)$, and the terminal rule is described in (24). For other values of $D^i > 0$, due to the piecewise linearity of $T^i(\pi^i_0)$ and concavity of $J^i(\pi^i_0)$, we can see that $J^i(\pi^i_0)$ and $T^i(\pi^i_0)$ have two intersection points $\pi^i_L$ and $\pi^i_H$ such that $\pi^i_L \le \pi^i_H$. It can be shown that for some reasonably low cost $D^i$ and $\pi^i_0$ such that $\pi^i_L < \pi^i_0 < \pi^i_H$, an IDS optimizes its risk by requesting another feedback; otherwise, an IDS should choose to raise an alarm when $\pi^i_0 \le \pi^i_L$ and report no intrusion when $\pi^i_0 \le \pi^i_L$.

Assuming that it takes the same cost $D^i$ for IDS $i$ to acquire a feedback, the problem has the same form after obtaining a feedback from a peer. IDS $i$ can use the feedback to update its apriori probability. After $n$ feedbacks are obtained, $\pi^i_0$ can be updated as follows:

$$\pi^i_0(n) = \frac{\pi^i_0}{\pi^i_0 + (1-\pi^i_0)L^i_n}; \tag{26}$$

where $L^i_n := \prod_{j\in\mathcal{N}_{i,n}} \frac{p^i(y^i_j|H_1)}{p(y^i_j|H_0)}$. We can thus obtain the optimum Bayesian rule captured by Algorithm 1 below, known as the sequential probability ratio test (SPRT) for a reasonable cost $D^i$. The SPRT Algorithm 2 can be used to replace step 2 in Algorithm 1. In FACID, it is important to note that the choice between Algorithm 2 and Algorithm 1 depends on the number of acquaintances of an IDS and its computational and memory resources. For smaller scale IDS networks or new members of the CIDN, Algorithm 1 is more desirable because it allows IDSs to collect more data and learn the level of expertise and trustworthiness of their peers. However, Algorithm 2 becomes more efficient when an IDS has a large number of acquaintances and limited resources.

**Algorithm 2** SPRT rule for an IDS $i$.

**Step 1**: Start with $n = 0$. Use (27) as a stopping rule until $\phi^i_n = 1$ for some $n \ge 0$.

$$\phi^i_n = \begin{cases} 0 & \text{if } \pi^i_L < \pi^i_0(n) < \pi^i_H, \\ 1 & \text{otherwise.} \end{cases} \tag{27}$$

or in terms of the likelihood ratio $L^i_n$, we can use

$$\phi^i_n = \begin{cases} 0 & \text{if } A^i < L^i_n < B^i \\ 1 & \text{otherwise} \end{cases},$$

where $A^i = \frac{\pi^i_0(1-\pi^i_H)}{(1-\pi^i_0)\pi^i_H}$ and $B^i = \frac{\pi^i_0(1-\pi^i_L)}{(1-\pi^i_0)\pi^i_L}$.

**Step 2**: Go to Step 3 if $\phi^i_n = 1$ or $n = |\mathcal{N}_i|$; otherwise, choose a new peer from the acquaintance list to request a diagnosis and go to Step 2 with $n = n + 1$.

**Step 3**: Apply the terminal decision rule as follows to determine whether there is an intrusion or not.

$$\delta^i_n = \begin{cases} 1 & \text{if } \pi^i_0(n) \le \pi^i_L \\ 0 & \text{if } \pi^i_0(n) > \pi^i_H \end{cases} \text{ or } \delta^i_n = \begin{cases} 1 & \text{if } L^i_n \le A^i \\ 0 & \text{if } L^i_n > B^i \end{cases}$$

### 5.1. Threshold approximation

In the likelihood sequential ratio test of Algorithm 2, the threshold values $A$ and $B$ need to be calculated by finding $\pi^i_L$ and $\pi^i_H$ from $J^i(\pi^i_0)$ and $T^i(\pi^i_0)$ in (25). The search for these values can be quite involved using dynamic programming. However, in this subsection, we introduce an approximation method to find the thresholds. The approximation is based on theoretical studies made in [38] and [39] where a random walk or martingale model is used to yield a relation between thresholds and false positive

and false negative rates. Let $P_D^i, P_F^i$ be the probability of detection and the probability of false alarm of an IDS $i$ after applying the sequential hypothesis testing for feedback aggregation. We need to point out that these probabilities are different from the probabilities $p_D^i, p_F^i$ discussed in the previous subsection, which are the raw detection probabilities without feedback in the collaboration network. Let $\bar{P}_D^i$ and $\bar{P}_F^i$ be the desired performance bounds such that $P_F^i \leq \bar{P}_F^i$, $P_D^i \geq \bar{P}_D^i$. Then, the thresholds can be chosen such that $A^i = \frac{1-\bar{P}_D^i}{1-\bar{P}_F^i}$, $B^i = \frac{\bar{P}_D^i}{\bar{P}_F^i}$.

The next proposition gives a result on the bound of the peers that need to be on the acquaintance list to achieve the desired performances.

**Proposition 5.1.** *Assume that each IDS makes independent diagnosis on its peers' requests and each has the same distribution* $p_0^i = \bar{p}_0 := \bar{p}(\cdot|H_0)$, $p_1^i = \bar{p}_1 := \bar{p}(\cdot|H_1)$, $\bar{p}_0(y_i = 0) = \theta_0$, $\bar{p}_1(y_i = 0) = \theta_1$, *for all* $i \in \mathcal{N}$.

