A Fuzzy Model for the Composition of Intrusion Detectors

Inez Raguenet and Carlos Maziero

Abstract The performance of an intrusion detector depends on several factors, like its internal architecture and the algorithms it uses. Thus, distinct detectors can behave distinctly when submitted to the same inputs. The project diversity theory has been successfully used in the fault tolerance domain, and can also bring benefits to the intrusion detection area. The objective of this paper is to propose and evaluate a mathematical model, based on the fuzzy set theory, for the composition of heterogeneous intrusion detectors analyzing the same event flow. This model intends to combine the individual detectors' results into a more accurate global result. Experimental results show the usefulness of this approach.

1 Introduction

In most facilities where it is necessary to detect unauthorized access to sensitive resources and data, usually only one intrusion detection system (IDS) is deployed, for practical reasons. In some cases, there is more than one IDS program working collaboratively; when it happens, IDSs are generally located in distinct strategic places in the system, to detect and to analyze distinct events.

It can be difficult to deploy several IDS programs in the same installation, due to difficulties in tuning the software and consolidating data from different sources. On the other hand, replicating several copies of the same IDS program can lead to biased results, because the chosen IDS may replicate the same errors, warning about events that are not attacks (false positives) or ignoring attacks (false negatives).

The concept of project diversity [1] has proven to be very helpful in the fault tolerance and security areas [13]; it can also be applied to intrusion detection. In

Inez Raguenet
PPGIa PUCPR, Curitiba – Brazil, e-mail: raguenet@ppgia.pucpr.br

Carlos Maziero
PPGIa PUCPR, Curitiba – Brazil, e-mail: maziero@ppgia.pucpr.br

[14] it was shown that the detection capacity of a specific IDS software is related to several factors, which include its internal architecture and algorithms. So, distinct detectors can perform distinctly when submitted to the same event flow. By applying project diversity to intrusion detection, we can obtain a composite IDS, based on individual detectors that can have distinct behaviors, not replicating the same errors, and with complementary results. This can potentially lead to better detection results.

This work introduces a mathematical model based on the Set Theory that leverages the results of individual heterogeneous intrusion detection programs, allowing us to build a composite intrusion detection system (CIDS) based on project diversity. This model is capable of mapping and evaluating the results of each individual IDS, and consolidating them in a final result which is more accurate and reliable than the individual results.

This article is divided in 6 sections: Section 2 introduces the CIDS concept; Section 3 presents some basic definitions and develops simple models, based on the traditional Set Theory; in Sect. 4 the model is extended through the use of the Fuzzy Set Theory and the *alarm relevance* concept is introduced; Section 5 shows some experiments that validate the proposition; Section 6 presents some possible extensions to the model; Section 7 discusses some related work; finally, Sect. 8 concludes the paper and discusses possibilities for future research.

2 Composition of Intrusion Detection Systems

Traditionally, the use of an IDS composition aims to cover a large distributed system whose size surpasses the capacity of any individual detectors. This approach, called *Distributed IDS*, consists of deploying detectors in distinct regions of the system, whose responsibility is to capture and analyze events in that part of the system. As the number of alarms each detector can generate may be huge, several techniques for managing and using such data were introduced, like standard alarm representations [3], centralized configuration [10], and alarm data correlation [4, 5, 6].

Another possibility of IDS composition is demonstrated by [2], who introduces the *Collaborative Intrusion Detection System* concept. It aggregates three different levels of detectors (network, kernel and application) and a level of components that help consolidating individual results. The aggregation of complementary detectors is also discussed in [7] and [9].

Recent studies have shown that distinct detectors can have distinct detection capabilities, which are sometimes complementary. For instance, [14] shows that anomaly-based detection algorithms are bound to "blind spots", and proposes to combine IDS programs based on algorithm diversity. The precision of IDS results is also discussed in [15], involving, among other things, the detection capacity, the probability of raising false alarms versus the probability of detecting a real attack, the strength against attacks to the IDS host itself, the scalability, and the capacity of detecting new attacks.

In the following sections, a mathematical model that combines the results of N distinct IDS programs treating the same event flow is proposed. It is important to stress that each detector is considered as a black box, so its internal architecture and internal algorithms are considered irrelevant for the composition. In this generic model, the input data can be packets from a network, system calls in an operating system, log files, and so on. A knowledge base stores rules that define known attacks (for a signature-based IDS) or the parameters that define a normal behavior (for abnormal behavior-based detectors). The detector applies the rules from the knowledge base in the input data and raises alarms when necessary.

