Performance Evaluation for Automated Threat Detection

ROBERT C. SCHRAG

MASAMI TAKIKAWA

PAUL GOGER

JAMES EILBERT

We have developed a performance evaluation laboratory (PE Lab) to assess automated technologies that fuse fragmentary, partial information about individuals' activities to detect modeled terrorist threat individuals, groups, and events whose evidence traces are embedded in a background dominated by evidence from similarly modeled non-threat phenomena. We have developed the PE Lab's main components—a test dataset generator and a hypothesis scorer—to address two key challenges of counter-terrorism threat detection performance evaluation:

- **Acquiring adequate test data to support systematic experimentation; and**
- **Scoring structured hypotheses that reflect modeled threat objects' attribute values and inter-relationships.**

The generator is parameterized so that the threat detection problem's difficulty may be varied along multiple dimensions (*e.g.*, dataset size, signal-to-noise ratio, evidence corruption level). We describe and illustrate, using a case study, our methodology for constraint-based experiment design and non-parametric statistical analysis to identify which among varied dataset characteristics most influence a given technology's performance on a given detection task.

The scorer generalizes metrics (precision, recall, F-value, area under curve) traditional in information retrieval to accommodate partial matching over structured case hypothesis objects with weighted attributes. Threat detection technologies may process time-stamped evidence in either batch, forensic mode (to tender threat event hypotheses retrospectively) or in incremental, warning mode (to tender event hypotheses prospectively—as "alerts"). Alerts present additional scoring issues (*e.g.*, timeliness) for which we discuss practical techniques.

PE Lab technology should be similarly effective for information fusion or situation assessment technologies applied in other domains (besides counter-terrorism), where performance evaluation presents similar challenges.

Manuscript received April 23, 2007; revised October 29, 2008; released for publication February 29, 2008.

Refereeing of this contribution was handled by Pierre Valin.

Authors' addresses: R. Schrag, Global InfoTek, Inc., 1920 Association Dr., Suite 200, Reston, VA 20191, E-mail: (rschrag@globalinfotek.com); M. Takikawa, Art Technology Group, Inc., 673 NW Jackson Ave., Corvallis, OR 97330, E-mail: (takikawa@atg.com); P. Goger, Metron, Inc., 19111 Freedom Dr., Suite 800, Reston, VA 20190, E-mail: (goger@metsci.com); J. Eilbert, AP Technology, LLC, 6 Forrest Central Dr., Titusville, NJ 08560, E-mail: (jleilbert@verizon.com).

R. Schrag and M. Takikawa were employed by Information Extraction and Transport, Inc. (IET) when this work was begun. J. Eilbert was employed by CHI Systems, Inc.

1557-6418/07/$17.00 © 2007 JAIF

1. INTRODUCTION

Threat detection by sifting high-volume data streams for indicators has been likened to the problem of recognizing a complete "threat" needle by selecting from among many haystack-sized piles of threat and non-threat needle pieces [33]. Under this analogy, problem difficulty may vary depending on factors such as how many stacks there are, how many threat and non-threat needles are distributed among them, and how like are threat and non-threat needles. A key goal in developing a performance evaluation laboratory (PE Lab) is to understand how variation along dimensions like these can affect the performance of a threat detection technology.

As the haystack analogy suggests, many characteristics that contribute to threat detection's difficulty may be modeled simply using convenient abstractions of real-world phenomena. We want to identify well-performing regions of an information fusion approach—*e.g.*, its power to resolve ambiguities arising from partial, potentially corrupted, and temporally overlapping evidence fragments. We deliberately aim to drive the evaluated technology toward explicit representations of and reasoning about structured data and connections between entities and events. Abstraction serves to factor out issues inessential to this, and we model key relationships among threat and non-threat actors, events, and evidence characteristics approximating qualitative real-world relationships and quantitative values. We also factor out user interaction—*e.g.*, evidence visualization and mixed-initiative hypothesis development—so that technology evaluation is in principle entirely automated (although in practice we have not yet required hands-off execution for detection technologies).

We have followed these principles in developing the PE Lab's dataset generator during a multi-year, multi-contractor, multi-agency Government research program, where it has served in several program-wide technology evaluations that have necessitated our development of novel, compatible scoring methods. Many results have already been reported [1] [2] [3] [4] [5] [7] [8] [9] [10] [11] [12] [13] [14] [18] [22] [23] [24] [25] [29] [30] [31] [32] [40] [41] [43] [44].

The PE Lab is schematized in Fig. 1, where square-cornered boxes represent artifacts, round-cornered boxes represent processes, and arrows represent flow of artifacts. The threat detection component—assumed to employ link discovery (LD) technology and also referred to here as an LD component—is rendered 3-dimensionally to indicate its status outside of the PE Lab proper (as the technology under test).

Synthetic dataset generation creates evidence used to challenge LD and (synthetic) ground truth used in scoring LD's hypotheses. LD processes evidence to hypothesize threat phenomena. Generation uses simulation driven by discrete, stochastic event patterns that also are provided to LD. Hypothesis scoring compares technologies' output hypotheses to ground truth.

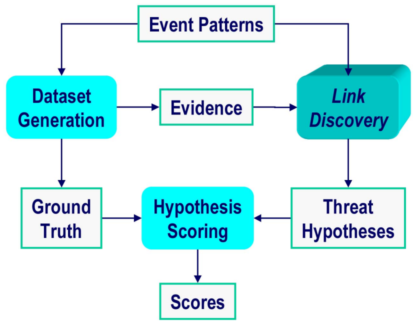


Fig. 1. PE Lab schematic.

In subsequent sections, we describe the following.

- Abstract challenge problem domain (Section 2)
- General hypothesis scoring methods (Section 3)
- Alert scoring methods (Section 4)
- Experiment design to identify performance influences in the problem space (Section 5)
- PE Lab advantages for information fusion system design (Section 6)
- Conclusions (Section 7)

2. COUNTER-TERRORISM THREAT DOMAIN

Synthetic datasets have the advantage for evaluation that (synthetic) ground truth is readily available for scoring. To support unclassified, exploratory counter-terrorism research, we have developed synthetic datasets presenting the same key sources of threat detection difficulty that intelligence analysts have described in real data. In the terminology of the information fusion community [37] [38], real and synthetic datasets present common "referencing" and "registration" problems: *Is the "man in the white shirt" in one report the same "man in a white shirt" described in another report about a different event?* They also present several types of "association" problems: *Are several lower-level events all parts of the same higher-level event, or are people members of the same organization?* Finally, they present "estimation" problems: *Is a group of events that have already been associated really an instance of a particular type of behavior, and if so can upcoming events be predicted based on our model of the behavior?* The issues addressed by PE Lab-based evaluation fall mainly in the Joint Director of Laboratories data fusion model's Level 2, estimation of relationships among entities—or situation assessment [21].

We generate our synthetic datasets over an artificial world that is tunable, mitigating privacy and security classification concerns and supporting systematic experimentation. That our artificial world is also abstract facilitates parameterized overlap between threat and non-threat activities and de-emphasizes knowledge representation and reasoning requirements in comparison to (threat) signal detection requirements, consistent with the funding program's goals. Our synthetic datasets, while thus simplified, present deliberately selected technical challenges.

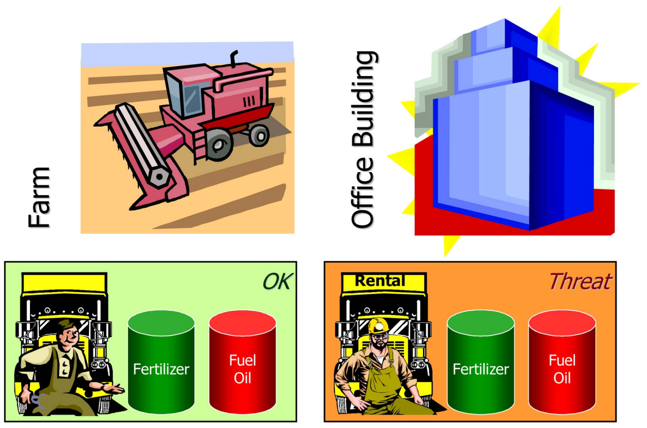


Fig. 2. Real-world motivation for challenge problem.

Fig. 2 exhibits some real-world motivation behind the abstract, artificial world challenge problem domain we have developed. The PE lab's dataset generator uses an artificial world abstraction style inspired by that of Hats [27] [28]. A key difference is that the PE Lab is structured deliberately to emphasize exploratory experimentation, as described in Section 5.

On the left-hand side of Fig. 2, "Farmer Fred" buys fertilizer and fuel oil and transports these *via* truck to his farm. He applies the fertilizer using his tractor which (along with his truck) burns the fuel oil. (Fred is an honest, hard-working man.) On the right-hand side, "Demolition Dan" acquires the same resources but mixes them into a slurry that he transports (*via* rental truck) to the basement of an office building. (Dan is up to no good.)

In the artificial world, capabilities (like farming and demolition) and resources (like fertilizer and fuel oil) are mapped to abstract elements that individuals can possess intrinsically or acquire. Infrastructure elements (like office buildings) are mapped to "targets" that support both legitimate/productive and destructive modes of use or "exploitation." Non-threat and threat individuals (like Fred and Dan) each may belong to any of various groups whose members collaborate in sub-group teams towards different goals. Exploitations play out the general scheme of Fig. 3.

The exploitation scheme on the left-hand side of Fig. 3 includes four main, sequential subevents, each of which unfolds through several levels of task decomposition (illustrated in Fig. 4), bottoming out in transactions with record types indicated on the right-hand side of Fig. 3 and at the bottom of Fig. 4. In a threat exploitation, the final, consummation phase—in which capabilities and resources are (destructively) applied to the target—defines the time by which alerting must occur to be at all effective. Transactions appearing in incrementally presented, time-stamped evidence are the

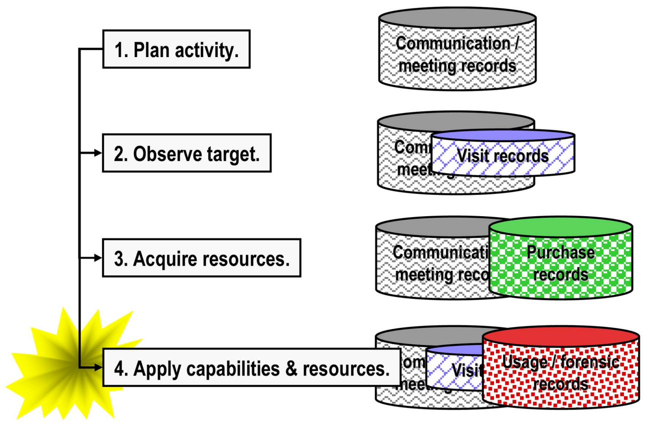


Fig. 3. Generic exploitation scheme.

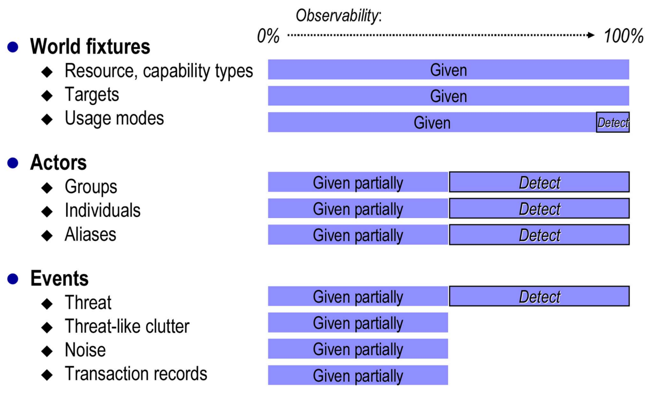


Fig. 5. Threat detection objectives and notional instance observabilities.

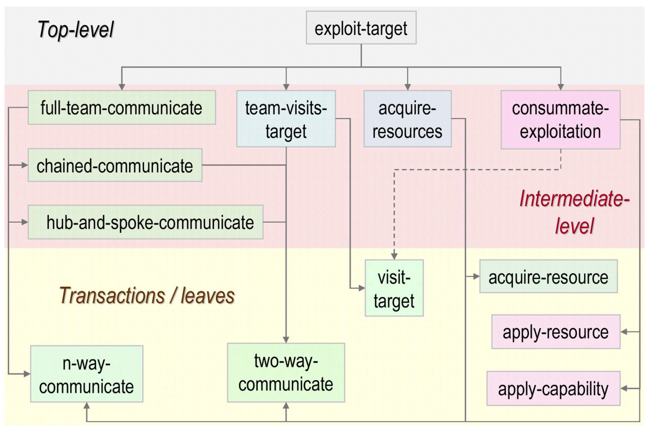


Fig. 4. Invocation relationships among event generation patterns.

sole basis LD has for issuing alerts; intermediate-level events are never materialized in evidence.

In the real world, people typically interact simultaneously in several different social spheres associated with (*e.g.*) work, family, faith, neighborhood, sports/hobbies, civic involvement, shopping, and other relationships. People interact to coordinate times and locations for all of their activities, negotiate inter-activity constraints, and travel as necessary to interact. To make large dataset generation efficient, we have abstracted away such details, modeling all group activities with the same abstraction (the exploitation pattern), allowing individuals to participate in arbitrarily many activities simultaneously, and assuming that all activities take place in a single location (*e.g.*, a metropolitan area).

The challenge to threat detection technology is to identify and report threat *cases*—top-level objects with attributes and values summarizing extant threat phenomena at a level sufficient for scoring. The case types that are LD is tasked to detect include threat actors (groups, individuals, and their aliases) and (ideally, impending) threat events/attacks. To perform this challenge, an automated threat detector is given information about the underlying artificial world that is relatively complete (excepting only a few, novel exploitation modes) and about events and actors that is only partial—per settings of "observability" parameters, as depicted notionally in Fig. 5.

We further describe the artificial world problem domain as follows.

- *Individuals* have *assets.*
  —They have permanent *capabilities.*
  —They can acquire consumable *resources* as necessary to exploit a target in one of its modes.
- Both resources and capabilities are abstract enumerations.
- *Exploitation modes* are sets of capabilities and resources.
  —*Vulnerability modes* are exploited by threat actors.
  —*Productivity modes* are exploited by both threat and non-threat actors.
- *Groups* are collections of individuals. Only *threat individuals* belong to *threat groups.* Both threat and *non-threat* individuals can belong to *non-threat groups.* Groups have designated exploitation modes—vulnerability modes for threat groups and productivity modes for both group types. A group can exploit a target that exhibits one of its modes.
- Groups have subgroups—*exploitation teams*—that focus on particular exploitation modes for which a team has qualified members.
- Groups' and teams' member individuals tend to share abstract *social/demographic attributes.*
- *Noise events* masking threat activity occur at several levels. We refer to non-threat exploitations as *clutter. Structured noise* events share intermediate structure with exploitations. *Transaction noise* events are atomic.