*Let* $D_{KL}(\bar{p}_0||\bar{p}_1)$ *be the Kullback–Leibler (KL) divergence defined as follows.*

$$D_{KL}(\bar{p}_0||\bar{p}_1) = \sum_{k=0}^{1} \bar{p}_0(k) \ln \frac{\bar{p}_0(k)}{\bar{p}_1(k)}, \tag{28}$$

$$= \theta_0 \ln \frac{\theta_0}{\theta_1} + (1-\theta_0) \ln \frac{1-\theta_0}{1-\theta_1} \tag{29}$$

*and likewise introduce the K–L divergence* $D_{KL}(\bar{p}_1||\bar{p}_0)$. *Then on the average, an IDS needs* $N_i$ *acquaintances such that*

$$N_i \geq \max\left( \left\lceil -\frac{D_M^i}{D_{KL}(\bar{p}_0||\bar{p}_1)} \right\rceil, \left\lceil \frac{D_F^i}{D_{KL}(\bar{p}_1||\bar{p}_0)} \right\rceil \right), \tag{30}$$

*where* $D_M^i = P_F \ln(\frac{P_D^i}{P_F^i}) + P_D \ln(\frac{1-P_D^i}{1-P_F^i})$ *and* $D_F^i = P_F^i \ln(\frac{1-P_D^i}{1-P_F^i}) + P_D^i \ln(\frac{P_D^i}{P_F^i})$. *If* $P_F^i \ll 1$ *and* $P_M^i \ll 1$, *we need approximately* $N_i$ *acquaintances such that*

$$N_i \geq \max\left( \left\lceil \frac{P_D^i - 1}{D_{KL}(\bar{p}_0||\bar{p}_1)} \right\rceil, \left\lceil -\frac{P_F^i}{D_{KL}(\bar{p}_1||\bar{p}_0)} \right\rceil \right). \tag{31}$$

**Proof.** The conditional expected number of feedbacks needed to reach a decision on the hypothesis in SPRT can be expressed in terms of $P_F$ and $P_D$, [38,39].

$$\mathbb{E}[N|H_0] = \frac{1}{-D_{KL}(\bar{p}_0||\bar{p}_1)} \left[ P_F^i \ln\left(\frac{P_D^i}{P_F^i}\right) + P_D^i \ln\left(\frac{1-P_D^i}{1-P_F^i}\right) \right],$$

$$\mathbb{E}[N|H_1] = \frac{1}{D_{KL}(\bar{p}_1||\bar{p}_0)} \left[ P_F^i \ln\left(\frac{1-P_D^i}{1-P_F^i}\right) + P_D^i \ln\left(\frac{P_D^i}{P_F^i}\right) \right],$$

Hence, to reach a decision we need to have at least $\max\{\mathbb{E}[N|H_0], \mathbb{E}[N|H_1]\}$ independent acquaintances. Under the assumption that both $P_F$ and $P_M^i$ are much less than 1, we can further approximate

$$\mathbb{E}[N|H_0] \sim -\frac{1-P_D^i}{D_{KL}(\bar{p}_0||\bar{p}_1)}, \mathbb{E}[N|H_1] \sim -\frac{P_F^i}{D_{KL}(\bar{p}_1||\bar{p}_0)}.$$

This lead us to inequalities (31) and (30). □

## 6. Experiments and results

In this section, we use a simulation approach to evaluate the efficiency of the FACID feedback aggregation scheme and compare it with other heuristic approaches, such as the simple average aggregation and the weighted average aggregation (described later in this section).

**Table 2**
Experimental parameters.

| Parameter | Value | Meaning |
|---|---|---|
| $\tau_{SA}$ | 0.5 | Decision threshold of the simple average model |
| $\tau_{WA}$ | 0.5 | Decision threshold of the weighted average model |
| $n$ | 10 | Number of IDSs in the network |
| $d$ | 0.5 | Difficulty level of intrusions and test messages |
| $\lambda$ | 0.9 | Forgetting factor |
| $\pi_0, \pi_1$ | 0.5 | Probability of no-attack and under-attack |
| $C_{00}, C_{11}$ | 0 | Cost of correct decisions |

Specifically, we present a set of experiments conducted to evaluate the average cost of collaborative detection using the FACID aggregation model in comparison with the simple average and the weighted average models. We validate and confirm our theoretical results on the number of acquaintances needed for consultation. Each experimental result presented in this section is derived from the average of a large number of replications with an overall negligible confidence interval. The parameters we use are shown in Table 2.

### 6.1. Simulation setting

The simulation environment uses a CIDN of $n$ peers. Each IDS is represented by two parameters, expertise level $l$ and decision threshold $\tau_p$. Expertise level $l$ represents the ability of the IDS to catch suspicious traces from a given observation, and $\tau_p$ represents the sensitivity of the IDS. Lower value of $\tau_p$ represents a more sensitive IDS (to be elaborated more in the Section 6.2). 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. In our simulation setup, we assumed that the truth is revealed after the diagnosis is made. For data points that are not revealed, we can exclude them from the data set of evaluation. 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 FACID aggregation. We compare their efficiency based on the average cost of false decisions. The reason that we choose to compare with simple average and weighted average model is that they have been adopted in existing CIDN models and they can be suitable benchmark methods.

#### 6.1.1. Simple average model

If the average of all feedbacks is larger than a threshold, then raise an alarm. This model has been used in ABDIAS [20] where a simple voting is used to detect intruding nodes.

$$\delta_{SA} = \begin{cases} 1 \text{ (Alarm)} & \text{if} \frac{\sum_{k=1}^{n} \mathbf{y}_k}{n} \geq \tau_{SA}, \\ 0 \text{ (No alarm)} & \text{otherwise}, \end{cases} \tag{32}$$

where $\tau_{SA}$ 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.