3 A simple composition model

The mathematical approach used in this work will be based on the traditional set theory at first. Our intention is to point out each subset of entities occurring when an IDS is at work. A similar model was proposed by [11], but this one was restricted to a single IDS; our model considers N detectors.

3.1 Some definitions

Before introducing the model, some concepts should be defined, like events and attacks:

- *Event* (e_k): an action that occurs inside a host, between two hosts, or between a host and an user, which can be captured by an intrusion detector (such as packets in a network, system calls, and log files entries).
- *Attack* (a_k) : any event aiming at exploiting a vulnerability in a given system.
- *Normal event* (n_k): any event that is not an attack.
- *Intrusion detector* (d_i): a black-box capable of classifying input events as normal events or attacks[1].
- *Composite IDS* (*CIDS*): a group of N intrusion detectors $d_1, d_2, \ldots, d_N$ analyzing the same input event flow.

As our model is based on the set theory, some sets will also be defined, and shown in Fig. 1:

- *Universe set* ($\mathbb{U}$): comprises all possible events in any system, i.e. any access or operation in a computer-based system.
- *Normal events* ($\mathbb{N}$): comprises all events expected by a system, to which it is prepared to respond (i.e. they are in the system specification).

[1] We only consider stateless intrusion detection, in which individual input events are classified either as attacks or normal events. Stateful intrusion detection, in which specific sequences of input events can constitute an attack, is not considered here.

- *Events targeted to the system* ($\mathbb{T}$): comprises all events targeted to a system, including normal events and attacks. We assume that $\mathbb{N} \cap \mathbb{T} \neq \emptyset$, as this does not change the results and leads to a richer analysis.
- *Attacks* ($\mathbb{A}_i$): comprises all events classified by an intrusion detector d_i as attacks.

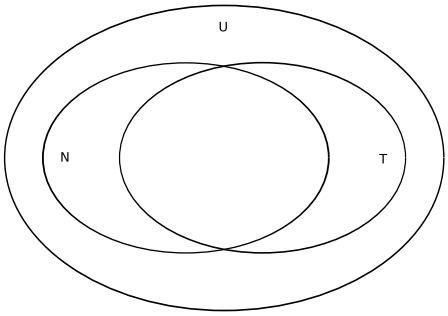

Fig. 1 The $\mathbb{U}$, $\mathbb{N}$, and $\mathbb{T}$ sets.

3.2 Modeling a single IDS

The first model considers a system with only a single detector d_1, which can detect the attacks defined in the $\mathbb{A}_1 \subset \mathbb{T}$ event set. As d_1 is not perfect, some events it classifies as attacks may be normal events (i.e. false positive detections) and some attacks may be misclassified as normal events (i.e. false negative detections). This behavior is represented in the Venn diagram of Fig. 2.

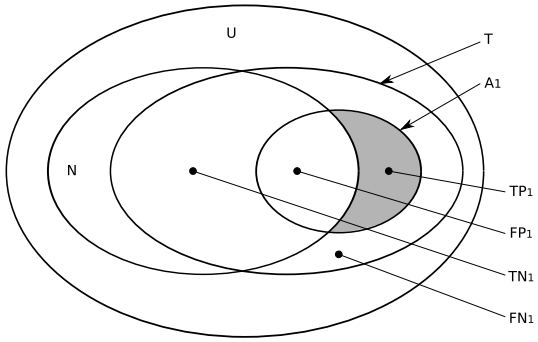

Fig. 2 Venn diagram for a single detector.

On the diagram shown in Fig. 2 it is possible to identify four subsets of interest:

- True positive alerts: $TP_1 = \mathbb{A}_1 - \mathbb{N}$ contains all attacks correctly detected by d_1 ; this area, in gray in Fig. 2, corresponds to the correct behavior expected from d_1 ;

- False positives alerts: $FP_1 = \mathbb{A}_1 \cap \mathbb{N}$ contains all normal events erroneously classified as attacks by d_1 ;
- True negatives: $TN_1 = (\mathbb{N} \cap \mathbb{T}) - \mathbb{A}_1$ contains all normal events targeted to the system that were correctly identified by d_1 .
- False negatives: $FN_1 = \mathbb{T} - (\mathbb{A}_1 \cup \mathbb{N})$ contains all attacks targeted to the system that were not recognized by d_1 ;

An ideal intrusion detector d_i should present $FP_i = FN_i = \emptyset$ (in other words, no classification errors). However, real detectors can fail, giving false positive or negative results.

3.3 Modeling a composite IDS

A model considering two detectors is presented in Fig. 3, in which the set of events perceived by each detector d_i is represented as $\mathbb{A}_i$.