In this world, inter-connections abound. Modes overlap with respect to capabilities and resources (as suggested in Fig. 2). Groups overlap with respect to modes, as do targets. Individuals overlap with respect to teams and groups and with respect to capabilities. Exploitations overlap in time with each other and with noise and clutter events. All of these inter-connections contribute to threat detection difficulty.

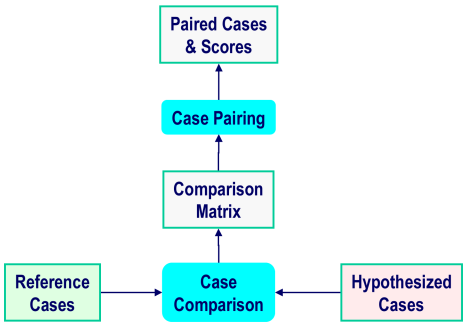


Fig. 6. Generic hypothesis scoring scheme.

3. GENERAL HYPOTHESIS SCORING METHODS

We want to score structured hypotheses that reflect modeled threat objects' attribute values and interrelationships—*e.g.*, a threat event case mentions a threat group case, which includes threat individuals that may be named by their aliases. (Section 3.2 gives full attribute type details.) In this object-oriented context, we need metrics analogous to traditional information fusion's probability of detection and probability of false alarm. For this purpose, we generalize the related recall and precision metrics from traditional information retrieval to accommodate partial matching over structured objects with weighted attributes. Fig. 6 depicts the generic scoring scheme. We require LD to return hypothesis objects that are definite (incorporate neither logical nor probabilistic uncertainty).

In Fig. 6, the reference cases are summaries computed from ground truth, and the hypothesized cases are from LD. Because case objects have significant structure, we want to credit LD when hypothesized cases approximately match reference cases. Match quality is determined by case comparison. When a hypothesized case object's existence has been inferred from lower-level evidence, we can decide which reference case to pair the hypothesized one with only by comparing the hypothesized case with all relevant reference cases—on the basis of their attribute values. We store comparison results for the candidate pairs in a matrix. With inexact matching, it also can be ambiguous which of the one-to-one mappings admitted by the candidate pairs should be selected, so we use an algorithm that optimizes dataset-level scores. Given these pairs, we compute scores for object-oriented metrics based on traditional precision, recall, and F-value metrics.

Subsequent subsections present our scoring methods in more detail, as follows.

- Section 3.1 summarizes the issues of case comparison and case pairing that arise with inexact structured object matching.
- Section 3.2 summarizes the scored object types and attributes in our counter-terrorism domain.
- Section 3.3 presents the algorithmic details of case comparison.

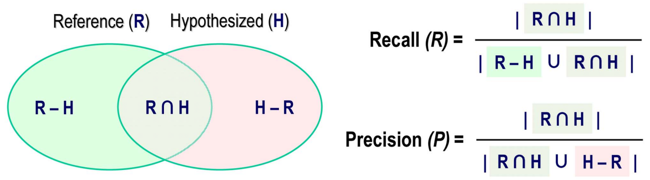

Reference (R)
Hypothesized (H)
R – H
R ∩ H
H – R
Recall (R) = | R ∩ H | / | R – H ∪ R ∩ H |
Precision (P) = | R ∩ H | / | R ∩ H ∪ H – R |

Fig. 7. Traditional precision and recall.

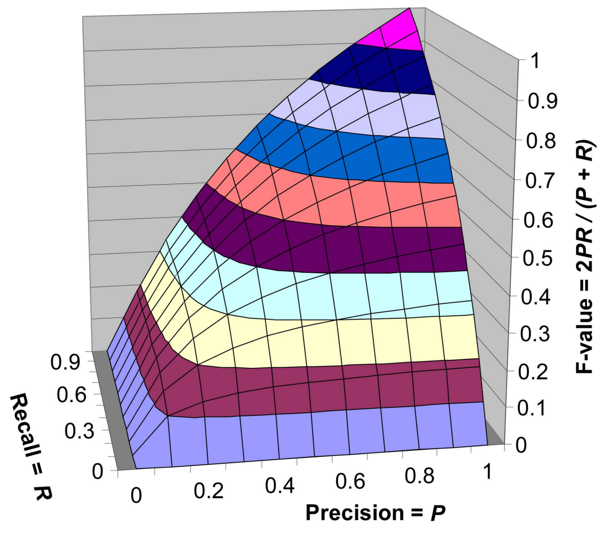


Fig. 8. The F-value surface.

- Section 3.4 describes how we apply the algorithm at the attribute and value level.
- Section 3.5 summarizes additional extant and contemplated hypothesis scoring capabilities.
- Section 3.6 discusses others' work related to our hypothesis scoring approach.

3.1. Case Comparison and Pairing

Case comparison determines the quality of match between any two cases. We characterize this quality by generalizing the traditional precision and recall metrics that presume exact matching between hypothesized and reference items. Fig. 7 illustrates the traditional versions of these metrics.

Traditionally, recall R is the number of valid hypotheses divided by the number of detection targets (the required number of valid hypotheses). Precision P is the number of valid hypotheses divided by the number of all hypotheses.

A single metric to summarize the values of recall and precision is frequently useful, and it is traditional to appeal to F-value = $2PR/(P+R)$—the harmonic mean of precision and recall. (When both precision and recall are zero, we define F-value as zero.)

F-value (shown in Fig. 8 and also known as "F-score" or "F-measure") has the same extremes as a simple average of precision and recall but discounts differences between them (less aggressively than $\min(P,R)$ would discount such differences).

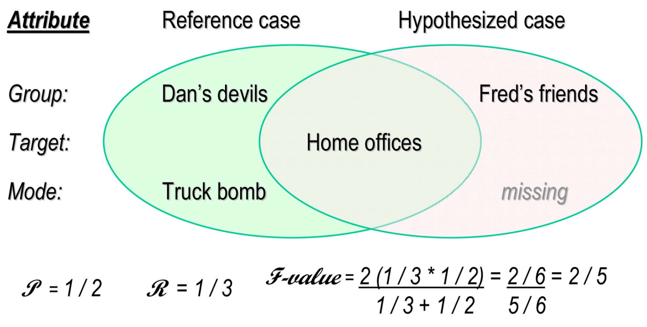


Fig. 9. Object-oriented metrics.

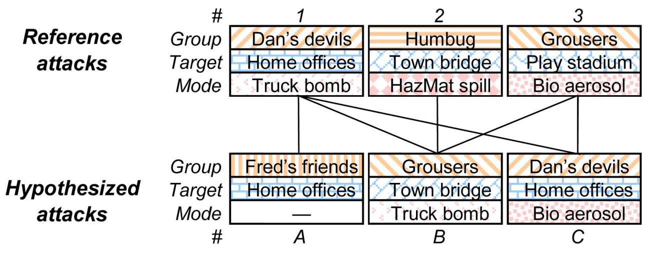
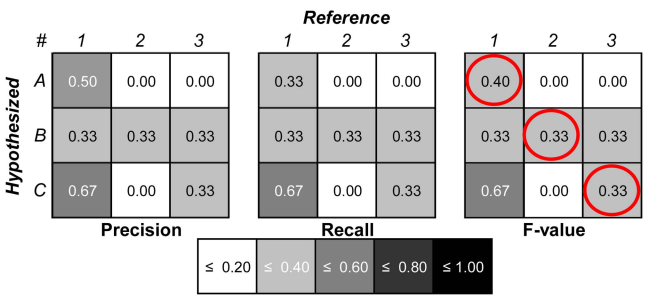

Reference attacks
Hypothesized attacks
#
Group
Target
Mode
1 Dan's devils Home offices Truck bomb
2 Humbug Town bridge HazMat spill
3 Grousers Play stadium Bio aerosol
A Fred's friends Home offices —
B Grousers Town bridge Truck bomb
C Dan's devils Home offices Bio aerosol

Fig. 10. Case pairing issue.


Reference
Hypothesized
#
Precision
Recall
F-value
≤ 0.20
≤ 0.40
≤ 0.60
≤ 0.80
≤ 1.00

#	Precision 1	Precision 2	Precision 3	Recall 1	Recall 2	Recall 3	F-value 1	F-value 2	F-value 3
A	0.50	0.00	0.00	0.33	0.00	0.00	0.40	0.00	0.00
B	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
C	0.67	0.00	0.33	0.67	0.00	0.33	0.67	0.00	0.33

Fig. 11. Case pairing matrix and metrics.

To accommodate inexact matching over structured case objects, we define object-oriented versions of precision, recall, and F-value, as illustrated in Fig. 9. Our complete definitions—in Section 3.3—address object attributes that may be weighted differently, so that attributes contribute to scores non-uniformly.

Of the three attribute values in the reference case of Fig. 9, the hypothesized case agrees only for the Target attribute, so the object-oriented recall score $\mathcal{R}$ is 1/3. Of the two attributes included in the hypothesis, only one agrees with the reference, so the object-oriented precision score P is 1/2. The corresponding object-oriented F-value ($\mathcal{F}$-value) is 2/5, as shown.

Case pairing determines which hypothesized cases to pair with which reference cases—since this may not be obvious, as illustrated in Fig. 10.

In Fig. 10, we have three reference and three hypothesized attack cases. (Reference Case 1 and Hypothesized Case A correspond to the pairing of Fig. 9.) Links appear in the bipartite graph between reference and hypothesized cases wherever these share one or more attributes. Fig. 11 illustrates how we perform one-to-one case pairing using a matrix over all possible pairings.

In Fig. 11, we compute per-pair object-oriented precision, recall, and F-value (as in Fig. 9). Then we use an optimization algorithm to select (red-circled) pairs leading to the greatest average object-oriented F-value. So, we have computed a matching for Fig. 10's bipartite case graph including just the strictly vertical edges.

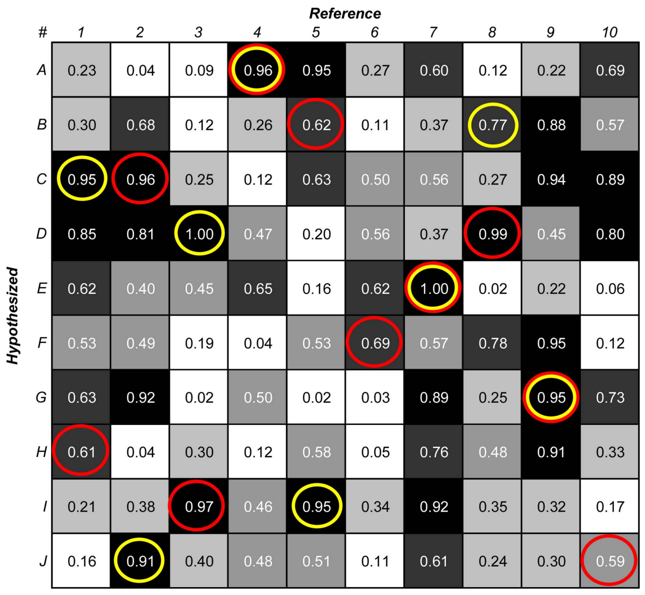
Case pairing is necessary only for objects whose existence LD has hypothesized based on lower-level evidence, when we require it to invent a unique identifier (UID) in its own namespace. Otherwise LD reports objects' UIDs as they appear in evidence. We forcibly pair any like-UID hypothesized and reference objects, and we omit them from the case pairing matrix.

When the numbers of reference and hypothesized cases do not match, we (effectively) pad the matrix, as necessary to make it square, with null cases. Precision and recall with respect to null cases are defined as zero.

As an optimization algorithm to select a best-scoring one-to-one case pairing, we have often used greedy, heuristic search with a sparse matrix representation that can process thousands of structured hypothesized and reference cases in tens of minutes on conventional hardware. We also have implemented an optimal assignment algorithm [19] that can process hundreds of cases in minutes. The greedy algorithm always selects next the best score in the yet-unselected row or column with the greatest standard deviation among F-values, thus has $O(n^2)$ behavior. In practice, when its F-values differ from those optimally computed, it is only by a few percentage points. The optimal algorithm is $O(n^3)$. For small n (up to about 10,000), both algorithms are dominated by the $O(n^2)$ matrix set-up time. Our current implementation doesn't support non-sparse, square matrices of more than about 5,000 on a side, though, and, as the optimal algorithm does not readily accommodate a sparse matrix representation, we fall back to the greedy algorithm as an alternative. We also are interested in the optimal forward/reverse asymmetric assignment algorithm of Bertsekas and Castañon [6] which is reported to work efficiently with sparse matrices.

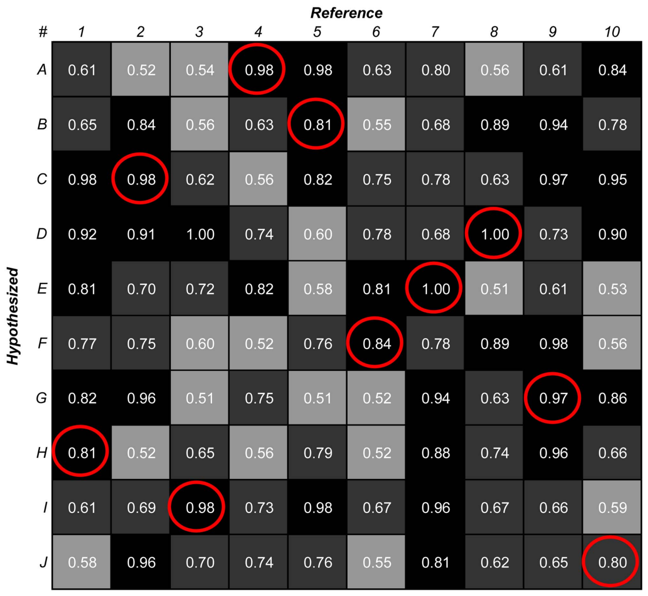
To help illustrate how we develop dataset-level metrics, Fig. 12 depicts a somewhat larger, notional case pairing matrix. Again, it holds computed object-oriented F-values for candidate reference-*versus*-hypothesized case pairings.