#### 6.1.2. Weighed average model

Weights are assigned to feedbacks from different acquaintances to distinguish their detection capability. For example, high expertise IDSs are assigned with larger weights compared to low exper-

tise IDSs. In [6,7,21], the weights are the trust values of IDSs:

$$\delta_{WA} = \begin{cases} 1 \text{ (Alarm)} & \text{if} \frac{\sum_{k=1}^{n} w_k \mathbf{y}_k}{\sum_{k=1}^{n} w_k} \geq \tau_{WA}, \\ 0 \text{ (No alarm)} & \text{otherwise}, \end{cases} \quad (33)$$

where $w_k$ is the weight of the feedback from acquaintance $k$, which is the trust value of acquaintance $k$ in [6,7,21]. $\tau_{WA}$ is the decision threshold for the weighted average algorithm. It is fixed to 0.5 since no cost is considered for false positives (FP) and false negatives (FN). In this simulation, we adopt trust values from [7] to be the weights of feedbacks.

#### 6.1.3. FACID aggregation model

As described in Section 4.2, the FACID 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. The FACID decision model investigates the cost of all possible decisions and chooses the one which leads to the minimal expected cost.

### 6.2. Simulation 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:

$$f(\bar{p}|\bar{\alpha}, \bar{\beta}) = \frac{1}{B(\bar{\alpha}, \bar{\beta})} \bar{p}^{\bar{\alpha}-1}(1-\bar{p})^{\bar{\beta}-1};$$

$$B(\bar{\alpha}, \bar{\beta}) = \int_0^1 t^{\bar{\alpha}-1}(1-t)^{\bar{\beta}-1} dt, \quad (34)$$

$\bar{\alpha}$ and $\bar{\beta}$ are defined as follows.

$$\bar{\alpha} = 1 + \frac{l(1-d)}{d(1-l)} r,$$

$$\bar{\beta} = 1 + \frac{l(1-d)}{d(1-l)} (1-r). \quad (35)$$

where $\bar{p} \in [0, 1]$ is the assessment result from the host IDS about the likelihood of intrusion, and $f(\bar{p}|\bar{\alpha}, \bar{\beta})$ is the distribution of assessment $\bar{p}$ from a peer with expertise level $l$ to an intrusion with 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. $r \in \{0, 1\}$ is the expected result of detection. $r = 1$ indicates that there is an intrusion and $r = 0$ indicates that there is no intrusion. For a fixed difficulty level, the above model has the property of assigning higher probabilities of producing correct intrusion decisions to peers with higher levels of expertise. A peer with expertise level $l$ has a lower probability of producing correct intrusion decisions for alerts of higher difficulty ($d > l$). $l = 1$ or $d = 0$ represent the extreme cases where the peer can always accurately rank the alert. This is reflected in the Beta distribution by $\alpha, \beta \to \infty$. Fig. 4 shows the feedback probability distribution function (PDF) for peers with different expertise levels, where we fix $r = 1$ and the difficulty level of test messages to 0.5. The x-axis is the perceived likelihood $\bar{p} \in [0, 1]$ of a real intrusion from an IDS. We can see that, IDSs with higher expertise levels are more likely to correctly assess an intrusion incident to be harmful.

$\tau_p$ 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. $\tau_p$ indicates the sensitivity of an IDS detector, where lower $\tau$ value implies a more sensitive detector. i.e., the IDS is more likely to raise alert when suspicious activities are noticed.

Fig. 5 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 $r = 1$. Fig. 6 shows that the FP decreases with the decision threshold increase while the FN increases with the decision threshold increase. When the decision threshold is 0, all feedbacks are positive; when the decision threshold is 1, all feedbacks are negative.