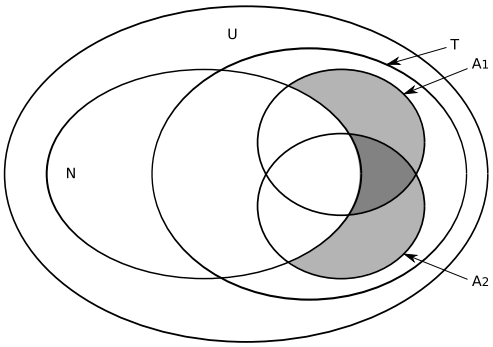

Fig. 3 Venn diagram for two detectors.

All $\mathbb{A}_i$ sets, generated by each d_i detector, can be combined together to generate a compound result $\mathbb{A}_c$, in two basic approaches:

a) by considering the attacks detected by both detectors: $\mathbb{A}_c = \mathbb{A}_1 \cap \mathbb{A}_2$
b) by considering the attacks detected by any detector: $\mathbb{A}_c = \mathbb{A}_1 \cup \mathbb{A}_2$

Their generalization for N detectors are, respectively:

$$\mathbb{A}_c = \bigcap_{i=1}^{N} \mathbb{A}_i \quad \text{or} \quad \mathbb{A}_c = \bigcup_{i=1}^{N} \mathbb{A}_i \tag{1}$$

The subsets of interest for the compound IDS are defined as:

$$TP_c = \mathbb{A}_c - \mathbb{N} \tag{2}$$

$$FP_c = \mathbb{A}_c \cap \mathbb{N} \tag{3}$$

$$TN_c = (\mathbb{N} \cap \mathbb{T}) - \mathbb{A}_c \quad (4)$$

$$FN_c = \mathbb{T} - (\mathbb{A}_c \cup \mathbb{N}) \quad (5)$$

The approach *a* is restrictive, because only attacks detected by all detectors will be considered as attacks (i.e. an event *e* is considered as an attack *iff* $\forall d_i \; e \in \mathbb{A}_i$). This leads to a smaller $\mathbb{A}_c$ set, and consequently to a smaller FP_c set. However, attacks not recognized by all detectors may be ignored in the final result, leading to false negative errors (i.e. a bigger FN_c set). So, it lowers false positive rates, but rises false negative rates (i.e. discards attacks that could have been detected only by some IDS). Therefore, if only one detector of the CIDS detects a given attack, this attack will not be in $\mathbb{A}_c$, but it does not necessarily mean that this event is unimportant. This approach is depicted in Fig. 4.

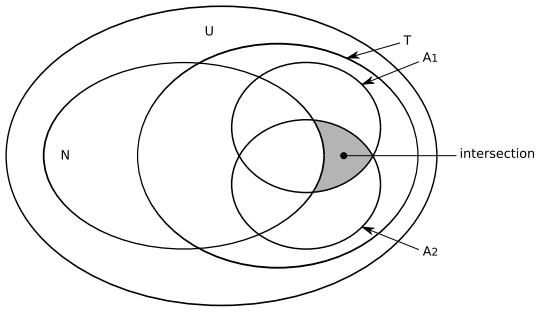

Fig. 4 Intersection of the individual results.

On the other hand, approach *b* is more comprehensive, as even attacks detected by just one detector are considered as attacks (i.e. an event *e* is an attack *iff* $\exists d_i \mid e \in \mathbb{A}_i$). This leads to a bigger $\mathbb{A}_c$ set, and consequently to a smaller FN_c set. However, there is a higher risk of false positive alerts (i.e. a bigger FP_c set). This approach is depicted in Fig. 5.

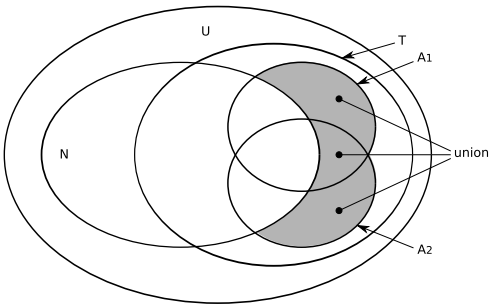

Fig. 5 Union of the individual results.

Both approaches represent extreme situations. In the next section, fuzzy sets are used to propose an intermediate model, mixing characteristics of both solutions.