In Fig. 12, we again have circled entries to optimize average F-value in a one-to-one pairing. Generally, we admit entries to candidacy per a user-specified F-value threshold that must be exceeded for a hypothesis to be deemed adequate for an analyst or other consumer. The larger, red circles apply when all non-zero entries are candidates for pairing (*i.e.*, when the threshold is zero). The smaller, yellow circles apply when the threshold is set at 0.75. Notice that the pairings under the different thresholds are different—each case pairing process considers just the eligible entries. Giving these pairs and object-oriented precision and recall scores, we compute dataset-level precision and recall. Under zero thresholding, the dataset's average object-oriented F-value is



	#	1	2	3	4	5	6	7	8	9	10
						Reference					
Hypothesized	A	0.23	0.04	0.09	0.96	0.95	0.27	0.60	0.12	0.22	0.69
	B	0.30	0.68	0.12	0.26	0.62	0.11	0.37	0.77	0.88	0.57
	C	0.95	0.96	0.25	0.12	0.63	0.50	0.56	0.27	0.94	0.89
	D	0.85	0.81	1.00	0.47	0.20	0.56	0.37	0.99	0.45	0.80
	E	0.62	0.40	0.45	0.65	0.16	0.62	1.00	0.02	0.22	0.06
	F	0.53	0.49	0.19	0.04	0.53	0.69	0.57	0.78	0.95	0.12
	G	0.63	0.92	0.02	0.50	0.02	0.03	0.89	0.25	0.95	0.73
	H	0.61	0.04	0.30	0.12	0.58	0.05	0.76	0.48	0.91	0.33
	I	0.21	0.38	0.97	0.46	0.95	0.34	0.92	0.35	0.32	0.17
	J	0.16	0.91	0.40	0.48	0.51	0.11	0.61	0.24	0.30	0.59

Fig. 12. Larger, notional case pairing matrix.



	#	1	2	3	4	5	6	7	8	9	10
						Reference					
Hypothesized	A	0.61	0.52	0.54	0.98	0.98	0.63	0.80	0.56	0.61	0.84
	B	0.65	0.84	0.56	0.63	0.81	0.55	0.68	0.89	0.94	0.78
	C	0.98	0.98	0.62	0.56	0.82	0.75	0.78	0.63	0.97	0.95
	D	0.92	0.91	1.00	0.74	0.60	0.78	0.68	1.00	0.73	0.90
	E	0.81	0.70	0.72	0.82	0.58	0.81	1.00	0.51	0.61	0.53
	F	0.77	0.75	0.60	0.52	0.76	0.84	0.78	0.89	0.98	0.56
	G	0.82	0.96	0.51	0.75	0.51	0.52	0.94	0.63	0.97	0.86
	H	0.81	0.52	0.65	0.56	0.79	0.52	0.88	0.74	0.96	0.66
	I	0.61	0.69	0.98	0.73	0.98	0.67	0.96	0.67	0.66	0.59
	J	0.58	0.96	0.70	0.74	0.76	0.55	0.81	0.62	0.65	0.80

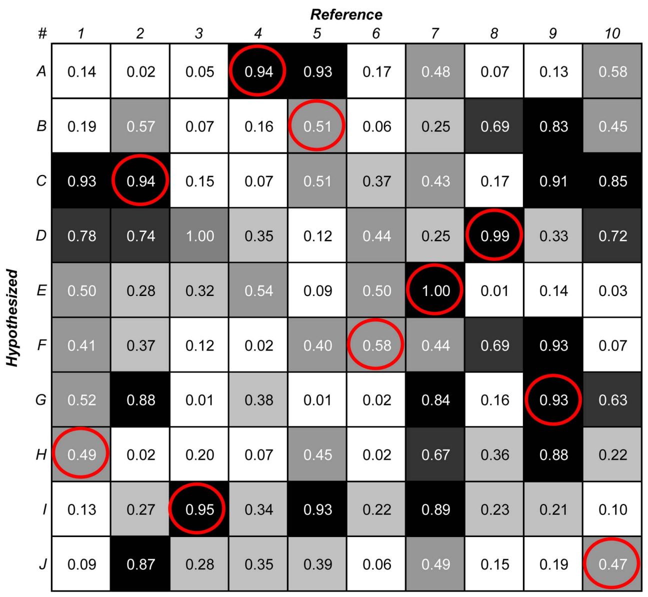
Fig. 13. Notional recall values for Fig. 12's F-values.

0.834, and the traditional metrics aren't particularly informative (always equal one), as all cases (given equal numbers) are paired. Under non-zero thresholding, we always apply traditional metrics given these pairings, so the dataset-level precision, recall, and F-value under the 0.75 threshold all equal 0.8. (A potential expedient would be to develop different higher-thresholded dataset-level scores from a single, zero- or otherwise-lower-thresholded case pairing by dropping any below-threshold pairs.)

To support forthcoming examples, we exhibit, in Fig. 13 and Fig. 14, matrices with notional object-oriented precision and recall values that combine to yield the F-values in Fig. 12. For each cell in the F-value matrix, we have set $\mathcal{R} = (1 + \mathcal{F}\text{-value})/2$ and $\mathcal{P} = \mathcal{R}(\mathcal{F}\text{-value})/(2\mathcal{R} - \mathcal{F}\text{-value})$. We have set the F-value threshold for case pairing at zero and include only the larger, red circles from Fig. 12.

When LD can rank its hypotheses with respect to estimated quality, this ranking supports developing a precision-recall curve and computing the area under the curve (AUC). Any consistently applied variant of precision and recall—*e.g.*, using any consistent F-value threshold—suffices here. Fig. 15 illustrates the AUC for the example values in Fig. 13 and Fig. 14, under two different hypothesis rankings.

At each ith curve point, we compute precision and recall with respect to the full reference case set and the set of LD's 1st- through ith-ranked hypotheses. Fig. 15 notes the hypotheses accounted for by each rectangle contributing to the example's AUC supposing LD ranks its hypotheses in the order (A, B, C, D, E, F, G, H, I, J). Instead of performing full case pairing at each point, we expediently take the case pairings over the full sets of reference and hypothesized cases as authoritative and impose them as we consider each successively presented case to develop the curve.



	#	1	2	3	4	5	6	7	8	9	10
						Reference					
Hypothesized	A	0.14	0.02	0.05	0.94	0.93	0.17	0.48	0.07	0.13	0.58
	B	0.19	0.57	0.07	0.16	0.51	0.06	0.25	0.69	0.83	0.45
	C	0.93	0.94	0.15	0.07	0.51	0.37	0.43	0.17	0.91	0.85
	D	0.78	0.74	1.00	0.35	0.12	0.44	0.25	0.99	0.33	0.72
	E	0.50	0.28	0.32	0.54	0.09	0.50	1.00	0.01	0.14	0.03
	F	0.41	0.37	0.12	0.02	0.40	0.58	0.44	0.69	0.93	0.07
	G	0.52	0.88	0.01	0.38	0.01	0.02	0.84	0.16	0.93	0.63
	H	0.49	0.02	0.20	0.07	0.45	0.02	0.67	0.36	0.88	0.22
	I	0.13	0.27	0.95	0.34	0.93	0.22	0.89	0.23	0.21	0.10
	J	0.09	0.87	0.28	0.35	0.39	0.06	0.49	0.15	0.19	0.47

Fig. 14. Notional precision values for Fig. 12's F-values.

AUC summarizes LD's performance on a dataset in a way that rewards good rankings. When LD returns lower-quality hypotheses earlier, these drag down incrementally computed scores for later points in the curve as well. When LD returns its hypotheses in the order (E, D, I, C, A, G, F, B, H, J) induced by decreasing F-value scores, the AUC rises to 0.851.

Note that our detection task, where structured threat hypotheses must be developed from lower-level evidence, leads to expectations different from those for the traditional information retrieval task, where every presented item merely must be classified as positive or negative. In the traditional setting, the AUC expected from a strategy of pure guessing is 0.5. In our setting, some reference threats may never be reported, given what-

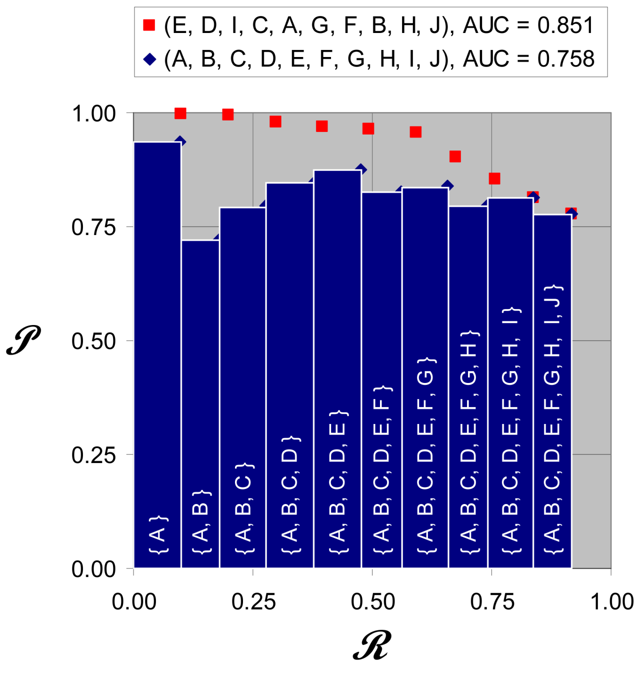


Fig. 15. Precision-recall curve and area for the example values in Fig. 13 and Fig. 14.

TABLE I
Scoring-Relevant Types and Attributes

Scoring-relevant type	Scoring-relevant attribute	Attribute domain	Reference attribute	
			Weight	Cardinality
VulnerabilityExploitationCase				
	startingDate	Date (integer)	1	1
	endingDate	Date (integer)	2	1
	minAssetApplicationEndingDate	Date (integer)	2	1
	maxAssetApplicationEndingDate	Date (integer)	2	1
	performedBy	ThreatGroup	3	1
	directingAgent	ThreatIndividualEC	2	1
	deliberateActors	ThreatIndividualEC	1	1+
	targetInExploitation	ExploitationTarget	5	1
	modelInExploitation	VulnerabilityMode	4	1
ThreatGroup				
	exploitsVulnerabilities	VulnerabilityMode	1	1+
	memberAgents	ThreatIndividualEC	1	1+
ThreatIndividualEC				
	hasMember	ThreatIndividual	1	1+
ThreatIndividual				
ExploitationTarget				
VulnerabilityMode				
	modeCapabilities	Capability		1+
	modeResourceTypes	ResourceType		1+
Capability				
ResourceType				

ever practical minimum-estimated precision threshold a detector may set. Also, the universe of potential—*e.g.*, syntactically admissible—hypotheses for a given dataset is practically unbounded, rather than limited to presented items as in the traditional case, so that precision scores may be dominated by a practically arbitrary number of false-positive responses (making guessing a practically ineffective strategy).

The size of the information fusion hypothesis space also presents issues for some other metrics commonly used in performance analysis of (binary) classifiers. In particular, the so-called "false-positive rate" used in receiver operator characteristic (ROC) curves and in the calculation of the "detection capability" metric of Gu *et al.* [16][1] assumes a practically enumerated set of "true-negative" responses (*i.e.*, the presented items known in ground truth to be non-threat). True negatives also must be enumerated for the machine learning community's commonly used "accuracy" metric (the number of true—positive and negative—responses divided by the number of all—true and false—responses). Schrag and Takikawa [35] describe analogies between our hypothesis scoring approach and binary classification.

3.2. Scored Object Types and Attributes

Table I presents the types and attributes that are considered during scoring in the counter-terrorism domain, per the artificial world's representation. Along with each attribute is specified its domain, scoring weight (reflecting the challenge problem developer's intuition of an attribute's importance), and reference attribute cardinality (either single or multiple).

The first three types in Table I are just the scored case types. The remaining types are those that appear as values of scored attributes of cases (*i.e.*, as subcases) or in turn as values of the subcases' attributes.

For each type, each instance also has a unique identifier (UID) by which it may be referred to. An object of class ThreatIndividualEC is used to represent an equivalence class (EC) of threat individual identities, supporting aliases. In attribute values, we interpret any of an EC's member UIDs as denoting the full EC.

Note that an instance of the exploit-target event generation pattern in Fig. 4 is represented in Table I only at the top level (the VulnerabilityExploitationCase threat event type). To simplify scoring, we have engineered this object type to include relevant attributes that it might not explicitly have otherwise—*e.g.*, minAssetApplicationEndingDate, determined by comparing occurrence dates of (lower-level) events associated with the apply-resource and apply-capability patterns.

Note also that event objects have as attribute values objects of other types, some of which also are scored. We rely on the fact that our counter-terrorism domain's supercase type-to-subcase (whole-to-part) type graph (depicted in Fig. 16) is directed and acyclic, as we compute scores for leaf types first and work our way up to root types. (In Fig. 16, only the object types requiring case pairing are shown.)

[1]The authors, working in the domain of computer network intrusion detection, describe an "intrusion detection capability" metric which is in fact applicable in any binary classification setting.

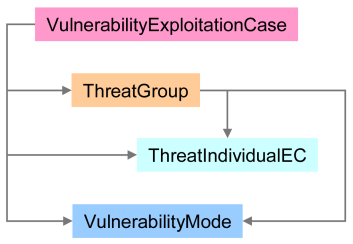


Fig. 16. Counter-terrorism domain supercase type-to-subcase type graph.

3.3. Algorithmic Details of Case Comparison

We compare two like-type cases to determine their object-oriented precision $\mathcal{P}$ and object-oriented recall $\mathcal{R}$, as follows.

We treat a case as a set of assertions regarding the case's attribute values—*e.g.*, (hasMembers Group-4931 Individual-2437). Note that a given case can have multiple, distinct assertions pertaining to a given (multi-valued) attribute. *e.g.*, Group-4931 can have more than one member. Note also that the reference and hypothesized cases can have different numbers of assertions and of attributes, depending on numbers of values per attribute reported by the reference and by the hypothesis.

For each case type, for each defined attribute, a case scoring specification indicates an assertion weight, as summarized in Table I. For a given attribute, the same weights are used for assertions of hypothesized as for those of reference cases.

For a given reference case with the set of assertions $\{r_1, r_2, \ldots, r_m\}$ and corresponding set of weights $\{w_1, w_2, \ldots, w_m\}$, we define the "object-oriented recall basis" $\mathcal{R}_b = \sum_{(i=1\ldots m)} w_i$. (So, each weight is counted once for each assertion in which the attribute appears.) For a given hypothesized case with the set of assertions $\{h_1, h_2, \ldots, h_n\}$ and corresponding set of weights $\{w_1, w_2, \ldots, w_n\}$, we similarly define the "object-oriented precision basis" $\mathcal{P}_b = \sum_{(j=1\ldots n)} w_j$. Note that, for a given comparison of two cases, $\mathcal{R}_b$ and $\mathcal{P}_b$ may differ depending on numbers of values per attribute reported by the reference and by the hypothesis.

We pair reference and hypothesis attribute assertions one-to-one, computing for each pair (r_i, h_i) the following (per the next section's rules for assertion comparison).