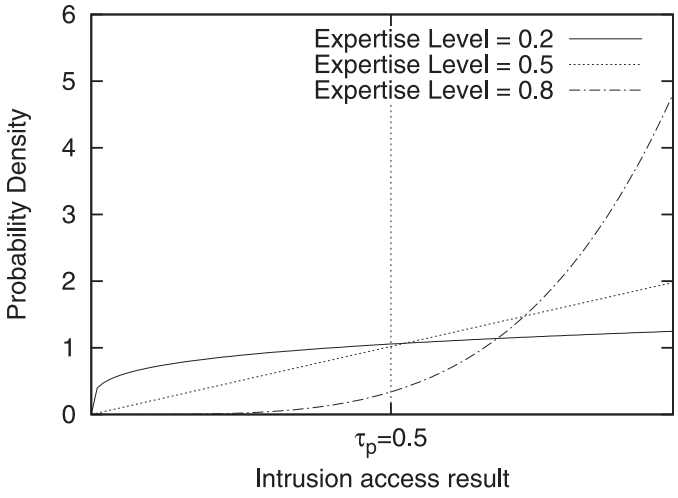



**Fig. 4.** Expertise level and detection rate.

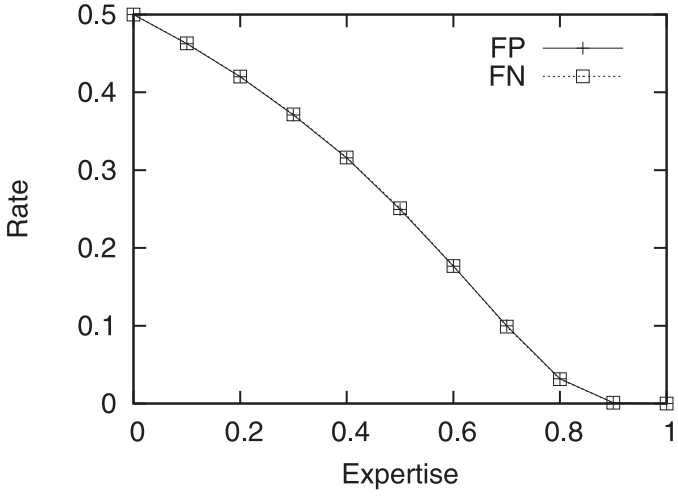



**Fig. 5.** FP and FN vs. expertise level $l$.

### 6.3. 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 FACID 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.

#### 6.3.1. 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 $C_{01} = C_{10} = 1$ for the fairness of comparison, since the simple average and the weighted average models do not account for the cost difference

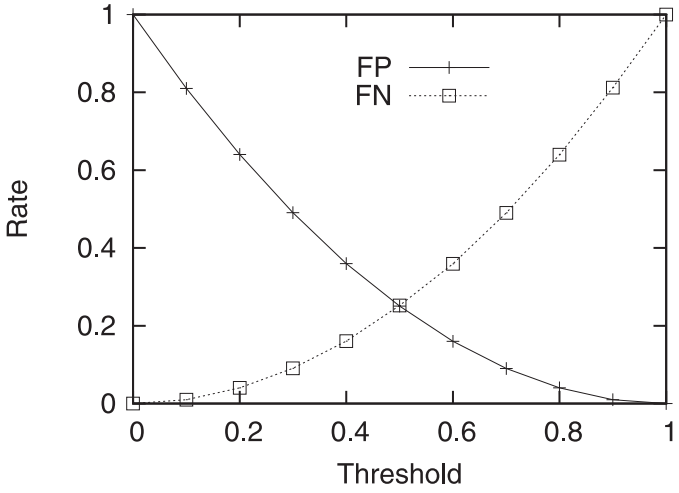



**Fig. 6.** FP and FN vs. threshold $\tau_p$.

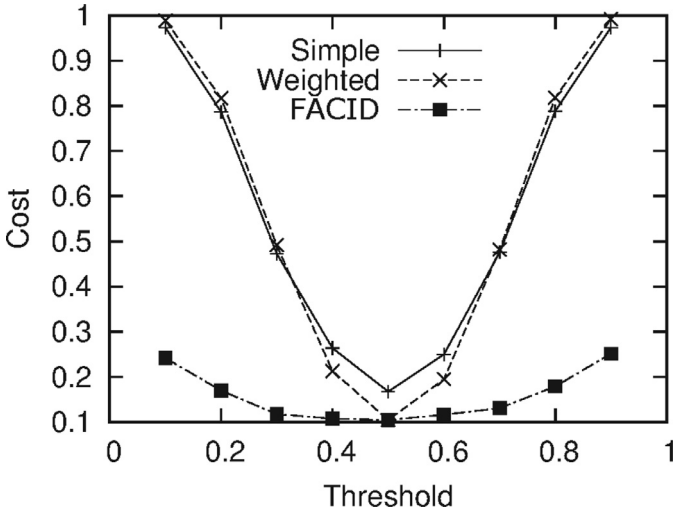



**Fig. 7.** Average cost vs. threshold $\tau_p$.

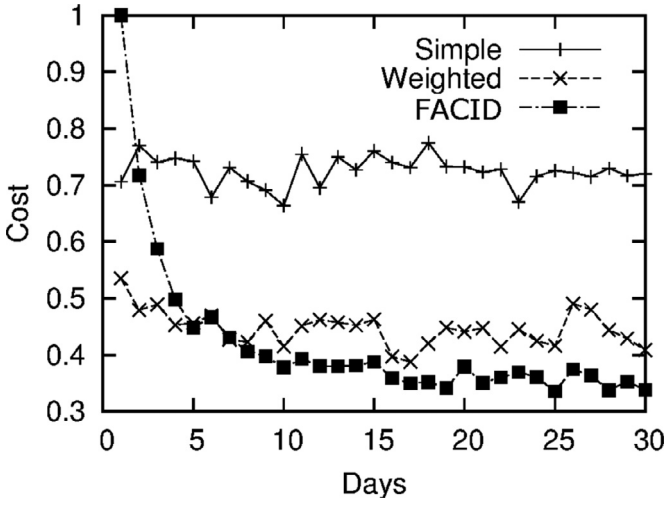



**Fig. 8.** Average costs for three different aggregation models.

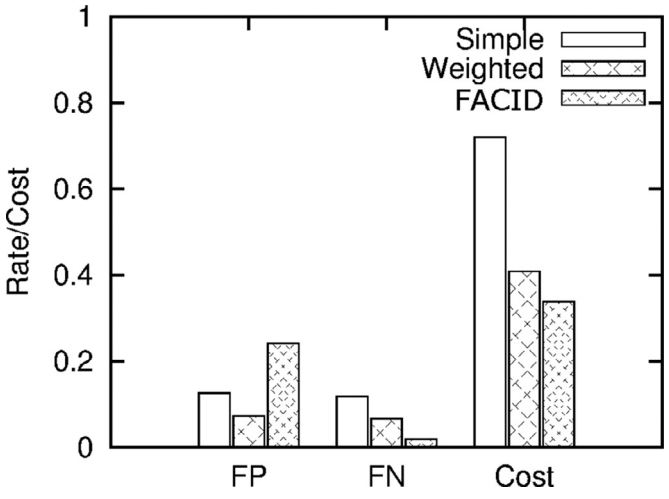



**Fig. 9.** Comparison of three aggregation models.

between FP and FN. We fix the decision threshold for each IDS ($\tau_p$) 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 Fig. 7, the average costs on false decisions yielded by FACID remain 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 in the FACID 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.