4 A fuzzy composition model

The first CIDS model presented in Sect. 3 can discard important attacks not observed by all detectors. The second model considers a larger attack set, but increases the uncertainty on the results. Therefore, they should be improved to grasp the best of both worlds: the first model's accuracy and the second model's comprehensiveness. For that, some Fuzzy Set concepts are used to build a more general model. In this new model, the binary result of each detector (whether an event is an attack or not) will have some meaning in the collective result.

To consider all individual detector results, we need to define the *relevance* of a given event e as $r(e)$, meaning how much e is considered as an attack, according to the number of IDSs that detected it and the total number of detectors in the composition. Attacks detected by a larger number of detectors will have a bigger relevance than attacks detected by a smaller number of detectors. The relevance $r(e)$ of an event e can be seen as the *membership function* of a fuzzy set[2]. Thus, the relevance function $r(e)$ should satisfy the following properties:

a) it informs how much the event e can be considered as an attack: if $r(e) = 0$, e is a normal event (not an attack), and if $r(e) = 1$, e is surely an attack;
b) its input parameters are the number of detectors in the CIDS and the number of those that detected the event as an attack;
c) the number of elements in the CIDS should influence the result: it is better to have 100% of detection in a CIDS with 5 detectors than to have a 100% detection in a CIDS with just one detector;
d) it should have a negative exponential behavior, growing faster for the first values and tending to 1 as the number of detections increase; this provides a better sensibility in larger CIDS compositions.

Each intrusion detector d_i provides as output a binary result on each input event: for d_i, e is an attack ($e \in \mathbb{A}_i$) or e is not an attack ($e \notin \mathbb{A}_i$). From this, it is possible to define an *alarm count* $c(e)$, which indicates how many detectors in a CIDS consider the event e as an attack:

$$c(e) = \sum_{i=1}^{N} \left\{ \begin{array}{l} 0, e \notin \mathbb{A}_i \\ 1, e \in \mathbb{A}_i \end{array} \right\} \tag{6}$$

A first candidate for the relevance function would be $r(e) = c(e)/N$. However, although it satisfies the properties a and b, properties c and d are not satisfied. The exponential function $f(x) = 1 - 1/(x+1)$ satisfies properties c and d. The combination of both equations results in:

$$r(e) = \frac{c(e)}{N} \times \left(1 - \left(\frac{1}{c(e)+1}\right)\right) = \frac{c(e)^2}{N(c(e)+1)} \tag{7}$$

[2] A fuzzy set is defined as a pair (S, μ) where S is a set and $\mu : S \to [0,1] \in \mathbb{R}$ is a real function. For each $x \in S$, $\mu(x)$ represents how much x belongs to S. If $\mu(x) = 1$, x belongs totally to S, whereas if $\mu(x) = 0$, x does not belong to S at all [8].

The expected behavior of this $r(e)$ function can be seen in Table 1, which shows the relevance values for various CIDS configurations and detection levels. It shows that $r(e)$ values consider both the number of detectors in the CIDS and how many detectors classified e as an attack.

Table 1 Event relevance $r(e)$ in a CIDS with up to 6 detectors

Detections	Detectors in the CIDS (N)					
$c(e)$	1	2	3	4	5	6
0	0.000	0.000	0.000	0.000	0.000	0.000
1	0.500	0.250	0.167	0.125	0.100	0.083
2	-	0.667	0.444	0.333	0.267	0.222
3	-	-	0.750	0.563	0.450	0.375
4	-	-	-	0.800	0.640	0.533
5	-	-	-	-	0.833	0.694
6	-	-	-	-	-	0.857

To interpret $r(e)$ values, we propose defining a *relevance threshold* (r_t). This threshold defines the minimum relevance level for an event to be considered as an attack by the CIDS. For a given CIDS, its threshold should be tuned for the best results (as presented in the next section). Obviously, the CIDS behavior depends on the r_t threshold. For instance, considering $r_t = 0.5$, a CIDS with 4 detectors will consider an event as relevant only if all detectors detect it, but a CIDS with 6 detectors will do so if at least 4 detectors detect it. The next section presents the application of this model in a real scenario, and its evaluation.

5 Model evaluation

The proposed model was analyzed using data extracted from some controlled experiments. The CIDS performance was compared with individual detectors' performances using ROC curves [16], a technique frequently used to evaluate data classification algorithms.

The setup used for evaluating the proposed model consisted in a hub-based local area network with four distinct IDSs and an "attack generator". The DARPA Data Sets [12], commonly used for evaluating intrusion detectors, were not used here because they contain the MAC addresses of the target machines; some detectors we used only recognize attacks targeted specifically to the MAC addresses of the hosts they are installed in.