- Object-oriented recall $\mathcal{R}(r_i, h_i)$
- Object-oriented precision $\mathcal{P}(h_j, r_j)$

We define the "object-oriented recall contribution" $\mathcal{R}_c$ as the sum over the hypothesized case's assertions of assertion weight w_i pro-rated by the corresponding recall—$\mathcal{R}_c = \sum_{(i=1\ldots n)} \mathcal{R}(r_i, h_i) * w_i$. The "object-oriented precision contribution" $\mathcal{P}_c$ is the sum over the reference case's assertions of assertion weight w_j pro-rated by the corresponding precision—$\mathcal{P}_c = \sum_{(j=1\ldots m)} \mathcal{P}(h_j, r_j) * w_j$.

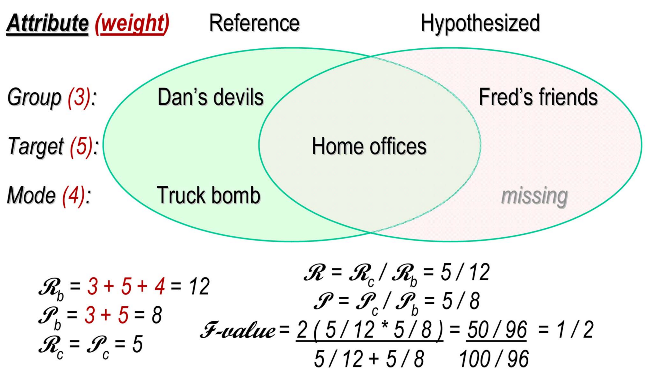

Attribute (weight)
Reference
Hypothesized
Group (3):
Target (5):
Mode (4):
Dan's devils
Truck bomb
Home offices
Fred's friends
missing
$\mathcal{R}_b = 3 + 5 + 4 = 12$
$\mathcal{P}_b = 3 + 5 = 8$
$\mathcal{R}_c = \mathcal{P}_c = 5$
$\mathcal{R} = \mathcal{R}_c / \mathcal{R}_b = 5/12$
$\mathcal{P} = \mathcal{P}_c / \mathcal{P}_b = 5/8$
$\mathcal{F}\text{-value} = \frac{2(5/12 * 5/8)}{5/12 + 5/8} = \frac{50/96}{100/96} = 1/2$

Fig. 17. Object-oriented metrics with non-uniform attribute weights.

For a given pair of reference and hypothesized cases, we define the following.

$$\mathcal{R} = \mathcal{R}_c / \mathcal{R}_b$$

$$\mathcal{P} = \mathcal{P}_c / \mathcal{P}_b$$

$$\mathcal{F}\text{-value} = 2(\mathcal{P} * \mathcal{R}) / (\mathcal{P} + \mathcal{R}).$$

We compute the metrics for a given dataset's cases of a given type as follows. Let N_R be the number of reference cases and N_H the number of hypothesized cases. Let the set $\{p_1, p_2, \ldots, p_o\}$ be the computed pairs, and $\mathcal{R}(p_k), \mathcal{P}(p_k)$ the object-oriented recall and precision (respectively) of the kth pair. Then for the dataset we have the following.

$$\mathcal{R} = \left(\sum_{(k=1\ldots o)} \mathcal{R}(p_k) \right) \Big/ N_R$$

$$\mathcal{P} = \left(\sum_{(k=1\ldots o)} \mathcal{P}(p_k) \right) \Big/ N_H.$$

Fig. 17 adds non-uniform attribute weights to the example of Fig. 9 to illustrate their use in the object-oriented metrics.

The metrics' sensitivities to specified attribute weights depend on the numbers of values for each attribute in compared cases (so, how many times each weight is counted) and—for nested objects—on weights applied in supercases.

3.4. Pairing and Comparison for Attribute Values

We require that paired assertions have the same attribute, so single-valued attributes pair straightforwardly and multi-valued attributes require a one-to-one pairing over their values. In principle, the values may be scalars (of type, *e.g.*, Date) or structured objects (cases or other objects—*e.g.*, Targets) that have UIDs. Per Table I, we have no multi-valued attributes with scalar values in the counter-terrorism domain. Thus, we emphasize here the pairing of multi-valued attribute assertions with nested object values.

We have two alternative general methods for pairing multi-valued attribute assertions.

- Rely on a global one-to-one pairing between all the hypothesized and reference instances of the nested case types for any multi-valued attributes. This has the advantage of consistent pairing across all contexts (because the same hypothesized case is always paired with the same reference case).
- Compute a local one-to-one pairing addressing just the context of a given candidate pair of hypothesized and reference cases and their attribute values. This has the advantage of optimizing per-candidate pair scores—at the expense of global consistency (because different hypothesized cases may be paired with different reference cases in different contexts). Local pairing over nested cases with multi-valued attributes might be prohibitively expensive, but these do not arise in our counter-terrorism domain.

Once pairs have been established over the candidate hypothesized and reference objects' attributes, we can read off each pair's object-oriented precision and recall. We have two alternative general methods of doing this.

- Interpret the computed comparison scores smoothly (accept them at face value)—wherever they fall in the interval $[0,1]$. Smooth comparison reflects the combined matching quality of a given case and all of its nested subcases.
- Interpret the selected pairing crisply, by returning one—for both precision and recall—if the hypothesized object has been paired with a reference object, zero otherwise. Thus, scores fall in the set $\{0,1\}$. Crisp attribute comparison minimizes the impact of inexact matching of nested subcases. The crisp setting matters when the F-value threshold for case pairing is zero. Under non-zero thresholding, comparison is always "crisp" (using traditional metrics, as noted in Section 3.1).

Of the object types with attributes considered during scoring, instances only of ThreatIndividualEC appear as values of multi-valued attributes in other types (VulnerabilityExploitationCase and ThreatGroup). So, in our counter-terrorism domain, only instances of type ThreatIndividualEC may require local case pairing. For these instances, a pair's object-oriented precision and recall depend on how we handle any aliases.

- If both the reference and hypothesized ECs are singleton (perhaps because the dataset does not invoke aliasing), the pair's object-oriented precision and recall are both one if the UIDs match, otherwise both zero.
- Otherwise (at least one of the reference and hypothesized ECs is not singleton), we have two choices.
  —Apply smooth or crisp comparison to the globally or locally computed pairing of reference and hypothesis ThreatIndividualECs.
  —Anti-alias by appeal to the ECs defined in ground truth, then (after discarding any resulting duplicate assertions) score as for singletons. This may be appropriate when LD does not have access to an alias detection capability.

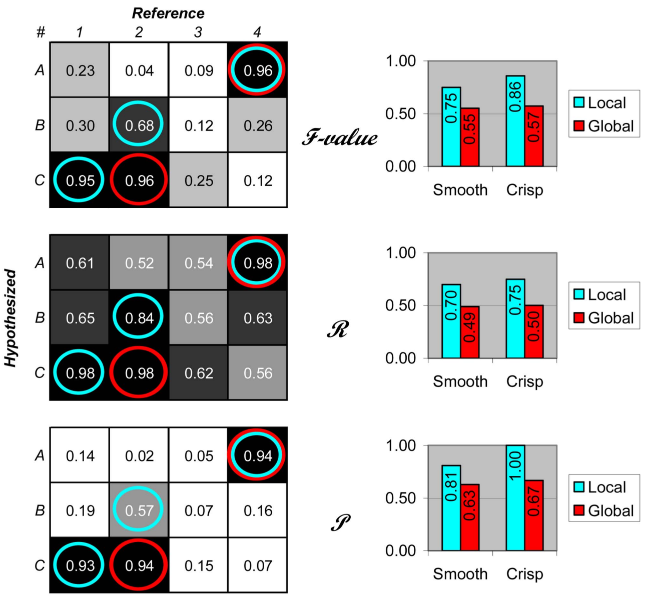


Fig. 18. Alternative pairings (local and global) and interpretations (smooth and crisp) for objects (of type ThreatGroup) with object-valued attributes (of type ThreatIndividualEC).

To illustrate the above concepts, suppose that Fig. 12 compares objects of type ThreatIndividualEC. Suppose that we would like to compare two ThreatGroups and that we have for the present zeroed out the scoring weight for the exploitsVulnerabilities attribute. Then only the memberAgents attribute counts; suppose for the compared groups it has the following values.

- Reference: {EC-1, EC-2, EC-3, EC-4}
- Hypothesized: {EC-A, EC-B, EC-C}

We have reproduced the relevant portions of Fig. 12, Fig. 13, and Fig. 14 in Fig. 18. The global pairing is given by the larger, red circles, the local pairing by the smaller, cyan ones.

Of the types in Table I, the following either have no attributes or have no attributes that are considered during scoring.

- Instances of type Date are represented by integers. The object-oriented precision and recall for a pair of reference and hypothesized Dates are both (identically) defined in terms of their normalized temporal distance, as illustrated in Fig. 19. (The "ratio" parameter is computed with respect to a nominal distance specified for the given attribute.)
- The types Target, Capability, and ResourceType have no scored attributes. If a pair's UIDs are the same, the object-oriented precision and recall are both one, otherwise both zero.
- For the type VulnerabilityMode, we require strict set equivalence. For each of the two multi-valued attributes, if a pair's respective sets of attribute values

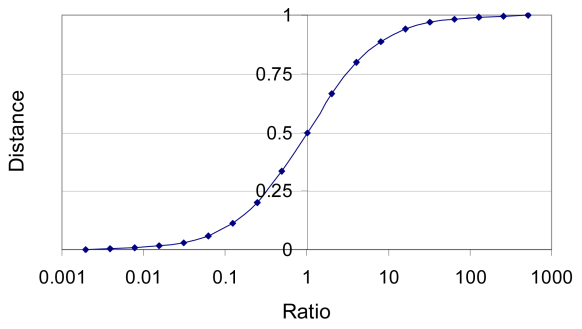


Fig. 19. A normalization function for Date distances.

have the same members, the pair's object-oriented precision and recall are both one, otherwise both zero.

3.5. Additional Hypothesis Scoring Capabilities

The PE Lab also implements many-to-one and many-to-many case pairing methods that—unlike their one-to-one counterpart—do not necessarily penalize LD for submitting multiple competing hypotheses. We have considered (but not yet implemented) support for logical and probabilistic uncertainty. One practical approach may be for LD to submit disjunctive hypotheses with each disjunct's probability noted. Entire disjunctions would then be paired one-to-one with reference cases.

Because we include some (partially described, potentially corrupted) case objects in evidence, we would like a scoring method to factor this information out of reported scores. We score evidence's case objects directly (treating them as hypotheses), to establish a baseline, as described in Section 5.4. To obtain more diagnostic value, we are contemplating a refinement to develop separate hypothesis scores regarding reference case content that is:

- Correct in evidence;
- Clearly incorrect (corrupted) in evidence;
- Ambiguous in evidence; or
- Omitted from evidence.

Our hypothesis scorer's many parameters (only some of which have been mentioned here) support customized experimentation. For example, we have on occasion (as in Section 3.4's illustration) tailored case scoring specifications to zero out the weights of attributes that a given technology does not address. Another parameter lets us count the weight of each multi-valued attribute only once for a given object instance, rather than counting the attribute's weight time it appears in the object. Counting just once is appropriate when attributes' relative imports are roughly independent of their per-instance cardinalities, which agreed with our intuition when we applied the overall PE lab approach in the computer network intrusion detection information fusion domain [34].

We are considering the following alternative treatment of temporal information (*e.g.*, events' endpoints)

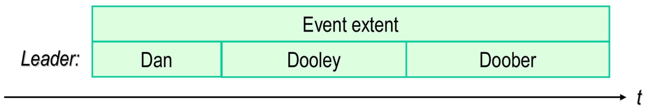


Fig. 20. Ground truth representation of a time-varying attribute.

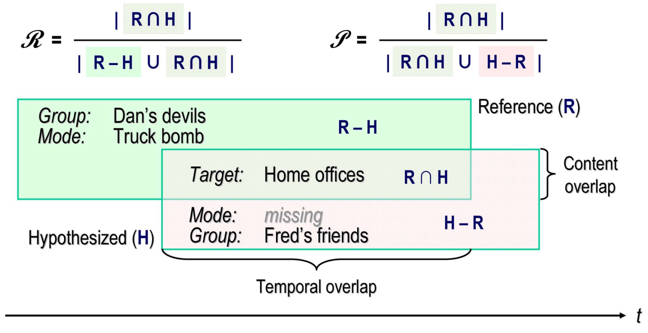


Fig. 21. Temporal object-oriented recall and precision.

to accommodate scored objects with time-varying properties that may be included in future versions of the artificial world or that may arise in other application domains—*e.g.*, groups with time-varying membership (memberAgents) or events during which an exploitation's leader (directingAgent) role may alternate among team members. Instead of treating an event's endpoints as individual scalar attributes (startingDate and endingDate), we would record, for each object attribute with temporal extent, a set of contiguous temporal intervals over which its potentially different values hold. In ground truth, for each single-valued attribute there would be a set of disjoint, adjacent intervals that together cover the object's extent, as in Fig. 20.

Fig. 21 suggests how we would compute temporal object-oriented precision and recall for the reference and hypothesized threat events shown in Fig. 9. (For simplicity, Fig. 21 omits the weights of Fig. 17. The fact that our current threat events have static attributes also simplifies the illustration, in that we require only a single temporal interval per event.)

Note that scores for these temporal versions of object-oriented recall and precision are zero whenever the intervals do not overlap, which means we can omit any non-overlapping pairs from the case pairing matrix without excluding any pairs with non-zero comparison scores.

Under some circumstances, it might make sense to compare temporal objects with attributes summarized at different levels of temporal abstraction; thus two objects' coarse time structures could be seen to match relatively well even when their fine time structures did not.

3.6. Related Work (Hypothesis Scoring)

Our object-oriented metrics may be compared with other metrics of inexact matching, such as the graph edit distance metric used in LAW [44]. Some of the key differences between this approach and ours are:

- Our strong object orientation *versus* their accommodation of arbitrary relationship graphs;
- Our separate tracking of false-positive and false-negative discrepancies *versus* their uniform distance tracking; and
- Their accommodation of ontological distances between typed nodes that we haven't required (because our ontology of case- and scoring-relevant attribute types in Table I includes no subtype relationships).

Our reliance on a directed acyclic graph of subcase types means that (given global pairing of multi-valued attributes) we compare two cases of a given type only once, regardless of how interconnected the different-type hypotheses may be. It also allows us to deprecate the fine differences between nested cases in comparison to the coarser differences at the supercase level. They also take advantage of graphs' hierarchical structure and cache computed subgraph distances locally, but not globally as we do. Ghallager [15] surveys additional graph-based pattern matching metrics.