*6.3.2. 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 false positive cost $C_{10} = 1$ and false negative cost $C_{01} = 5$ 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.

Fig. 8 shows that the average costs of the three different models converge after a few days of learning process. The cost of FACID 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 Fig. 9 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 FACID 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 FACID model trades some FP with FN to reduce the overall cost of false decisions.

*6.3.3. 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 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 $\tau_p = 0.5$ in all cases. We observe in Fig. 10 that, under all cases, the average cost decreases when more acquaintances are consulted. Suppose a cost goal $U_g$ is set, then we 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 average to reach a cost of $U_g = 0.1$, under the case where all acquaintances are with a high expertise level of 0.8. Correspond-

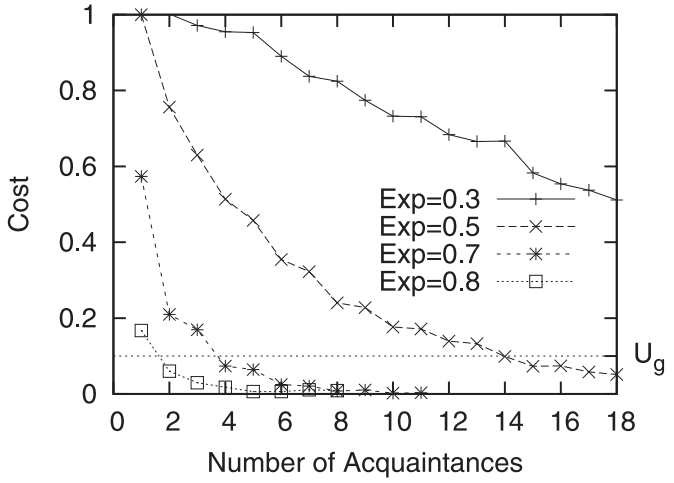



**Fig. 10.** Average cost vs. number of acquaintances consulted ($U_g$ is the cost goal).

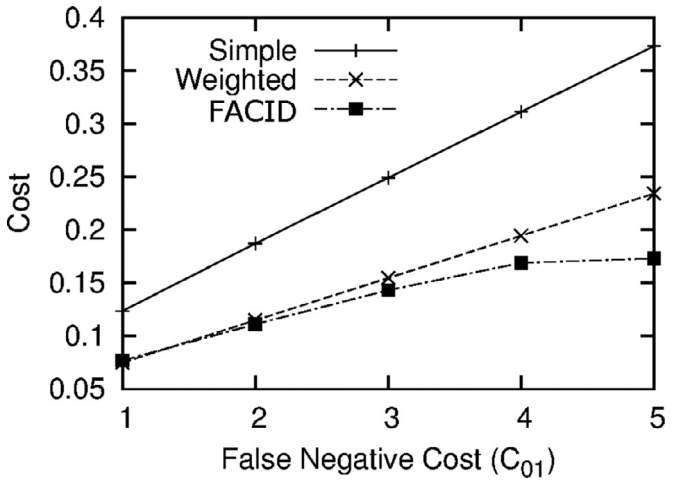



**Fig. 11.** Cost vs. $C_{01}$ under three models.

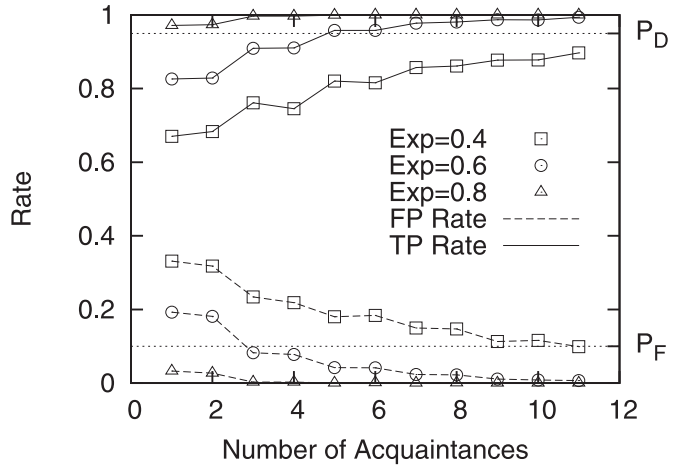



**Fig. 12.** FP, TP vs. number of acquaintances.

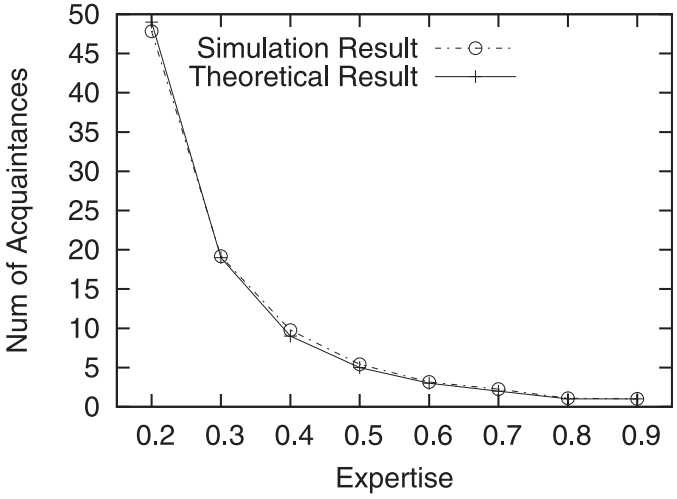



**Fig. 13.** Number of acquaintances vs. expertise.

ingly, the number of acquaintances needed are 4 and 15 on average when the acquaintance expertise levels are 0.7 and 0.5 respectively. In the case where 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.