The IDSs used in our experiments are *KFSensor 4.2.0* (from *KeyFocus Ltd.*), *X-Ray* (from *GroundZero Security Research*), *HoneyBOT* (from *Atomic Software Solutions*), and *Snort 2.4.3 build 26* (from *Sourcefire, Inc.*), all updated and patched up to the date of the experiment. In order not to make direct comparisons, they will be referred here randomly as d_1, d_2, d_3, and d_4. The attack generator consisted in a computer running the *Nessus* vulnerability scanner, version 3.0.4 (from *Tenable*

Network Security, Inc). As the number of vulnerabilities scanned by *Nessus* is huge, we selected 25 distinct attacks that we found to be representative of frequent situations. Such attacks are presented in Table 2.

Table 2 Attacks used in the evaluation

Attack	Attack name	Nessus category
a_1	BackOrifice	Backdoors
a_2	BugBear	
a_3	DeepThroat	
a_4	FingerBackdoor	
a_5	IISPossibleCompromise	
a_6	PortalOfDoom	
a_7	Sygate BackDoor	
a_8	MyDoom	
a_9	IISFPDoS	DoS
a_{10}	PFPImageFile	
a_{11}	Winlogo.exe DoS	
a_{12}	Personal Web Sharing	
a_{13}	NetStat	Useless Services
a_{14}	Windows Terminal Service	
a_{15}	Telnet	
a_{16}	WriteSrv	
a_{17}	FrontPage Passwordless	WebServers
a_{18}	IISRemoteCommExecution	
a_{19}	CyDoor detection	Windows
a_{20}	GatorDetection+Gain	
a_{21}	I-Nav ActiveX BufferOverflow	
a_{22}	IE VersionCheck	
a_{23}	FTP Shell DoS Vuln	FTP
a_{24}	RPC Port Mapper	RPC
a_{25}	AliBaBa Port Climbing	Remote File Access

Two independent experiments were performed. The first one aimed at identifying the detection capabilities of each detector against such attacks, i.e. true positives (attacks correctly identified) and false negatives (attacks not detected). The second experiment analyzed the behavior of the detectors in the presence of a valid, normal background network traffic, in order to identify false positive detections. Both experiments are described next.

5.1 Experiment 1

In this experiment, each IDS was submitted to the 25 attacks of Table 2, one at a time. Each attack was tested against each IDS three times. Between two tests, the computer running the IDS was restarted, in order to prevent any cross-effects. During the tests, the network was isolated and there was no background traffic. The detection results are presented in Table 3 (each "•" corresponds to a raised alert).

This experiment allowed us to identify attacks correctly detected by the detectors (true positives) and attacks not detected by them (false negatives).

Table 3 Individual IDS alerts

Attack	d_1	d_2	d_3	d_4	$r(a_i)$	Attack	d_1	d_2	d_3	d_4	$r(a_i)$	Attack	d_1	d_2	d_3	d_4	$r(a_i)$
a_1	•	•	•	-	0.563	a_{11}	•	•	•	-	0.563	a_{21}	•	•	-	-	0.333
a_2	•	•	•	-	0.563	a_{12}	•	•	•	•	0.800	a_{22}	•	•	-	-	0.333
a_3	-	-	-	-	0.000	a_{13}	-	•	•	-	0.333	a_{23}	-	•	-	-	0.125
a_4	-	-	-	-	0.000	a_{14}	•	•	-	-	0.333	a_{24}	•	•	•	-	0.563
a_5	•	•	•	-	0.563	a_{15}	-	•	•	-	0.333	a_{25}	•	•	•	-	0.563
a_6	•	-	-	-	0.125	a_{16}	-	-	•	-	0.125						
a_7	•	•	-	-	0.333	a_{17}	•	-	•	-	0.333						
a_8	•	•	•	-	0.563	a_{18}	•	•	•	-	0.563						
a_9	•	-	•	-	0.333	a_{19}	•	•	-	-	0.333						
a_{10}	•	-	•	-	0.333	a_{20}	•	•	-	•	0.563						

5.2 Experiment 2

This experiment was conducted to identify false positive alerts generated by the detectors. For that, the detectors were deployed in a local area network with real traffic (DNS, HTTP, SMTP, POP, IMAP, Windows Terminal Service, NetBios, SNMP, and FTP traffic). The four detectors were deployed and their detections observed during some hours. Each generated alert was manually analyzed to verify its validity. False alerts were classified as false positives (only one instance of each type of event was counted for each detector). Results obtained from this experiment are summarized in Table 4 (each "•" corresponds to a raised alert).