Working in the computer network intrusion detection domain, Tadda *et al.* [39] have adopted our unweighted object-oriented metrics and our one-to-one case pairing (with recall, rather than F-value, used in case comparison matrices) and have developed some original metrics—inspired by metrics in the field of target tracking—to summarize a resulting case pairing matrix. This is for a single-level case structure where each attribute value is assumed to appear in no more than one hypothesis.

To support scoring in the network intrusion detection PE Lab, Schrag and Takikawa [34] developed object-oriented precision and recall scores at different levels by defining different case types corresponding to different abstractions (information-reducing mappings) of a full-information case type accommodating hypotheses from the technology under test. This requires separate case pairings for the different case types but supports scoring the abstractions using common metrics. Precision for the most-reduced case type, a bag of evidence, corresponds to the "track purity" metric used in target tracking.

Mahoney *et al.* [26] describe scoring for a two-level military situation hypothesis that includes a joint probability distribution (*e.g.*, a Bayesian network) regarding component objects' existence and attribute values. They compute (un-normalized) attribute value distances at the situation component level, accounting for hypothesized probabilities and applying attribute weights. At the situation level, they invoke a distance threshold to qualify potential matches of hypothesized and reference components, then develop all possible sets of one-to-one pairs and estimate the likelihood of the observed distance given that the pair is a correct match and given that the pair does not match. By aggregating the likelihoods computed for a given pair across the sets, they estimate the overall likelihood for each pair. It is not clear whether this work could be generalized easily to address more deeply nested hypotheses.

4. ALERT SCORING METHODS

The scoring methods of Section 3 allow us to compare, for a given scored object type, a static set of hypothesized objects against a static set of reference objects. They thus are appropriate for scoring a batch of event hypotheses tendered retrospectively/forensically. By themselves, though, they are inadequate for scoring *alerts*—event hypotheses tendered prospectively, for warning, when LD incrementally processes time-stamped event records in evidence. Here, we describe methods that additionally account for alert scoring's dynamics—whether the alerts in effect during reference events are good hypotheses and whether they can be used to support effective warning.

Schrag *et al.* [36] describe an approach that was implemented late in the development of PE Lab and suggest an alternative practical approach and an associated (deemed-impractical) idealized approach. All three of these variants rely on specified costs of false-positive and false-negative reporting that are applied uniformly over a portion of the reference event's temporal extent. As explained further in Section 4.2, they also conflate evaluation of the quality of a prospectively tendered event hypothesis and of the decision about whether or not to tender it, in that they presume some mitigating action will be attempted for all tendered alerts. Recognizing the potential value of pure hypothesis quality evaluation and of more sophisticated models for cost and response, we here sketch two alternative, complementary approaches.

- Section 4.1 describes a cost-free approach based on the temporal variant of object-oriented precision and recall described in Section 3.5.
- Section 4.2 describes an alert-free, cost-based approach that excludes precision and recall because it doesn't even require the technology under test to present alerts, rather merely to invoke response actions when it considers this advantageous with reference to a furnished cost model.
- Section 4.3 describes related work.

4.1. Cost-free Alert Scoring

As LD advances incrementally through time-stamped evidence, it must examine only the events that are reported (with respect to simulation time) either at or before its current processing time. LD may submit alerts at any (simulation) time, with any frequency it chooses. Finer incremental processing time intervals may incur greater overall processing time but also may afford more opportunity to detect impending threat events and issue alerts sooner. Coarser intervals may not pick up evidence regarding threat events until after they have been consummated—and so miss the opportunity for alerting. Requiring alerts at fixed simulation-time frequencies would entail similar issues. Requiring alerts at specified simulation times (*e.g.*, near threat event consummations) might engender gaming by LD.

LD may supersede an alert tendered earlier in simulation time with one tendered later, or it may retract an alert without superseding it. We call an alert that has been tendered and that has been neither superseded nor retracted an "active" alert. The temporal scores we've described in Section 3.5 for retrospectively tendered event hypotheses can be construed as applying to the hypotheses active at the end of simulation time. For any given simulation timepoint, we could make a similar comparison of the active alerts—either to the full set of reference events or just to those that have started. If we averaged the resulting scores across timepoints, though, earlier events that were considered in more per-timepoint scores would have disproportionate effect. We propose instead to consider all simulation timepoints simultaneously, admitting an alert as a candidate for pairing with a reference event based on its activity status (*i.e.*, whether it is active or not) at an anchor point in the reference when it might reasonably last be considered useful—the first time one of its capability or resource assets is applied to the target (noted in the attribute minAssetApplicationEndingDate) and after which we may consider the attack's success to be inexorable.

While we have thus shifted the focus from individual simulation timepoints to the reference events themselves, we'd like to go a bit further and reflect how the quality of alerting has evolved over the course of each reference event (*i.e.*, not just at one anchor point). Given unlimited computing power, we might compute the per-anchor point scores using as anchors all earlier points in the reference event (*i.e.*, considering just the alerts active at each point). While this may be feasible when there are few alerts, for more general practicality we must limit our invocation of the expensive case pairing operation. We propose to do so by taking the case pairing that is computed for the reference cases' minAssetApplicationEndingDates as authoritative and by chaining backward from the alerts paired there to any alerts they have superseded.[2] Finally, we can take the average values we have computed for temporal precision and recall (or for temporal F-value) over each reference case.

In the example of Fig. 22, Alert 3 tendered at Simulation Time 3 supersedes Alert 2 tendered at Time 2. Alert 4 (not shown) is tendered past Anchor Point A

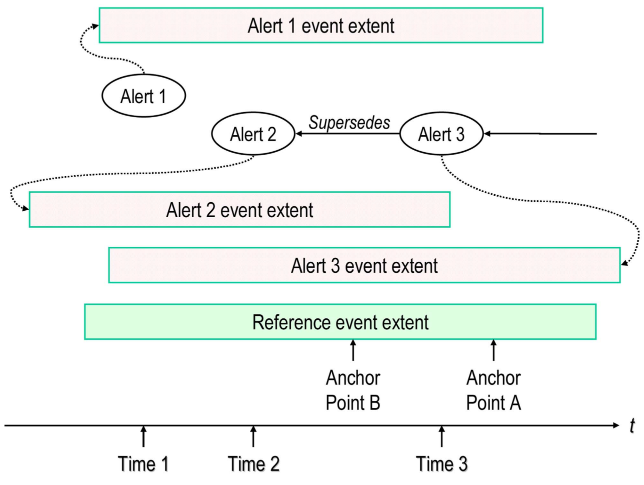


Fig. 22. Alerts in a supersession chain.

corresponding to the minAssetApplicationEndingDate, so is not eligible for comparison to this reference event. Both Alert 1 and Alert 3 are active with respect to Anchor Point A; Alert 2, the better match, is paired, and the alert supersession chain it heads becomes authoritative over the rest of the reference event. Even though Alert 1 is a better match at Anchor Point B than Alert 2, Alert 1 is not considered. The overall score computed for this reference event thus is based on Alert 3 from Time 3 to the event's end, on Alert 2 from Time 2 to Time 3, and on no alert before Time 2.

Under this treatment of supersession chains, LD should supersede one alert with another when it believes they apply to the same event. Otherwise, it should retract the earlier alert and start a new chain. We require supersession chains to be non-branching. Backward branching would introduce ambiguity as we chain backward. Forward branching might make two reference intervals end up at the same alert—if the heads of their respective chains were paired with different reference cases.

4.2. Alert-free Response Action Scoring

In the earlier approaches [36], we specified costs for false-positive and false-negative predictions of an attack on a target, indicating (respectively) the costs of actions taken in response to a false prediction (*e.g.*, escalated protection, evacuation) and of inaction resulting from a non-prediction (destruction of target, loss of life). We applied these costs uniformly over the portion of the reference event's temporal extent preceding the minAssetApplicationEndingDate and discounted them to reward good hypotheses—discounting the false-positive cost by object-oriented precision and the false-negative cost by object-oriented recall. See Fig. 23, whose dark-shaded fraction corresponds to the discount and light-shaded fraction to the assessed cost.

In the real world, the costs of action and of inaction depend on complex interactions. Even in our artificial world, it's hard to tell a consistent story (in terms

[2]The earlier practical and implemented approaches (Schrag *et al.* 2006) chained backward over supersession links established as authoritative by pairing reference cases with retrospectively tendered hypotheses. While this affords more perspective for LD, it might (especially since some reference events become visible in evidence after they are complete) be considered artificial.

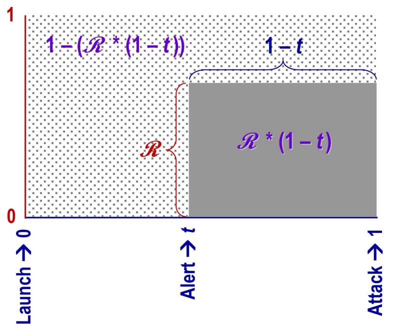


Fig. 23. Discounting uniform false-negative cost by object-oriented recall (in earlier approaches).

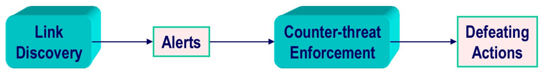


Fig. 24. Processes in automated threat defeat.

of actions and effects) about why false-negative and false-positive costs should be applied symmetrically and uniformly over time.

More fundamentally, our funding program's scope was limited to an LD component tendering threat hypotheses (see Fig. 1), not a "counter-threat enforcement" component issuing actions to interdict threat agents or to defeat their attacks (see Fig. 24). Imposing an action interpretation on LD's prospective event hypotheses effectively required our evaluation participants to conflate enforcement with LD and to make decisions about whether to issue alerts at all.

In hindsight, we recommend that cost-based scores be developed following an approach closer to that taken by Morrison *et al.* [27] in the Hats simulator, where an agent with threat detection capability (like that of LD) acts as a player in a game. The player agent must act on whatever threat event hypotheses it develops to interdict suspected or known threat actors before they inflict damage. Performance is determined by the game's final score that accounts for incurred costs associated with things like surveillance, damage, and false arrest. Because the player agent is included in the Hats simulation, its actions can have downstream effects as rich as the simulation supports. For example, an incarcerated agent will not participate in future attacks.

The simulator currently underlying PE Lab's dataset generator supports no such enforcement actions. One reason we wanted datasets that could be processed entirely off-line was to compare easily performance results for a given dataset across different technologies under test, without the technologies' different sets of enforcement actions resulting in different simulation histories (effectively making the datasets different and the technologies' scores incomparable). The closest we can come to Hats' style of cost-based performance evaluation and still maintain dataset-based comparability is to admit "hypothetical" actions and effects that do not actually manifest in simulation history. Instead, we propose to infer tendered actions' (determined or likely) effects and (determined or expected) costs according to a model that we furnish. Such a hypothetical action approach also might be used to evaluate an enforcement component's performance against real-world historical datasets, which inherently require off-line processing.

TABLE II
Attack Effects and Eosts

Attack Effect	One-time cost
Good individual dies.	High
Target is destroyed.	Very High

TABLE III
Hypothetical Counter-Threat Enforcement Actions, Effects, and Costs

Enforcement Action	Unit-time cost	Latency	Effect
Escalate target security.	Low	Short	CE can detain, evacuate.
Detain individual.	Medium	Nil	*Role-dependent…*
Exploitation team leader:			Exploitation is abandoned.
Half of exploitation team members:			
Evacuate target.	High	Medium	Nobody good at target dies.

To illustrate how this approach might be realized in our artificial world, Table II suggests some notional one-time costs to be associated with a successful attack. Table III suggests continuing (per-time unit) costs that a counter-threat enforcement (CE) component could incur in its actions to defeat attacks.

Table III also suggests time constraints. Details follow.

- Upon invoking the action to escalate security at a target, there is a short "security escalation period" (latency). Thereafter, CE can detain (effectively incarcerate) individuals at that target and can evacuate it.
- The action to evacuate a target must be invoked a medium time before the attack's last resource or capability is applied (maxAssetApplicationEndingDate) to avoid the deaths of visitors (everyone whose last visit was to the target, including attackers') when the target is destroyed.
- CE may choose to detain (*e.g.*) suspected threat individuals or any individual it can determine may possess a resource or capability supporting a suspected-impending threat mode. The exploitation team leader or at least half its members must be detained throughout the asset application interval (from minAssetApplicationEndingDate to maxAssetApplicationEndingDate) to defeat the attack. We could (as in Hats) accord greater cost to detaining a non-threat individual.

Fig. 25 illustrates CE's hypothetical actions to defeat a specific attack, in a scenario where CE is able to detain three attackers but doesn't have confidence that it can thwart the attack. It decides to evacuate and is able to do so in time.

Fig. 26 illustrates (not to scale) cumulative costs associated with the enforcement actions and with the attack in Fig. 25. In the described scenario, costs in the

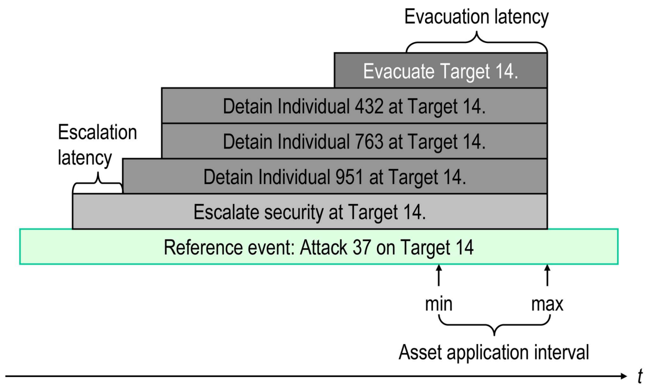

Evacuation latency
Evacuate Target 14.
Detain Individual 432 at Target 14.
Escalation latency
Detain Individual 763 at Target 14.
Detain Individual 951 at Target 14.
Escalate security at Target 14.
Reference event: Attack 37 on Target 14
min
max
Asset application interval
t

Fig. 25. Actions to defeat a specific attack.

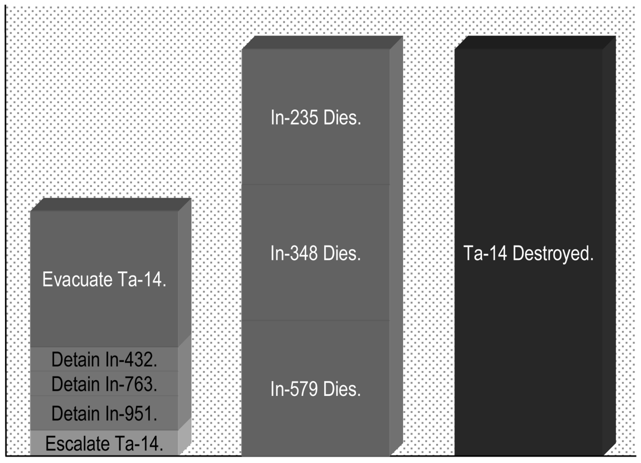


Fig. 26. Costs of hypothetical enforcement (left), enforcement-defeated loss of life (center), and attack damage (right).

center column are averted; those in the left and right columns are incurred.