In the next experiment, the expertise levels of all nodes remain 0.5 and their decision thresholds vary from 0.1 to 0.9. We set $C_{10} = C_{01} = 1$ in the first batch run and increase $C_{01}$ by 1 in every subsequent batch run. We observe the costs under three different models. Fig. 11 shows that the costs of the simple average model and the weighted average model increase linearly with $C_{01}$ while the cost of hypothesis testing model grows the slowest among the three. This is because the hypothesis testing model has a flexible threshold tuned to optimize its cost. The hypothesis testing model has superior performance when the cost difference between FP and FN is large.

### 6.4. Sequential consultation

In this experiment, we study the number of acquaintances needed for consultation to reach a predefined goal. Suppose the TP lower-bound is $\bar{P}_D = 0.95$ and the FP upper-bound is $\bar{P}_F = 0.1$. We observe the change of FP rate and TP rate with the number of acquaintances consulted ($n$). Fig. 12 shows that FP rate decreases and TP rate increases with $n$. Consulting higher expertise nodes leads to a higher TP rate and a lower FP rate. In the next experiment we implement Algorithm 1 on each node and measure the average number of acquaintances needed to reach the predefined TP lower-bound and the FP upper-bound. Fig. 13 compares the simulation results with the theoretical results (see Equation (31)), where the former confirms the latter. In both cases, the number of consultations decreases quickly with the expertise levels of acquaintances. For example, the IDS needs to consult around 50 acquaintances of expertise 0.2, while only 3 acquaintances of expertise 0.7 are needed to achieve the same goal. This is partly because low expertise nodes are more likely to provide conflicting feedbacks and consequently increase the number of required consultations. The analytical results are useful for determining the size of an IDSs acquaintance list.

### 6.5. Robustness and scalability of the system

Robustness and scalability are two important features of a CIDN. FACID is robust to malicious insiders since it has an inherent robust trust management model where malicious insiders can be quickly discovered and removed from the acquaintance list. To verify this, we simulate the scenario of betrayal attack under a homogeneous environment. We fix all 10 IDSs with $l = 0.5$ and $\tau_p = 0.5$. We let one IDS turn malicious at day 20 and start to give opposite diagnosis. We observe the FP and TP rate of the malicious node and its impact on the decision of other nodes. From Fig. 14 we can see that the FP rate and TP rate of the malicious node raise/drop

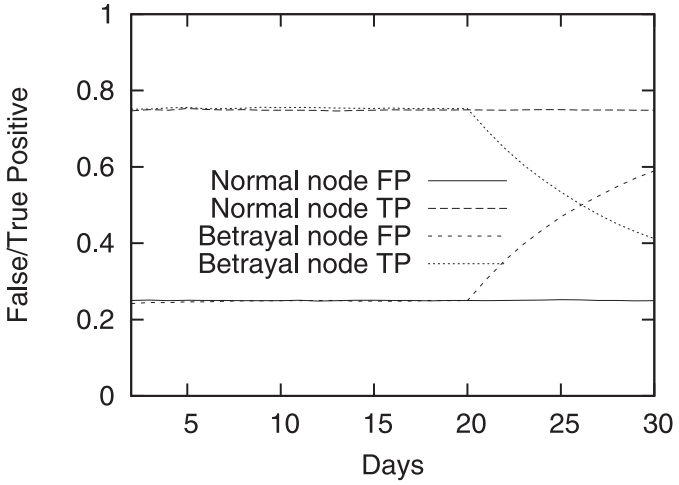



**Fig. 14.** False positive and true positive of single IDS under betrayal attack.

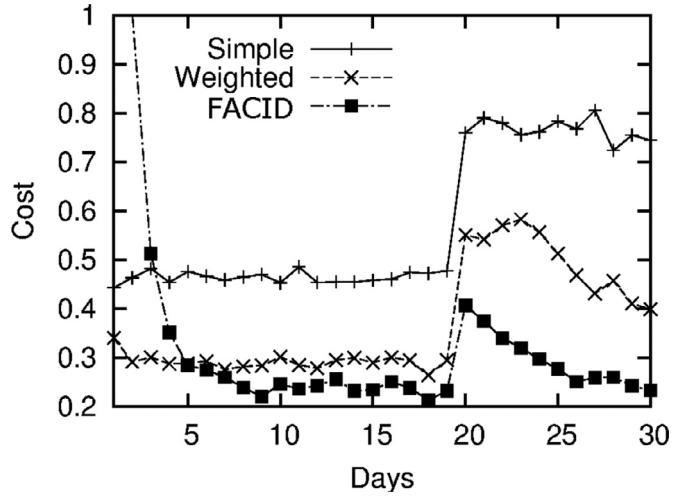



**Fig. 15.** False decision cost under betrayal attack.

quickly after day 20. Fig. 15 shows that the cost of false decisions of the other nodes raises quickly at day 20 and drops back to normal after a few days. Compared with the other two aggregation models, the FACID model results in the least impact from the malicious node.