Table 4 Individual false positive alerts

Event	Activity	d_1	d_2	d_3	d_4	$r(n_i)$	Event	Activity	d_1	d_2	d_3	d_4	$r(n_i)$
n_1	TCP 139	•	•	-	-	0.333	n_6	UDP 138	-	•	-	-	0.125
n_2	TCP 3088	•	-	-	-	0.125	n_7	UDP 1027	-	•	-	-	0.125
n_3	TCP 3089	•	-	-	-	0.125	n_8	UDP 2967	-	•	-	-	0.125
n_4	TCP 3090	•	-	-	-	0.125	n_9	UDP 38293	-	-	•	-	0.125
n_5	UDP 137	-	•	-	-	0.125	n_{10}	ICMP	•	•	-	-	0.333

5.3 ROC analysis

ROC (*Receiver Operation Characteristic*) analysis appeared in the 1950's, to help understanding radio signals contaminated by noise. It is frequently used in the med-

ical area, to evaluate the efficiency of medical tests [16]. In computer science, it is used to evaluate data classification algorithms. A ROC analysis basically consists in comparing the *sensitivity* of a classifier (a value related to its True Positives count) with its *specificity* (a value related to its False Positives count). According to [16]:

$$Sensitivity = \frac{TP}{TP+FN} \tag{8}$$

$$Specificity = \frac{TN}{TN+FP} \tag{9}$$

$$TP\ Rate = Sensitivity \tag{10}$$

$$FP\ Rate = 1 - Specificity \tag{11}$$

When drawing the curve of specificity $\times$ sensitivity for a classifier (its ROC curve), it can be shown that its best operating point, in which it presents the best compromise between FP and TP rates, is the point nearest to $[0,1]$ in the curve (where $FP = 0$ and $TP = 1$).

An IDS can be considered as a classifier, as it classifies input events as attacks or normal events. So, it is possible to plot ROC curves from the values gathered in experiments 1 and 2, in order to evaluate our model and to identify the event relevance threshold r_t, as defined in Sect. 4. Table 5 presents the event relevance levels calculated for the attacks listed in Table 2 and the false positives listed in Table 4. Values were calculated according to the $r(e)$ definition in Sect. 4, and results are presented in increasing $r(e)$ order.

Table 5 Ordered event relevance

Event	$r(e)$	Event	$r(e)$
$a_3\ a_4$	0.000	$n_9\ n_2\ n_3\ n_4\ n_5\ n_6\ n_7\ n_8$	0.125
$a_6\ a_{16}\ a_{23}$	0.125	$n_1\ n_{10}$	0.333
$a_7\ a_9\ a_{10}\ a_{13}\ a_{14}\ a_{15}\ a_{17}\ a_{19}\ a_{21}\ a_{22}$	0.333		
$a_1\ a_2\ a_5\ a_8\ a_{11}\ a_{18}\ a_{20}\ a_{24}\ a_{25}$	0.563		
a_{12}	0.800		

Table 6 shows the FP, TN, FN, and TP counts, and the corresponding calculated FP/TP rates. Here, false/true positives and negatives are counted according to the CIDS point of view:

- TN_C: normal events n_i for which $r(n_i) < r_t$
- FP_C: normal events n_i for which $r(n_i) \geq r_t$
- TP_C: attacks a_i for which $r(a_i) \geq r_t$
- FN_C: attacks a_i for which $r(a_i) < r_t$

FP and TP rates can then be plotted in a ROC curve representing the CIDS behavior (Fig. 6). The points in the ROC curve correspond to rows in Table 6.

The best operating point for our CIDS is D, because it is the point nearest to the [0,1] coordinates. So, we can adopt $r_t = 0.333$ as the detection threshold for

Table 6 FP and TP rates for the CIDS

	$r(e)$	TN_C	FP_C	TP_C	FN_C	FP rate	TP rate
A	1.000	10	0	0	25	0.00	0.00
B	0.800	10	0	1	24	0.00	0.04
C	0.563	10	0	10	15	0.00	0.40
D	0.333	8	2	20	5	0.20	0.80
E	0.125	0	10	23	2	1.00	0.92
F	0.000	0	10	25	0	1.00	1.00