Precise scalar costs would need to be tuned relative to each other to ensure that datasets would pose reasonable challenges. *e.g.*, we generally would want CE to incur a higher cost for incarcerating everyone it ever sees than it incurs for doing nothing. A limited cost budget (*e.g.*, one that would support only a limited number of concurrent individual detentions or of total target evacuation days) also could help to ensure plausible CE threat-defeating strategies.

To facilitate comparison across datasets, scores may be normalized in each dataset to the cost that would be incurred if CE were to do nothing (and all attacks were successful).

4.3. Related Work (Alert Scoring)

Besides Hats, related work appears to be limited. Others have evaluated event prediction where exact hypothesized-to-reference case matching is appropriate. Weiss and Hirsh [42] specialize precision to discount temporally close false-positive hypotheses. Létourneau, Famili, and Matwin [20] apply a nonmonotonic timeliness function to determine rewards and penalties for true- and false-positive predictions of appropriate aircraft component replacement times. Different metrics are certainly appropriate in different contexts, and we believe the accommodation of inexactly matching hypothesized and reference cases and attendant case pairing entail issues for structured threat alert scoring that others have not addressed. Mahoney *et al.* [26] suggest some overall strategies for comparing hypothesized *versus* reference situation histories and note the requirement for timeliness, without directly addressing threat event prediction.

TABLE IV
Coarse Problem Space Dimensions

Group Connectivity	How many groups an individual belongs to
Noise, Clutter	How much threat masking
Dataset Size	How many observable transactions
Population Size	How many individuals
Pattern Complexity	Minimalistic *vs.* richer threat event modeling
Observability	How likely observations are
Corruption	How corrupted observations are
Aliasing	How frequently aliases are used
Event Confusability	How like are threat and non-threat activities
Target Duty Level	How busy targets are
Individual Duty Level	How busy individuals are

5. EXPERIMENT DESIGN TO IDENTIFY PERFORMANCE INFLUENCES IN THE PROBLEM SPACE

We now describe our methodology for constraint-based experiment design and analysis to identify which among varied dataset characteristics most influence a given technology's performance on a given detection task. We illustrate this methodology with a case study using object-oriented F-value as the performance metric of interest. We describe our experimental approach, summarized as follows, in subsequent subsections.

1. Collapse many (fine) problem space parameters into a few dimensions with discrete (coarse) difficulty settings (Section 5.1).
2. Specify a mix of experimental datasets that maximizes diversity over the difficulty settings (Section 5.2).
3. Exercise participating detection technology configurations over datasets in the mix (Section 5.3).
4. Score technologies' output hypotheses relative to an baseline derived from evidence (Section 5.4).
5. Determine the statistical significance of apparent problem space performance influences by technology and detection objective (Section 5.5).

Section 5.6 describes some related experimental design work.

5.1. Problem Space Discretization

The coarse problem space dimensions are summarized in Table IV.

Each coarse dimension corresponds to one or more fine parameters. For some dimensions, we discretize the fine parameters based on quantitative annual or

TABLE V
Fine Parameter Discretizations by Problem Difficulty

Group Connectivity	None		Easy		Fair		Hard	
Individual status:	Threat	Non-threat	Threat	Non-threat	Threat	Non-threat	Threat	Non-threat
Mean groups per individual	1	1	2	4	4	6	6	10
Dev. groups per individual	0	0	1	2	2	3	3	4

TABLE VI
Fine Parameter Discretizations per Annual Performance Goals

Population Size	Y1	Y2.5	Y3
Number of individuals	~1,000	~10,000	~100,000
Mean threat group membership	20	80	80
Dev. threat group membership	5	20	20
Number of capabilities	50	100	150
Number of resources	50	100	150

Dataset Size	Y1	Y2.5	Y3
Number of observable transactions	N / A	~100,000	~1,000,000

Noise, Clutter	Y1	Y2.5	Y3
Threat-to-clutter event ratio	0.08	0.008	0.0008
Structured event SNR	0.08	0.008	0.0008
Transaction event SNR	0.08	0.008	0.0008
Individual SNR	0.4	0.08	0.008
Group SNR	0.8	0.16	0.016

semi-annual performance goals (set by the funding program)—see Table VI. For other dimensions we chose to explore, we discretize into difficulty settings such as Easy, Fair, Hard—see Table V for an example. We apply a stop-light color-coding over the discretized settings, adding light green for very easy settings and dark red for very hard.

5.2. Dataset Mix Specification

Several factors make effective experimentation challenging in this context. The evaluation dataset mix is scoped to occupy a few solid weeks of coordinated program effort. Processing is not always hands-off, with several disparate component developers sometimes manually handling intermediate results within a single technology configuration. A star-shaped experimental design with fixed baseline settings and single-dimension departures might serve individual technology configurations with single detection objectives, but—with each dataset—we must test multiple configurations over multiple objectives. What's easier for one technology/objective combination might be harder for another. At evaluation time, we have somewhat sparse prior performance data from dry-run activities. We need an experiment that effectively tests over multiple baselines simultaneously, so we choose a diversity-maximizing, fractional factorial design.

We take the following steps, discussed below, to maximize diversity.

1. Specify cross-dimension settings constraints that ensure well dataset generation.

TABLE VII
Cross-Parameter Constraints

Noise, Clutter	Dataset Size	Population Size	Pattern Complexity	Rationale for prohibited combinations of settings listed in rows
	Y2.5	Y3		People doing too few things during a simulation
Y3	Y2.5			Too few threat events (maybe none)
Y1	Y3		Thin	Too many threat events to score practically
Y3	Y3		Fat	Too few threat events (maybe none)
Y2.5	Y2.5		Fat	
	Y3	Y2.5	Fat	Too many time ticks for incremental threat detection to be viable (currently)
	Y3	Y1	Fat	
	Y2.5	Y1	Fat	

2. Perform constraint satisfaction to develop an initial dataset mix.

3. Perturb the initial mix in hill-climbing to optimize the experiment's coverage.

Table VII indicates some prohibited coarse setting combinations and associated rationale. *e.g.*, the Fat setting (corresponding to rich threat event modeling—resulting in more atomic transactions per threat event) results in too few threat events when the signal-to-noise ratio used is too low for the dataset size.

Other combinations of coarse settings over these dimensions have been verified to generate well datasets. The coarse discretizations themselves assure compatibilities at the fine parameter level. For example, the various signal-to-noise (threat-to-non-threat) ratios (SNRs) for a given coarse setting in Table VI are coordinated so that there are enough individuals to satisfy the generator's minimum group size requirement. The discretization process thus factors out such fine, numerical constraints (whose violation would raise run-time exceptions), so that coarse constraint satisfaction over symbolic domains is sufficient for the dataset mix specification/experiment design.

The constraint satisfaction problem is challenging in that we want a number of dataset specifications that draw without replacement from settings pools, fixed for each dimension, until all the pools are exhausted. The pool for Group Connectivity, *e.g.*, includes six instances each for the tokens None, Easy, Fair, and Hard. We have implemented an algorithm to specify a dataset mix respecting both the constraints and the pools. Alternatively, if the exact numbers of tokens drawn from each pool is not critical, we can draw with replacement to generate a random dataset specification and discard this if it does not satisfy constraints (or if it is a duplicate), until we have enough datasets.

With an initial mix in hand, we perform a hill-climbing random walk over the space of well datasets, swapping any two datasets' like settings along a given dimension whenever this decreases the maximum number of like settings shared across all datasets.

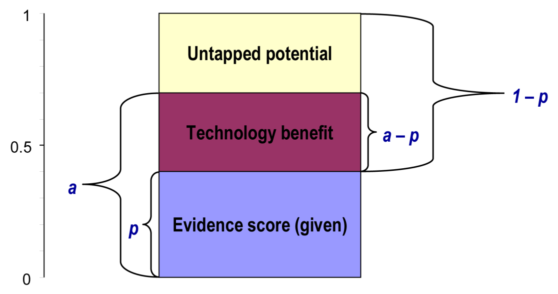


Fig. 27. Relative scoring $r = (a - p)/(1 - p)$.

5.3. Detection Technology Exercise

Technology developers receive the test datasets in database form and are required to return ranked hypotheses in the scorer's input format for each of the detection objectives noted in Fig. 5.

5.4. Adjusting Hypothesis Scores Relative to an Evidence Baseline

To compare dimension influences across different datasets requires comparable scores. As explained below, our default ("absolute") scoring method credits hypothesis content that is patently manifest in datasets to different extents. Comparability requires a ("relative") scoring method that factors this content out.

Evidence provided to LD (as illustrated in Fig. 5) includes partial top-level case descriptions for some instances of the detection object types (threat event, group, individual, and alias association). These descriptions, notionally corresponding to a legacy intelligence database, afford starting places for the detection process. The completeness, consistency, and transparency of these descriptions with respect to ground truth depend on settings for the Observability, Corruption, and Alias dimensions. In absolute scoring, LD gets credit for reporting detection objects whether the same information appears in evidence or not.

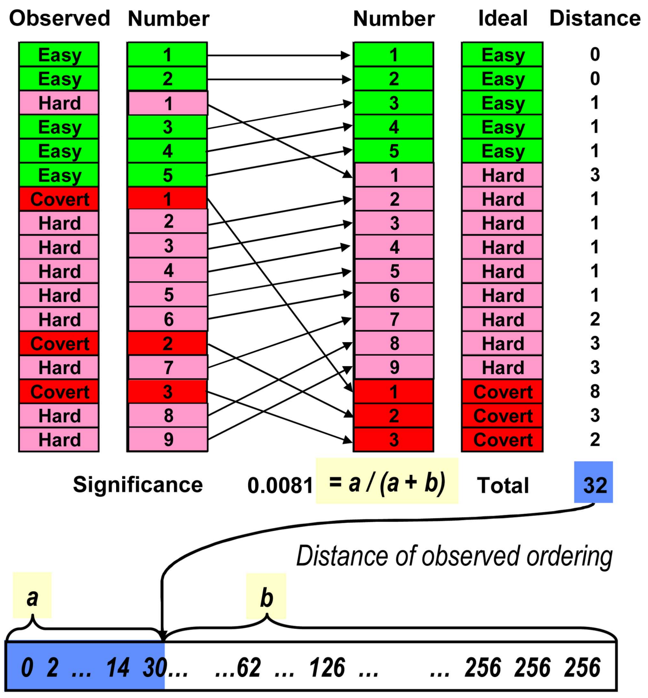
In relative scoring, the detection task may be reinterpreted as, "Find unknown and correct misreported threat objects and their attribute values." Let a stand for LD's absolute score, and let p stand for the score for returning exactly all and only the top-level threat case content provided in evidence. We use p as a baseline in computing the relative score $r = (a - p)/(1 - p)$. See Fig. 27.

Note that r can be negative—if LD does not perform as well as the baseline. Note also that the relative score rewards LD for any improvements to top-level threat case content provided in evidence—for supplying missing attribute values or correcting corrupted ones.

5.5. Identifying Performance Influences

Because of the coarse discretizations and constraints, our experiment design must be "unbalanced" (*i.e.*, have unequal numbers of settings within and across dimensions). This requires us to invent novel techniques to identify performance-influencing dataset characteristics, rather than, *e.g.*, applying ANOVA over coefficient means among regression fits.

TABLE VIII
Ranked Settings Significance Testing



Observed	Number	Number	Ideal	Distance
Easy	1	1	Easy	0
Easy	2	2	Easy	0
Hard	1	3	Easy	1
Easy	3	4	Easy	1
Easy	4	5	Easy	1
Easy	5	1	Hard	3
Covert	1	2	Hard	1
Hard	2	3	Hard	1
Hard	3	4	Hard	1
Hard	4	5	Hard	1
Hard	5	6	Hard	1
Hard	6	7	Hard	2
Covert	2	8	Hard	3
Hard	7	9	Hard	3
Covert	3	1	Covert	8
Hard	8	2	Covert	3
Hard	9	3	Covert	2
Significance	0.0081	= a / (a + b)	Total	32

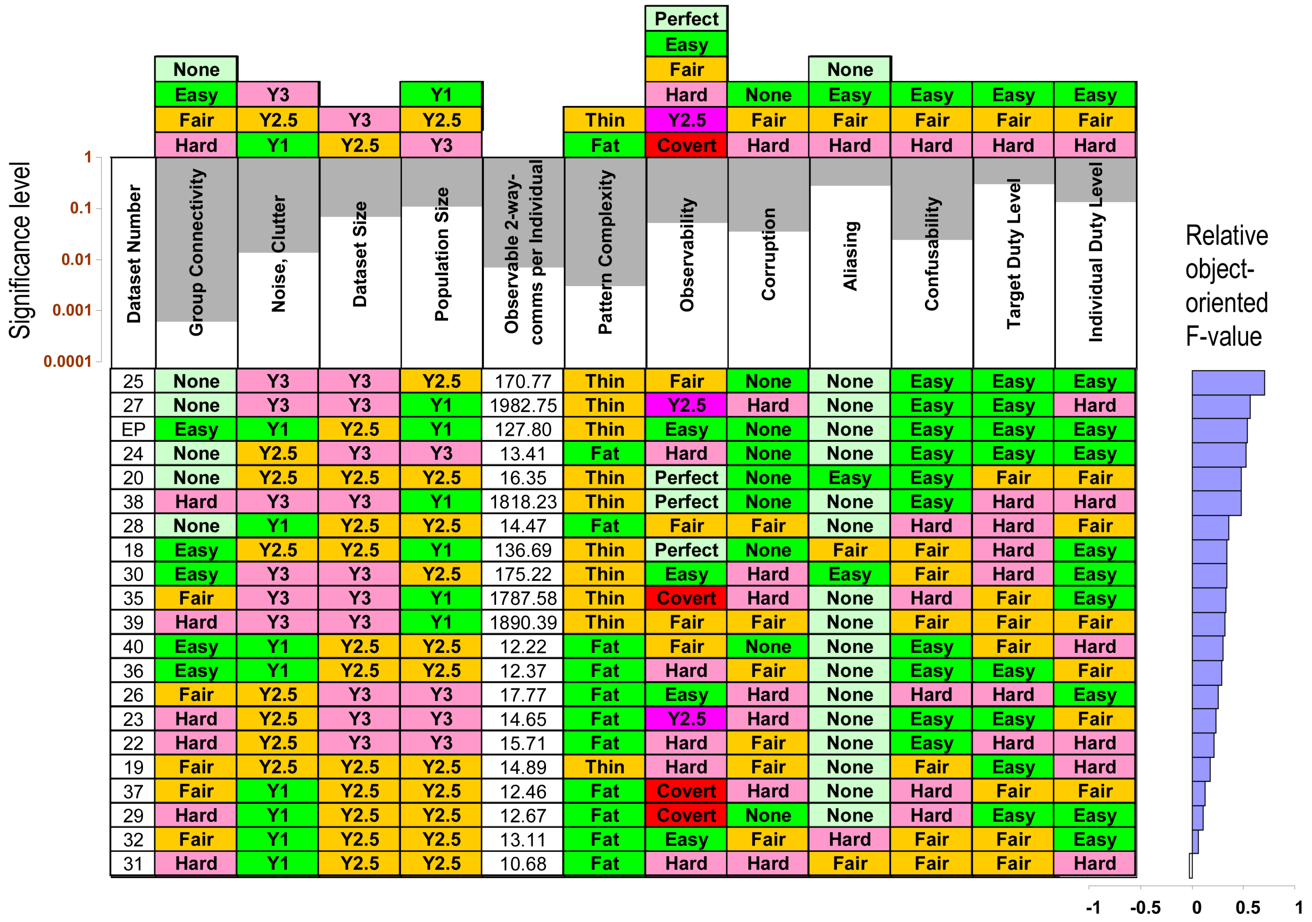
Distance of observed ordering
a
b
0 2 ... 14 30... ...62 ... 126 ... ... 256 256 256