The CIDN scalability is also ensured 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 reduces the number of consultation messages needed for collaborative intrusions detection.

## 7. Conclusion

In this paper, we have described an architecture for a collaborative intrusion detection network and discussed its system components. The feedback aggregation is a critical component for achieving an effective collaboration mechanism, and we have proposed FACID, a framework for trustworthy feedback aggregation. We have obtained optimal decision rules that minimize Bayes risks using hypothesis testing methods, and provided a data-driven mechanism for real-time, efficient, distributed, and sequential feedback aggregation. In this mechanism, an IDS consults sequentially for peer diagnoses until it is capable of making an aggregated decision that meets Bayes optimality. The decision is made based on a threshold rule leveraging the likelihood ratio approximated by beta distribution and thresholds. Our experimental results have shown that FACID is superior to other proposed models in the literature in terms of cost efficiency. Our simulation results have also corroborated our theoretical results on the average number of acquaintances needed to reach the predefined false positive upper-bound and true positive lower-bound. As future work, we intend to investigate large-scale collaboration networks and their topological impact. We also intend to integrate FACID system with communication networks, and design defense mechanisms against different cyber attacks such as denial of service, man-in-the-middle and insider attacks. Furthermore, we plan to extend our results to deal with the case of correlated feedbacks.

## Acknowledgement

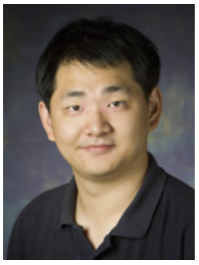
This work is partially supported by the grant CNS-1544782 from National Science Foundation (NSF).

## References

[1] R. Vogt, J. Aycock, M. Jacobson, Army of botnets, ISOC Symp. on Network and Distributed Systems Security, 2007.
[2] J. Mirkovic, P. Reiher, A taxonomy of ddos attack and ddos defense mechanisms, SIGCOMM Comput. Commun. Rev. 34 (2) (2004) 39–53. http://doi.acm.org/10.1145/997150.997156.
[3] K.C. Wilbur, Y. Zhu, Click fraud, (2009).
[4] C. Fung, D. Lam, R. Boutaba, RevMatch: an efficient and robust decision model for collaborative malware detection, IEEE/IFIP Network Operation and Management Symposium (NOMS14), 2014.
[5] P. Resnick, R. Zeckhauser, J. Swanson, K. Lockwood, The value of reputation on eBay: a controlled experiment, Exp. Econ. 9 (2) (2006) 79–101.
[6] C. Duma, M. Karresand, N. Shahmehri, G. Caronni, A trust-aware, p2p-based overlay for intrusion detection, DEXA Workshops, 2006.
[7] C. Fung, J. Zhang, I. Aib, R. Boutaba, Robust and scalable trust management for collaborative intrusion detection, 11th IFIP/IEEE International Symposium on Integrated Network Management, 2009.
[8] R. Janakiraman, M. Zhang, Indra: a peer-to-peer approach to network intrusion detection and prevention, in: Proc. of the 12th IEEE International Workshops on Enabling Technologies, 2003.
[9] V. Yegneswaran, P. Barford, S. Jha, Global intrusion detection in the domino overlay system, in: Proc. of Network and Distributed System Security Symposium, 2004.
[10] M.E. Locasto, J.J. Parekh, A.D. Keromytis, S.J. Stolfo, Towards collaborative security and p2p intrusion detection, in: Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC, IEEE, 2005, pp. 333–339.
[11] Z. Czirkos, G. Hosszú, Enhancing collaborative intrusion detection methods using a kademlia overlay network, in: Information and Communication Technologies, Springer, 2012, pp. 52–63.
[12] M. Cai, K. Hwang, Y. Kwok, S. Song, Y. Chen, Collaborative internet worm containment, IEEE Secur. Privacy 3 (3) (2005) 25–33.
[13] C.G. Cordero, E. Vasilomanolakis, M. Mühlhäuser, M. Fischer, Community-based collaborative intrusion detection, in: International Conference on Security and Privacy in Communication Systems, Springer, 2015, pp. 665–681.
[14] E. Vasilomanolakis, M. Fischer, M. Mühlhäuser, P. Ebinger, P. Kikiras, S. Schmerl, Collaborative intrusion detection in smart energy grids, in: Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research 2013, BCS, 2013, pp. 97–100.
[15] X. Liu, P. Zhu, Y. Zhang, K. Chen, A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure, IEEE Trans. Smart Grid 6 (5) (2015) 2435–2443.
[16] S. Al-Janabi, A. Patel, J.C. Junior, J.M. Pedersen, A.L.M. dos Santos, A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems, Comput. Secur. (2016).
[17] N.D. Man, E.-N. Huh, A collaborative intrusion detection system framework for cloud computing, in: Proceedings of the International Conference on IT Convergence and Security 2011, Springer, 2012, pp. 91–109.
[18] N.-F. Huang, C. Wang, I.-J. Liao, C.-W. Lin, C.-N. Kao, An openflow-based collaborative intrusion prevention system for cloud networking, in: Communication Software and Networks (ICCSN), 2015 IEEE International Conference on, IEEE, 2015, pp. 85–92.
[19] H. Sedjelmaci, S.M. Senouci, An accurate and efficient collaborative intrusion detection framework to secure vehicular networks, Comput. Electr. Eng. 43 (2015) 33–47.
[20] A. Ghosh, S. Sen, Agent-based distributed intrusion alert system, in: Proc. of the 6th International Workshop on Distributed Computing (IWDC'04), Springer, 2004.