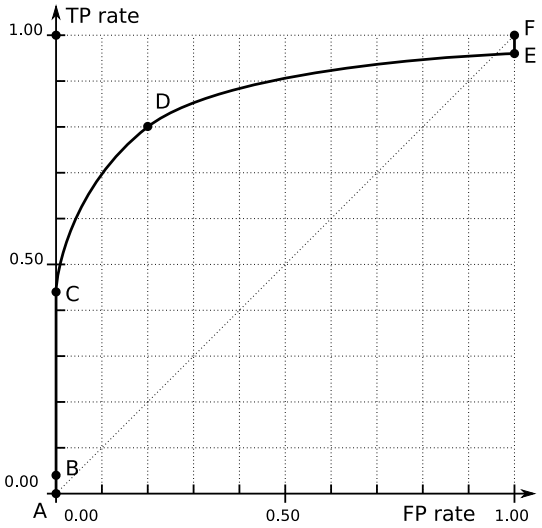

Fig. 6 ROC curve for the composite IDS.

the best detection results, in this case. Table 7 shows a comparison of CIDS results (operating at point D) with the individual detectors' results. Is shows clearly that (a) CIDS presents a smaller amount of false results (considering both false negatives and false positives), and (b) its true positive results are also better than any of the individual detectors.

Table 7 Comparing CIDS with individual detectors

Detector	CIDS	d_1	d_2	d_3	d_4
FN	5	4	5	8	21
FP	2	5	6	1	0
FN+FP	7	9	11	9	21
TP	20	19	18	15	2

To complement this comparison, we plotted the operating points for the individual detectors in the ROC space, using the experiments' data (Fig. 7). It can be seen that the CIDS operating point is closer to the [0,1] ideal point than the individual detectors. It also shows that the best individual detector would be d_3, in this case.

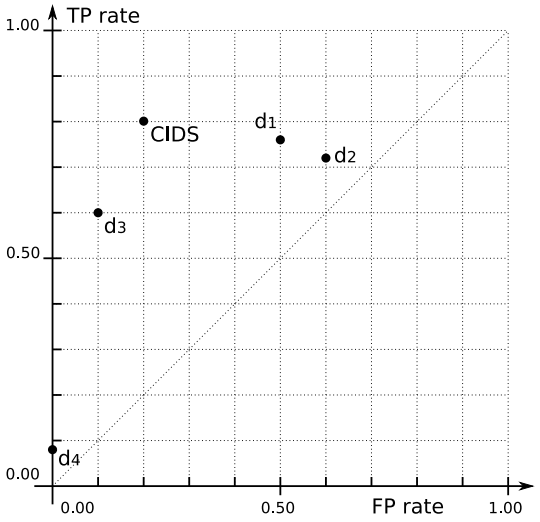

Fig. 7 ROC operating points for all the detectors

6 Model extensions

The experiments presented in Sect. 5 showed that detectors may have distinct detection performances. Consequently, the relevance function $r(e)$ could be adjusted to consider this. It is possible to define an "efficiency" factor $0 \leq \alpha_i \leq 1$ for each detector d_i based on its performance, leading to a new $c(e)$ definition:

$$c(e) = \sum_{i=1}^{N} \left\{ \begin{array}{l} 0, e \notin \mathbb{A}_i \\ \alpha_i, e \in \mathbb{A}_i \end{array} \right\} \tag{12}$$

The efficiency α_i of a given detector can be estimated from its *FP* and *TP* rates on the ROC curve. Usually, the euclidean distance from a detector's operating point $[FP_i, TP_i]$ to the $y = x$ diagonal is a good indicator of its efficiency [16]. In the ROC space, the $y = x$ line represents operating points in which FP and TP rates are equal. In such points, the detector's results are not better than random guesses. Thus, the $y = x$ line is also called the "random guess line".

Another possible extension to our model would be to define a "confidence" $0 \leq \gamma_i^k \leq 1$ of each detector d_i on each event e_k it classifies as an attack. This would be taken into account into $c(e_k)$ as:

$$c(e_k) = \sum_{i=1}^{N} \left\{ \begin{array}{l} 0, e_k \notin \mathbb{A}_i \\ \alpha_i \times \gamma_i^k, e_k \in \mathbb{A}_i \end{array} \right\} \tag{13}$$

However, this extension considers that each detector can inform its confidence on the alarms it produces, which it is not always the case. The manual definition of individual alarm confidences for each detector would not be feasible, either.

7 Related work

We identified some works that used mathematical models to represent and/or analyze intrusion detectors, some of them using fuzzy logic. Paper [11] defines a methodology to build anomaly-based IDSs called BSEADS (*Behavioral Secure Enclave Attack Detection System*). That methodology combines several data sources, like network traffic and user behavior, to identify anomalous behaviors. BSEADS analysis was done using a model similar to that presented in Sect. 3.2 (Fig. 2).