Relative scores support ranking experimental datasets by LD's performance for a given objective. Under this ranking, we expect the settings for a dataset dimension with significant performance influence to tend to exhibit the expected difficulty order—*e.g.*, "Easy, Fair, Hard" or "Y1, Y2.5, Y3." To determine the significance of the settings order actually observed, we first compute its distance to the expected, or "ideal," order, as illustrated in Table VIII. We first number the tokens for each setting (*e.g.*, "Hard") consecutively as they appear in each of the observed and ideal orders. Then, for each so-numbered token, we compute the distance between its ranks in the two orders. Finally, we sum the rank distances for all the setting tokens—yielding in the example an aggregate distance of 32.[3]

To determine the extent to which the observed order is significant with respect to the ideal—the extent to which the observed could have arisen strictly by chance, with lower values indicating greater significance—we similarly compute distances (represented in the abbreviated vector at the bottom of Table VIII) to the ideal from a sufficient number $N = a + b$ of randomly generated token orders, counting the number of times a the observed order is at least as close and reporting significance as a/N. The significance computation thus accounts both

[3]This example is taken from an experiment earlier than that reflected in Table IX. Here, the Observability dimension is discretized into just three settings: Easy, Hard, Covert.

TABLE IX
Performance Influence Case Study (Group Detection)



Dataset Number	Group Connectivity	Noise, Clutter	Dataset Size	Population Size	Observable 2-way-comms per Individual	Pattern Complexity	Observability	Corruption	Aliasing	Confusability	Target Duty Level	Individual Duty Level
							Perfect					
							Easy					
	None						Fair		None			
	Easy	Y3		Y1			Hard	None	Easy	Easy	Easy	Easy
	Fair	Y2.5	Y3	Y2.5		Thin	Y2.5	Fair	Fair	Fair	Fair	Fair
	Hard	Y1	Y2.5	Y3		Fat	Covert	Hard	Hard	Hard	Hard	Hard
25	None	Y3	Y3	Y2.5	170.77	Thin	Fair	None	None	Easy	Easy	Easy
27	None	Y3	Y3	Y1	1982.75	Thin	Y2.5	Hard	None	Easy	Easy	Hard
EP	Easy	Y1	Y2.5	Y1	127.80	Thin	Easy	None	None	Easy	Easy	Easy
24	None	Y2.5	Y3	Y3	13.41	Fat	Hard	None	None	Easy	Easy	Easy
20	None	Y2.5	Y2.5	Y2.5	16.35	Thin	Perfect	None	Easy	Easy	Fair	Fair
38	Hard	Y3	Y3	Y1	1818.23	Thin	Perfect	None	None	Easy	Hard	Hard
28	None	Y1	Y2.5	Y2.5	14.47	Fat	Fair	Fair	None	Hard	Hard	Fair
18	Easy	Y2.5	Y2.5	Y1	136.69	Thin	Perfect	None	Fair	Fair	Hard	Easy
30	Easy	Y3	Y3	Y2.5	175.22	Thin	Easy	Hard	Easy	Fair	Hard	Easy
35	Fair	Y3	Y3	Y1	1787.58	Thin	Covert	Hard	None	Hard	Fair	Easy
39	Hard	Y3	Y3	Y1	1890.39	Thin	Fair	Fair	None	Fair	Fair	Fair
40	Easy	Y1	Y2.5	Y2.5	12.22	Fat	Fair	None	None	Easy	Fair	Hard
36	Easy	Y1	Y2.5	Y2.5	12.37	Fat	Hard	Fair	None	Easy	Easy	Fair
26	Fair	Y2.5	Y3	Y3	17.77	Fat	Easy	Hard	None	Hard	Hard	Easy
23	Hard	Y2.5	Y3	Y3	14.65	Fat	Y2.5	Hard	None	Easy	Easy	Fair
22	Hard	Y2.5	Y3	Y3	15.71	Fat	Hard	Fair	None	Easy	Hard	Hard
19	Fair	Y2.5	Y2.5	Y2.5	14.89	Thin	Hard	Fair	None	Fair	Easy	Hard
37	Fair	Y1	Y2.5	Y2.5	12.46	Fat	Covert	Hard	None	Hard	Fair	Fair
29	Hard	Y1	Y2.5	Y2.5	12.67	Fat	Covert	None	None	Hard	Easy	Easy
32	Fair	Y1	Y2.5	Y2.5	13.11	Fat	Easy	Fair	Hard	Fair	Fair	Easy
31	Hard	Y1	Y2.5	Y2.5	10.68	Fat	Hard	Hard	Fair	Fair	Fair	Hard

for the closeness of the observed order to the ideal and for variability of settings among the datasets.

By way of a case study, we include Table IX, covering results for a selected technology configuration [2] with the group detection objective, (to provide membership lists for all of the threat groups). Table IX covers an additional dimension (not included in Table IV) relevant to the technology configuration: Observed 2-way-comms per Individual. The 21 datasets processed using the selected technology are sorted by group detection performance (noted lower right).[4] Each dataset dimension column is headed by an idealized settings order. Under the dimension name, significance is plotted on a log scale.

With a scoring option in effect to resolve aliases automatically from ground truth, Group Connectivity is the most significant influence: chance probability = 0.0006. (Without this option, Aliasing is.) We split the dataset mix along this dimension to continue analysis, with results shown below.

[4]The experiment reflected in Table IX included 24 datasets developed from the pool specifications discussed in Section 5.2. The results were developed in the context of a technology integration experiment; a different group detection technology was used to process the three datasets omitted from Table IX.

- Group Connectivity (GC) *at 0.0006 significance*:
  —GC = None: *(No dimension of convincing significance)*
  —GC = Easy: *(No dimension of convincing significance)*
  —GC = Fair or Hard: Observed 2-way-comms per Individual *at 0.0005 significance*

Both Group Connectivity and Observed 2-way-comms per Individual are relevant to group detection intuitively as well as in the group detector's implementation.

5.6. Related Work (Experiment Design)

Hoffman and Jameson [17] present a multi-sensor, multi-target geospatial tracking testbed with a multi-dimensional dataset generation facility to explore the performance boundaries of a particular data fusion system implementation. They identify dimensions of problem complexity (or solution difficulty), generate datasets with parameter values varying along the different dimensions (apparently following the same kind of star-shaped experimental design we discussed in Section 5.2), and determine limits of acceptable performance for the fusion system and subsystems using tra-

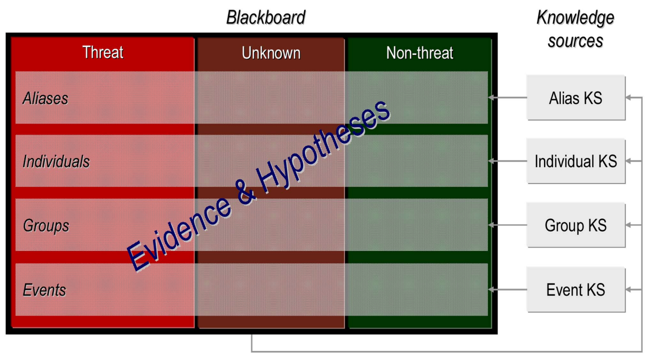


Fig. 28. Blackboard-based component integration.

ditional precision and recall along with other tracking metrics. Our work differs from theirs in several ways.

- Our problem focuses on inference of higher-level activities from discrete transaction evidence.
- Our evaluation applies object-oriented metrics to structured hypotheses.
- Our experimentation employs a fractional factorial design to differentiate performance of multiple solution implementations.
- Our analysis uses a novel rank correlation test to determine the problem dimensions most affecting a given technology's performance.

6. PE LAB ADVANTAGES FOR INFORMATION FUSION SYSTEM DESIGN

The overall PE Lab supports advanced threat detection technology development in several ways.

As reported here, we assess technical progress through program-wide evaluation and identify particular problem characteristics most influential to a technology's performance. Besides assisting individual technologists, this process can identify alternative technologies' relative strengths and elucidate potentially advantageous combinations.

Within a functional architecture (such as the blackboard architecture schematized in Fig. 28), we can employ the PE Lab to validate assumptions about the performance of a downstream component (or blackboard knowledge source—KS) based on that of an upstream one.

Suppose, *e.g.*, that a group detector depends on an alias resolver to deliver sufficiently de-aliased evidence about individuals. If the resolver is not yet performing at a goal level meeting the detector's input specs, we can still ascertain validity of performance claims for the latter by stubbing the former with a direct feed of evidence having per-spec de-aliasing. This can help to pinpoint performance gaps among functional components early in the development process.

In the future, we hope to facilitate such exploratory experimentation *via* a PE Lab-based component test harness and a program-wide commitment to automated (*i.e.*, hands-off) component execution. This has the potential to institutionalize the evaluation/experimentation process as a near-continuous loop in which experiments result in performance feedback to technology developers and developers respond to performance deficits with updated component versions. It also would enhance opportunities for large-scale experimentation.

7. CONCLUSIONS

Our hypothesis scoring methods are applicable in principle to performance evaluation in any domain where technologies return instances of one or more structured object types, given a problem for which an answer key is available. We expect that our alert scoring methods may be applied with benefit in other information fusion domains where hypothesis timeliness is important. Our experimental design methods may benefit other fusion (especially situation assessment) applications during exploratory system design and development.

PE Lab datasets and documentation are currently available to U.S. Government-approved users. Documentation covers concept of operations, event generation pattern language and counter-terrorism domain patterns, dataset generation algorithms, ontology, database schema, case scoring specifications, hypothesis format, and user instructions for the hypothesis scoring and dataset generation software.

ACKNOWLEDGMENTS

We thank our colleagues who have contributed to the PE Lab's overall design, implementation, and application, notably (with apologies to anyone we overlook) Jaffar Adibi, Chad Bisk, Chris Boner, Dan Bostwick, Hans Chalupsky, Thayne Coffman, Hans Dettmar, Seth Greenblatt, Ian Harrison, Dan Hunter, David Jensen, Kathy Laskey, Sherry Marcus, Andrew Moore, Kendra Moore, Peter Olson, David Page, Nick Pioch, Mike Pool, Josh Powers, Eric Rickard, Ben Rode, Jeff Schneider, Ted Senator, Andrei Sobolevskii, Sam Steingold, Afzal Upal, Andre Valente, Didier Vergamini, Bob Washburn, and Ed Wright. We additionally thank Randy Dieterle, Brandon Goldfedder, Tod Levitt, and anonymous reviewers for their constructive comments.

REFERENCES

[1] J. Adibi, T. Barrett, S. Bhatt, H. Chalupsky, J. Chame and M. Hall
Processing-in-memory technology for knowledge discovery algorithms.
In *Proceedings of the Second International Workshop on Data Management on New Hardware* (DaMoN 2006), June 2006, http://www.isi.edu/~hans/publications/DAMON06.pdf.

[2] J. Adibi and H. Chalupsky
Scalable group detection via a mutual information model.
In *Proceedings of the First International Conference on Intelligence Analysis* (IA-2005), https://analysis.mitre.org/proceedings/Final_Papers_Files/355_Camera_Ready_Paper.pdf.

[3] J. Adibi, H. Chalupsky, E. Melz and A. Valente
The KOJAK group finder: Connecting the dots via integrated knowledge-based and statistical reasoning.
In *Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence Conference* (IAAI-04), 2004, http://www.isi.edu/∼hans/publications/IAAI04.pdf.

[4] J. Adibi, P. R. Cohen and C. T. Morrison
Measuring confidence intervals in link discovery: A bootstrap approach.
In *Workshop on Link Analysis and Group Detection* (LinkKDD2004), held at the Tenth Association for Computing Machinery (ACM) Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) International Conference on Knowledge Discovery and Data Mining, Aug. 2004, http://crue.isi.edu/files/papers/adibi_2006_1141857733.pdf.

[5] D. W. Aha, J. W. Murdock and L. A. Breslow
Managing terrorist activity hypotheses.
Technical Report AIC-03-188, Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Intelligent Decision Aids Group, Washington, D.C., 2003.

[6] D. P. Bertsekas and D. A. Castañon
A forward/reverse auction algorithm for asymmetric assignment problems.
Technical Report LIDS-P-2159, Massachusetts Institute of Technology, 1993, http://citeseer.ist.psu.edu/bertsekas93forwardreverse.html.

[7] C. Boner
Automated detection of terrorist activities through link discovery within massive databases.
In Proceedings of the *American Association for Artificial Intelligence (AAAI) Spring Symposium on A.I. Technologies for Homeland Security*, Palo Alto, CA, Mar. 2005.

[8] C. M. Boner
Novel, complementary technologies for detecting threat activities within massive amounts of transactional data.
In *Proceedings of the First International Conference on Intelligence Analysis* (IA-2005), https://analysis.mitre.org/proceedings/Final_Papers_Files/ 318_Camera_Ready_Paper.pdf.

[9] J. Coble, D. Cook and L. Holder
Structure discovery in sequentially-connected data streams.
International Journal on Artificial Intelligence Tools, **15**, 6 (Dec. 2006), http://www.eecs.wsu.edu/∼holder/papers/CobleIJAIT06.pdf.

[10] J. Coble, R. Rathi, D. Cook and L. Holder
Iterative structure discovery in graph-based data.
International Journal on Artificial Intelligence Tools, **14**, 1–2 (Feb.–Mar. 2005), http://www.eecs.wsu.edu/∼holder/papers/CobleIJAIT05.pdf.