[21] C. Fung, O. Baysal, J. Zhang, I. Aib, R. Boutaba, Trust management for host-based collaborative intrusion detection, 19th IFIP/IEEE International Workshop on Distributed Systems, 2008.
[22] M.G. Pérez, F.G. Mármol, G.M. Pérez, A.F.S. Gómez, Repcidn: A reputation-based collaborative intrusion detection network to lessen the impact of malicious alarms, J. Netw. Syst. Manage. 21 (1) (2013) 128–167.
[23] S. Pastrana, J.E. Tapiador, A. Orfila, P. Peris-Lopez, Defidnet: A framework for optimal allocation of cyberdefenses in intrusion detection networks, Comput. Netw. 80 (2015) 66–88.
[24] J. Tsitsiklis, Decentralized detection, Adv. Stat. Signal Process. (1993) 297–344.
[25] K. Nguyen, T. Alpcan, T. Başar, A decentralized Bayesian attack detection algorithm for network security, in: Proceedings of the 23rd International Information Security Conference, 2005.
[26] C. Fung, Q. Zhu, R. Boutaba, T. Başar, Bayesian decision aggregation in collaborative intrusion detection networks, in: Proc. of 2010 IEEE Network Operations and Management Symposium (NOMS), 2010, pp. 349–356.
[27] Q. Zhu, C. Fung, R. Boutaba, T. Başar, A game-theoretical approach to incentive design in collaborative intrusion detection networks, in: Proc. of the International Symposium on Game Theory for Networks (GameNets), May, 2009, pp. 384–392.
[28] Q. Zhu, C. Fung, R. Boutaba, T. Başar, GUIDEX: a game-theoretic incentive-based mechanism for intrusion detection networks, IEEE J. Sel. Areas Commun. (JSAC) Special Issue Econ. Commun. Networks Syst. 30 (11) (2012) 2220–2230.
[29] Q. Zhu, C. Fung, R. Boutaba, T. Başar, A game-theoretic approach to knowledge sharing in distributed collaborative intrusion detection networks: fairness, incentives and security, in: Proc. of the 50th IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC), Orlando, USA, 2011.
[30] C. Fung, Q. Zhu, R. Boutabai, T. Başar, Poster: SMURFEN: a rule sharing collaborative intrusion detection network, in: Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS), 2011, pp. 761–764.
[31] F. Cohen, Defense-in-depth against computer viruses, Comput. Security 11 (6) (1992) 563–579.
[32] T. Bass, R. Robichaux, Defense-in-depth revisited: qualitative risk analysis methodology for complex network-centric operations, in: Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE, Vol. 1, IEEE, 2001, pp. 64–70.
[33] VirusTotal, https://www.virustotal.com/.
[34] W. Lee, S.J. Stolfo, K.W. Mok, A data mining framework for building intrusion detection models, in: Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on, IEEE, 1999, pp. 120–132.
[35] A. Sperotto, A. Pras, Flow-based intrusion detection, in: 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops, IEEE, 2011, pp. 958–963.
[36] C. Fung, J. Zhang, R. Boutaba, Dirichlet-based trust management for effective collaborative intrusion detection networks, IEEE Trans. Netw. Serv. Manage. (TNSM) 8 (2) (2011) 79–91.
[37] C. Fung, J. Zhang, R. Boutaba, Effective acquaintance management for collaborative intrusion detection networks, in: Proc. of 16th International Conference on Network and Service Management (CNSM 2010), 2010.
[38] A. Wald, Sequential Analysis, John Wiley and Sons, 1947.
[39] B.C. Levy, Principles of Signal Detection and Parameter Estimation, Springer-Verlag, 2008.

**Carol Fung** received her Bachelor degree and Master degree in computer science from the university of Manitoba (Canada), and her Ph.D. degree in computer science from the university of Waterloo (Canada). Her research interests include collaborative intrusion detection networks, social networks, security issues in mobile networks and medical systems, Security issues in next generation networking, and machine learning in intrusion detection. She is the recipient of the IEEE/IFIP IM 2015 Young Professional Award, Alumni Gold Medal of university of Waterloo in 2013, best dissertation awards in IM2013, the best student paper award in CNSM2011 and the best paper award in IM2009. She received numerous prestige awards and scholarships including Google Anita Borg scholarship, NSERC Postdoc fellowship, David Cheriton Scholarship, NSERC Postgraduate Scholarship, and President's graduate scholarship. She has been a visiting scholar at POSTECH (South Korea), a software engineer intern at Google, and a research intern at BlackBerry.



**Quanyan Zhu** received B. Eng. in Honors Electrical Engineering with distinction from McGill University in 2006, M.A.Sc. from University of Toronto in 2008, and Ph.D. from the University of Illinois at Urbana- Champaign (UIUC) in 2013. After stints at Princeton University, he is currently an assistant professor at the Department of Electrical and Computer Engineering, New York University. He spearheaded and chaired INFOCOM Workshop on Communications and Control on Smart Energy Systems (CCSES), Midwest Workshop on Control and Game Theory (WCGT), and 7th Game and Decision Theory for Cyber Security (GameSec). His current research interests include resilient and secure cyberphysical systems, adversarial signal processing, and interdependent networks. He is a recipient of best paper awards at 5th International Conference on Resilient Control Systems, 18th International Conference on Information Fusion, and 7th ACM CCS International Workshop on Managing Insider Security Threats (MIST).