Article [18] presents *Vismath*, a geometric model that visually represents the variations in a computer environment. The model uses graphical structures called *spicules* to create vectorial representations of monitored variables, like processor usage, number of processes, and number of open files. Once the normal behavior of the system is defined, spicules monitoring allows identifying abrupt changes and anomalous behaviors.

Paper [17] is the closest one to our approach. It presents a method to evaluate the performance of an IDS using ROC curves and a cost-based decision-tree analysis. They also propose to build a composite IDS by the combinations of two individual detectors, but there are several differences between that work and ours. First, the behavior of their composite IDS is determined only by superposing the ROC curves obtained from individual detectors, instead of defining a generic composition model that would have its own ROC curve. Also, in the decision-tree approach they propose, the results obtained by a CIDS can be equivalent to those obtained by an individual IDS (if only the corresponding IDS in the CIDS detects the attack). This behavior is close to those for the simpler models presented here, in Sect. 3.3.

8 Conclusion

This article proposes building a Composite IDS (CIDS) from individual heterogeneous detectors, according to the project diversity principles. A first model allows us to treat the results of a CIDS in two possible ways: in the first, more restrictive, only attacks detected by all IDSs are considered; the second, more comprehensive, considers all attacks detected by any of the detectors. In order to use the best of both worlds, we propose another model, based on the fuzzy sets theory.

The proposed model was evaluated using two experiments, in which four individual detectors were tested against an “attack generator” (a vulnerability scanner). The results showed that the CIDS results are better than individual detector results, giving less false results and more true results for the attacks under consideration.

During our experiments, two instances of the same IDS, running in exactly the same operating system, but in distinct hardware, behaved distinctly and produced distinct results. This issue was also observed between instances of the same IDS running in the same hardware, but on top of distinct operating systems. This leads us to conclude that project diversity is not only a matter of IDS implementation, but also of running environment diversity (hardware and operating systems).

Possible future research includes a deeper analysis of the composition model, using the DARPA Intrusion Detection Evaluation Data Sets [12]. We intend also to evaluate the effectiveness of the extensions proposed in Sect. 6.

References

1. A. Avizienis and J. P. Kelly. Fault tolerance by design diversity: Concepts and experiments. *IEEE Computer*, pages 67–80, August 1984.
2. S. Bachi, Y. Mei, B. Boo B, and Y Wu. Collaborative intrusion detection system (CIDS): A framework for accurate and efficient IDS. In *Annual Computer Security Applications Conference*, 2003.
3. N. Carey, A. Clark, and G. Mohay. IDS interoperability and correlation using IDMEF and commodity systems. In *Intl Conference on Information and Communications Security*, 2002.
4. O. Dain and R. Cunningham. Fusing heterogeneous alert streams into scenarios. In *ACM Conference on Computer and Communications Security*, 2001.
5. F. Cuppens F. and Miège. Alert correlation in a cooperative intrusion detection framework. In *IEEE Symposium on Security and Privacy*, 2002.
6. K. Julisch. Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security*, November 2003.
7. C. Kahn, P. Porras, S. Staniford-Chen, and B. Tung. A common intrusion detection framework, 1998.
8. G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall PTR, 1995.
9. K. Ko, T. Fraser, L. Badger, and D. Kilpatrick. Detecting and countering system intrusions using software wrappers. In *USENIX Security Symposium*, 2000.
10. C. Kreibich and R. Sommer. Policy-controlled event management for distributed intrusion detection. In *Intl Workshop on Distributed Event-Based Systems*, June 2005.
11. T. Leckie and A. Yasinsac. Metadata for anomaly-based security protocol attack deduction. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1157–1168, September 2004.
12. R. Lippmann, J. Haines, D. Fried, J. Korba, and K. Das. The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks*, 34(4):579–595, 2000.
13. B. Littlewood and Stringini. Redundancy and diversity in security. In *European Symposium on Research in Computer Security*, France, 2004.
14. R. Maxion and K. Tan. The effects of algorithmic diversity on anomaly detector performance. In *IEEE/IFIP Intl Conference on Dependable Systems and Networks*, July 2005.
15. P. Mell, V. Hu, R. Lippmann, J. Haines, and M. Zissman. An overview of issues in testing intrusion detection systems. Technical Report Interagency Report 7007, National Institute of Standards and Technologies, June 2003.
16. C. E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298, 1978.
17. J. Ulvila and J. Gaffney Jr. Evaluation of intrusion detection systems. *NIST Journal of Research*, 108(6), November 2003.
18. G. Vert, D. Frincke, and J. McConnell. A visual mathematical model for intrusion detection. In *21st NIST-NCSC National Information Systems Security Conference*, 1998.