[11] D. Davenport and C. R. Hill
Fast abductive reasoning over ontologies.
In *American Association for Artificial Intelligence (AAAI) Fall Symposium on Capturing and Using Patterns for Evidence Detection*, 2006, http://demo.cs.brandeis.edu/papers/author.html#viswanathan.

[12] J. Davis, V. Santos Costa, I. M. Ong, D. Page and I. Dutra
Using Bayesian classifiers to combine rules.
In *3rd Association for Computing Machinery (ACM) Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Workshop on Multi-Relational Data Mining (MRDM 2004)*, 2004, http://www.cs.wisc.edu/∼dpage/nbayes.pdf.

[13] J. Davis, I. Dutra, D. Page and V. Santos Costa
Establishing identity equivalence in multi-relational domains.
In *Proceedings of the First International Conference on Intelligence Analysis* (IA-2005), 2005, https://analysis.mitre.org/proceedings/Final_Papers_Files/ 378_Camera_Ready_Paper.pdf.

[14] J. Eilbert, J. Hicinbothom and A. Karnavat
A cognitive architecture focusing on situation estimation: merging perceptual and cognitive processes.
In *Behavior Representation in Modeling and Simulation* (BRIMS), 2005.

[15] B. Gallagher
Matching structure and semantics: A survey on graph-based pattern matching.
In *American Association for Artificial Intelligence (AAAI) Fall Symposium on Capturing and Using Patterns for Evidence Detection*, 2006.

[16] G. Gu, P. Fogla, D. Dagon, W. Lee and B. Škoríc
Measuring intrusion detection capability: An information-theoretic approach.
In *Proceedings of ACM Symposium on InformAtion, Computer and Communications Security* (ASIACCS'06), 2006, http://www-static.cc.gatech.edu/∼guofei/paper/Gu_asiaccs06_cid.pdf.

[17] M. Hoffman and S. Jameson
Complexity and performance assessment for data fusion systems.
Lockheed Martin Advanced Technology Laboratories, 1998, http://www.atl.external.lmco.com/overview/papers/920.pdf.

[18] L. Holder, D. Cook, J. Coble and M. Mukherjee
Graph-based relational learning with application to security.
Fundamenta Informaticae special issue on Mining Graphs, Trees and Sequences, **66**, 1–2 (Mar. 2005), 83–101, http://www.eecs.wsu.edu/∼holder/papers/HolderFI05.pdf.

[19] R. Jonker and A. Volgenant
A shortest augmenting path algorithm for dense and sparse linear assignment problems.
Computing. **38**, 4 (Dec. 1987), 325–340.

[20] S. Létourneau, F. Famili and S. Matwin
Data mining for prediction of aircraft component replacement.
In *IEEE Intelligent Systems* special issue on Data Mining, (Dec. 1999), 59–66, http://www.site.uottawa.ca/∼stan/papers/1999/componentFailureIEEE.pdf.

[21] J. Llinas, C. Bowman, G. L. Rogova, A. Steinberg, E. Waltz and F. White
Revisiting the JDL data fusion model II.
In *Proceedings of the Seventh International Conference on Information Fusion* (Fusion 2004), June 28–July 1, 2004, Stockholm, Sweden, 1218–1230, http://www.fusion2004.foi.se/papers/IF04-1218.pdf.

[22] S. A. Macskassy and F. Provost
A simple relational classifier.
In *2nd Association for Computing Machinery (ACM) Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Workshop on Multi-Relational Data Mining (MRDM 2003)*, http://www-ai.ijs.si/SasoDzeroski/MRDM2003/proceedings/macskassy.pdf.

[23] S. A. Macskassy and F. Provost
Suspicion scoring based on guilt-by-association, collective inference, and focused data access.
In *Proceedings of the International Conference on Intelligence Analysis*, 2005, https://analysis.mitre.org/proceedings/Final_Papers_Files/ 273_Camera_Ready_Paper.pdf.

[24] S. A. Macskassy, and F. Provost
Suspicion scoring based on guilt-by-association, collective inference, and focused data access.
In *Proceedings of the North American Association for Computational Social and Organization Sciences (NAACSOS) Conference*, 2005, http://www.research.rutgers.edu/∼sofmac/paper/naacsos2005-app/ macskassy-naacsos2005-app.pdf.

[25] S. A. Macskassy and F. Provost
A brief survey of machine learning methods for classification in networked data and an application to suspicion scoring.
In *Workshop on Statistical Network Learning* (poster) at *23rd International Conference on Machine Learning*, 2006, http://www.research.rutgers.edu/∼sofmac/paper/sna2006/abstract.html.

[26] S. Mahoney, K. Laskey, E. Wright and K. Ng
Measuring performance for situation assessment.
In *Proceedings of the Military Sensing Symposia (MSS) National Symposium on Sensor and Data Fusion*, San Antonio, TX, 2000, http://ite.gmu.edu/∼klaskey/papers/Mahoney_et_al_Fusion2000.pdf.

[27] C. Morrison, P. Cohen, G. King, J. Moody and A. Hannon
Simulating terrorist threat in the Hats simulator.
In *Proceedings of the First International Conference on Intelligence Analysis*, MITRE Corp., 2005, https://analysis.mitre.org/proceedings/Final_Papers_Files/ 265_Camera_Ready_Paper.pdf.

[28] C. T. Morrison and P. R. Cohen
The Hats information fusion challenge problem.
In *Proceedings of the 9th International Conference on Information Fusion* (Fusion 2006), special session on Distributed Inference and Decision-Making in Multisensor Systems, 2006, http://crue.isi.edu/files/papers/clayton_2006_1151624442.pdf.

[29] M. Mukherjee and L. Holder
Graph-based data mining for social network analysis.
In *Workshop on Link Analysis and Group Detection* (LinkKDD2004), held at the *Tenth Association for Computing Machinery (ACM) Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) International Conference on Knowledge Discovery and Data Mining*, Aug. 2004, http://www.eecs.wsu.edu/∼holder/papers/HolderFI05.pdf.

[30] Patrick Pantel
Alias detection in malicious environments.
In *American Association for Artificial Intelligence (AAAI) Fall Symposium on Capturing and Using Patterns for Evidence Detection*, 2006, http://www.patrickpantel.com/cgi-bin/Web/Tools/ getfile.pl?type=paper&id=2006/aaai-fs06.pdf.

[31] N. J. Pioch, D. Hunter, J. V. White, A. Kao, D. Bostwick and E. K. Jones
Multi-hypothesis abductive reasoning for link discovery.
In *Workshop on Link Analysis and Group Detection* (LinkKDD2004), held at the *Tenth Association for Computing Machinery (ACM) Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) International Conference on Knowledge Discovery and Data Mining*, Aug. 2004, www.cs.cmu.edu/∼dunja/LinkKDD2004/Nicholas-Pioch-LinkKDD-2004.pdf.

[32] J. Potts, L. Holder, D. Cook and J. Coble
Learning concepts from intelligence data embedded in a supervised graph.
In *Proceedings of the International Conference on Intelligence Analysis*, May 2005, http://www.eecs.wsu.edu/∼holder/papers/PottsIA05.pdf.

[33] T. Senator
Evidence Extraction and Link Discovery program.
In *DARPATech 2002 Symposium*, 2002, http://www.darpa.mil/DARPATech2002/presentations/iao_pdf/speeches/SENATOR.pdf.

[34] R. C. Schrag and M. Takikawa
Cyber situation assessment evaluation participant's guide.
Information Extraction and Transport, Inc., July 2006.

[35] R. C. Schrag and M. Takikawa
Scoring hypotheses from threat detection technologies: Analogies to machine learning evaluation.
In *American Association for Artificial Intelligence (AAAI) Workshop on Evaluation Methods for Machine Learning II*, 2007, http://www.globalinfotek.com/schrag/Schrag-Takikawa-AAAI-2007-WS04-color.pdf.

[36] R. C. Schrag, M. Takikawa, P. Goger and J. Eilbert
Scoring alerts from threat detection technologies.
In *American Association for Artificial Intelligence (AAAI) Fall Symposium on Capturing and Using Patterns for Evidence Detection*, 2006, http://www.globalinfotek.com/schrag/Schrag-et-al-AAAI-Fall-2006.pdf.

[37] A. N. Steinberg
Unification across data fusion levels 1–3.
In *Proceedings, Seventh International Conference on Information Fusion*, Stockholm, Sweden, 2004.

[38] A. N. Steinberg
Open networks: Generalized multi-sensor characterization.
In *Proceedings of the Ninth International Conference on Information Fusion*, Florence, Italy, 2006.

[39] G. Tadda, J. Salerno, D. Boulware, M. Hinman and S. Gorton
Realizing situation awareness in a cyber environment.
In Belur V. Dasarathy (Ed.), *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2006*, *Proceedings of SPIE* vol. 6242, 2006, 624204-(1–8).

[40] J. Thomere, I. Harrison, J. Lowrance, A. Rodriguez, E. Ruspini E and M. Wolverton
Helping intelligence analysts detect threats in overflowing, changing and incomplete information.
In *Proceedings of the 2004 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, July 2004, 39–45, http://www.ai.sri.com/pubs/files/1042.pdf.

[41] J. E. Tierno
Performance analysis and prediction for data mining systems.
In *American Association for Artificial Intelligence* (AAAI), Spring Symposium A.I. Technologies for Homeland Security, Palo Alto, CA, Mar. 2005.

[42] G. Weiss and H. Hirsh
Learning to predict rare events in event sequences.
In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998, 359–363, http://www.cs.rutgers.edu/∼gweiss/papers/ts-workshop.pdf.

[43] J. V. White and C. G. Fournelle
Threat detection for improved link discovery.
In *Proceedings of the International Conference on Intelligence Analysis*, May 2005, https://analysis.mitre.org/proceedings/Final_Papers_Files/ 333_Camera_Ready_Paper.pdf.

[44] M. Wolverton and J. Thomere
The role of higher-order constructs in the inexact matching of semantic graphs.
In D. Mladenic, N. Milic-Frayling and M. Grobelnik (Eds.), *Link Analysis: Papers from the 2005 American Association for Artificial Intelligence (AAAI) Workshop*, technical report WS-05-07, 1–7, http://www.ai.sri.com/pubs/files/1146.pdf.

Robert Carl Schrag was born at Griffiss Air Force Base in Rome, NY on December 3, 1957, the son of Vernon Dale Schrag and the former Marilyn Louise German. After high school in Endicott, NY, he attended the State University of New York at Binghamton (now Binghamton University) and the Massachusetts Institute of Technology and received a B.A. in natural sciences from The Johns Hopkins University in 1979.

In 1980, Schrag joined a new artificial intelligence research group at Rome Air Development Center (RADC—now part of the Air Force Research Lab) at Griffiss. While at RADC, Schrag monitored contracts and performed in-house research in knowledge-based systems and logic programming. He received an M.S. in computer and information sciences from Syracuse University in 1983 and in 1984 joined VOIS, Inc.—a small company researching speech recognition and psychoacoustics in Endicott. In 1985, he returned to logic-based artificial intelligence, joining the Honeywell Systems and Research Center in Minneapolis, MN.

Schrag led his first sponsored research effort at Honeywell, developing temporal reasoning technology in collaboration with Tom Dean of Brown University, with funding from the Defense Advanced Research Projects Agency (DARPA) and RADC. Having enjoyed the collegiality of this experience, he decided to return to school in 1992. At the University of Texas at Austin, he studied computer science, beginning under Ben Kuipers and finishing under Dan Miranker. He also spent the summer of 1993 at AT&T Bell Telephone Laboratories in Murray Hill, NJ, working with James Crawford. His 1996 Ph.D. dissertation, entitled "Search in SAT/CSP: Phase Transitions, Abstraction, and Compilation," focused on propositional satisfiability (SAT) and the finite-domain constraint satisfaction problem (CSP).

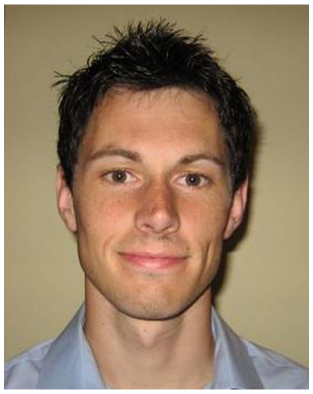
In 1996, Schrag joined Information Extraction and Transport, Inc. (IET), of Arlington, VA, where he developed and conducted multi-participant evaluations for six different artificial intelligence research programs funded by DARPA (High Performance Knowledge Bases, Rapid Knowledge Formation, Evidence Extraction and Link Discovery) and other U.S. Government agencies. At IET, he also led an effort to build a multiple-context facility for DARPA's UltraLog military logistics agent-based planning and scheduling environment.

In 2007, Schrag joined Global InfoTek, Inc., of Reston, VA, where he is a principal scientist working on applications of technology to U.S. national security problems.

Robert Carl Schrag was born again (in Jesus Christ) on December 3, 1986. In 1990, he covenanted marriage with the former Robin Sue Maxwell of Montevideo, MN. At this writing, they have two children and reside in Fairfax County, VA.

Masami Takikawa received his B.A. and M.A. in behavioral science from Hokkaido University in 1986 and 1988, respectively, and his M.S. and Ph.D. in computer science from Oregon State University in 1992 and 1998, respectively.

He is currently Architect of On-Demand Personalization at Art Technology Group (ATG), Inc., working on the application of relational real-time Bayesian inference technology to e-commerce. From 1997 to 2007, Dr. Takikawa was principal engineer at Information Extraction & Transport (IET), Inc., specializing in the development and application of various AI technologies to data fusion, situation assessment, and link discovery for government and industry.



Paul Goger graduated from William and Mary in 2001 with an undergraduate degree in mathematics and computer science. In his last year, he completed his senior thesis with Professor Michael Trosset in stochastic optimization. After college, Paul was hired by Metron, Inc., Reston, VA.

His early areas of work involved data fusion applied to the problem of anti-submarine warfare. For the last several years he has been working in the realm of counterterrorism.

James L. Eilbert earned a Ph.D. in biomathematics from North Carolina State University, Raleigh, NC (1980), and a M.S. in applied mathematics from Courant Institute, New York University, New York, NY (1975).

He is currently CEO of AP Technology, LLC in Titusville, NJ. He has twenty-five years experience in research and development focusing on cognitive modeling and computer vision. He has been principal investigator on projects that developed techniques for information fusion and situation assessment, cognitive modeling of intelligence analysts, modeling vision with a focus on the interaction of perception and cognition, modeling context and context switching, and automatic target recognition. He has applied statistical techniques and neural networks to the analysis of medical, robotic, and satellite imagery. He is currently working on biologically plausible models of attention and emotion, and their role in situation estimation and activity recognition.